UNIVERSITY OF DELHI

CNC-II/093/1/EC-1273/25/ 517
Dated: 06.03.2025

NOTIFICATION

Sub: Amendment to Ordinance V

(ECR 38-34 dated 17.01.2025)

Following addition be made to Appendix-II-A to the Ordinance V (2-A) of the Ordinances of the University;

Add the following:

The Syllabi of the following Programmes for Semester-V and Semester-VI under the Faculty of Technology based on Undergraduate Curriculum Framework 2022, are notified herewith for the information of all concerned:

1. B.Tech Electrical Engineering - As per ***Annexure-1***
2. B.Tech Computer Science and Engineering - As per ***Annexure-2***
3. B.Tech Electronics and Communication Engineering - As per ***Annexure-3***

6/3/25

REGISTRAR

Annexure-1

Department of Electrical Engineering
Faculty of Technology
University of Delhi
Detailed Course Structure and Curriculum of B.Tech. (EE) Third Year

Department of Electrical Engineering
Faculty of Technology
University of Delhi

Course Structure of B. Tech. (EE) Third Year

Semester V						
S. No.	**Course Domain**	**Course Title**	**Credits***			**Total Credits**
			L	**T**	**P**	
1	DSC-13	Power System Analysis	3	0	1	4
2	DSC-14	Control System	3	0	1	4
3	DSC-15	Electromagnetic Field Theory	3	0	1	4
4	DSE-03	Select a course from the specified list of DSE-3				4
5	GE-05	Select a course from the specified list of GE-5				4
6	SEC / IAPC	Choose one SEC or Internship / Apprenticeship / Project / Community Outreach (IAPC)				2
Total Credits						**22**

Semester VI						
S. No.	**Course Domain**	**Course Title**	**Credits***			**Total Credits**
			L	**T**	**P**	
1	DSC-16	Switchgear and Protection	3	0	1	4
2	DSC-17	Embedded System Technologies	3	0	1	4
3	DSC 18	Power Electronics	3	0	1	4
4	DSE-04	Select a course from the specified list of DSE-4				4
5	GE-06	Select a course from the specified list of GE-6				4
6	SEC / IAPC	Choose one SEC or Internship / Apprenticeship / Project / Community Outreach (IAPC)				2
Total Credits						**22**

**Credits*
L (01 Credit) is equivalent to 01 contact hour per week.
T (01 Credit) is equivalent to 01 contact hour per week.
P (01 Credit) is equivalent to 02 contact hours per week.

Department of Electrical Engineering
Faculty of Technology
University of Delhi

Pool of DSEs offered by the Department of Electrical Engineering in Third Year

S. No.	Semester	DSE	Course Title
1.	V	DSE-3	Utilization of Electric Power
2.			Economic Operations of Power System
3.	VI	DSE-4	Digital Control System
4.			Digital Signal Processing

List of SECs offered by the Department of Electrical Engineering in Third Year

S. No.	Semester	Course Title
1.	V	IAPC
2.	VI	Advanced Electrical Workshop-II

Department of Electrical Engineering
Faculty of Technology
University of Delhi

Specializations and Minor offered by the Department of Electrical Engineering

S. No.	Sem	DSE/ GE	Minor in EE (Open only for CSE/ ECE)	Specializations for EE / Minors for ECE and CSE		
				Robotics and Automation	**Sustainable Energy Engineering**	**Electric and Hybrid Vehicle**
1	V	DES-3/ GE-5	Electrical Measurement Technique	Introduction to Robotics and Mechatronics	Energy Conservation and Audit	Electrical Storage and Management System
2	VI	DSE-4/ GE-6	Power Converters and Applications	Industrial Automation	Energy Policies for Sustainable Development	Power Electronics Converters and Drives for Electric Vehicles

Detailed Syllabus of Discipline Specific Core (DSC) Courses of B. Tech. (EE) – SEMESTER V

Power System Analysis (DSC-13)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Power System Analysis	4	3	0	1	Power Transmission and Distribution

Course Hours: L-03, T-00, P-02

Course Objectives:

1. Understand Power System Components.
2. Analyze Power Flow.
3. To identify the impact of short circuits, line faults, and other disturbances on the system..
4. Explore Stability and Control.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. To model power system components for simulation in an integrated manner.
2. Analyzing and estimating power flow through the network to meet load demand from generation.
3. To evaluate short circuit levels at different buses for adequate protection against damage.
4. To keep the power system running in synchronism under any small or large disturbance.
5. To avoid disintegrating the power system due to voltage instability.

UNIT-I

Power System Components and Modeling: Transmission lines, one line diagram, impedance and reactance diagram, per unit system.

Load Flow Analysis: Introduction, nodal admittance matrix analysis (Y-bus), the concept of bus impedance matrix (Z-bus) and its building procedure, bus classifications, development of load flow equations, load flow solution using Gauss-Siedel, and Newton-Raphson, Jacobian Matrix, fast decoupled methods.

UNIT-II

Faults and Short Circuit Analysis: Symmetrical three-phase fault analysis, use of Z-bus in computation of short circuit currents, short circuit capacity at a bus, selection of circuit breaker, use of current limiting reactors.

Significance of positive, negative and zero sequence components, sequence impedances and sequence networks equations, unsymmetrical short circuit analysis - single line to ground fault, line to line fault, double line to ground fault on power systems, faults with fault impedance, open circuit faults.

UNIT-III

Power System Stability: Swing equation, power angle equation, synchronizing power coefficient, basic concepts of steady state, dynamic and transient stability, equal area criterion, solution of the swing equation, multi-machine transient stability studies with classical machine representation.

UNIT-IV

Voltage Stability: Introduction, comparison of angle and voltage stability, reactive power flow and voltage collapse, mathematical formulation of voltage stability problem, voltage stability analysis, prevention of voltage collapse, trends and challenges.

Suggestive Readings:

1. G. W. Stagg, and A. H. El-Abiad, “Computer Methods in Power System Analysis”, McGraw Hill Kogakusha, 1968.
2. John Grainger, William Stevenson, “Power System Analysis,” McGraw Hill, 2017.
3. Abhijit Chakraborty, and Sunita Halder, “Power System Analysis, Operation and Control”, PHI, New Delhi, 2011.
4. M. A. Pai, “Computer Techniques in Power System Analysis”, Tata McGraw Hill, New Delhi, 2006.
5. Carson W. Taylor, “Power System Voltage Stability”, McGraw-Hill, 1994.

List of Experiments:

1. To study and test a typical Radial DC Distribution system supplied from one and both ends.
2. To study and test a typical Ring main DC Distribution system.
3. Ferranti Effect of Single-Phase Transmission Line.
4. Study of Short Transmission Line for calculation of various parameters.
5. P-V Characteristics of Single-Phase Transmission Line.
6. Measurement of Capacitance of Three-Core Cable.
7. Determination of Voltage Drop in a Cable.
8. To locate the fault in a cable using the Murray Loop Test.
9. Simulation of the string of insulation with and without a guard ring and evaluation of its efficiency.
10. To determine the dielectric strength of the given transformer oil using an oil testing Kit.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Control System (DSC-14)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Control Systems	4	3	0	1	Introduction to Electrical and Electronics Engineering, Mathematics-I, Electrical Network Analysis

Course Hours: L-03, T-00, P-02

Course Objectives:

1. To acquire the knowledge of control systems.
2. To model and analyze the physical systems for controlling their responses.
3. To design and analyze the stability and performance of control systems.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Acquire and demonstrate knowledge of various types of control systems.
2. Model the physical system using the transfer function and state space method and analyze the time domain responses of first and second-order systems.
3. Analyze the stability of control systems.
4. Design PID control and various types of compensators.

UNIT-I

Introduction: Open loop and closed loop control systems, feedback, effects of feedback, linear and non-linear control systems, block diagrams, some examples.

Modelling: Modeling of physical system: electrical, mechanical, translational, rotational, electrical, mechanical analogies, Laplace transform, transfer function, characteristic equation, block diagram algebra, signal flow graphs, error detectors potentiometer, synchros, stepper motor, AC and DC tacho-generators.

UNIT-II

Time Domain Analysis: Importance of time response in transient and steady-state analysis, typical test input signals, transient response of the first order and second order system, time response specifications, dominant closed-loop poles of higher order systems, steady-state error and error coefficients.

Stability: Concepts of absolute and relative stability, pole-zero location, Routh Hurwitz criteria.

UNIT-III

Root Locus Technique: Introduction, root locus concept, construction of root loci, stability analysis.

Transient behaviour and initial conditions: Behavior of circuit elements under switching condition and their representation, evaluation of initial and final conditions in RL, RC and RLC circuits for AC and DC excitations.

UNIT-IV

Frequency Response: Introduction and importance of frequency response, bode diagram, polar plots, Nyquist stability criterion, stability analysis, relative stability, gain margin & phase margin, closed-loop frequency response.

Introduction to Design: Necessity of compensation, lag and lead compensation, PID controller.

State Space Analysis: Concept of state, state variable and state vector, state transition matrix, controllability and observability, solution of state equation.

Suggestive Readings:

1. Control Systems Engineering by I J Nagrath and M Gopal, Wiley Eastern.
2. Linear Control Systems by B S Manke.
3. Raymond A. DeCarlo, Pen-Min Lin, Linear Circuit Analysis, OUP USA; 2nd edition.
4. Automatic Control systems by B C Kuo.
5. Modern Control Engineering by K Ogata, PHI.

List of Experiments:

1. To obtain the time responses of first-order and second-order RLC circuits.
2. To simulate the various responses of the linear system using a linear system simulator.
3. To study and implement the temperature-controlled system.
4. To study the performances of open-loop and closed-loop systems.
5. To implement the characteristics of the stepper motor interfaced with a microprocessor.
6. To study the closed loop performances with P, PI and PID controllers.
7. To implement lag, lead and lag-lead compensators.
8. To study and implement the various characteristics of DC motor position control system.
9. To study synchro-transmitter and receiver and obtain output versus input characteristics.
10. To study the AC position trainer kit and analyze its performance.
11. To draw Nyquist plot of open loop transfer functions and examine the stability of the closed loop system.
12. To obtain the Bode frequency response for first and second-order systems.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Electromagnetic Field Theory (DSC-15)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Electromagnetic Field Theory	4	3	0	1	Introduction to Electrical and Electronics Engineering, Physics, Mathematics-I

Course Hours: L-03, T-00, P-02

Course Objectives:

1. To understand Maxwell's Equation and apply it to the basic electromagnetic problem.
2. To interpret the given problem and solve it using Maxwell's equations.
3. To analyze time varying electric and magnetic fields, wave propagation in different media.
4. To understand transmission line fundamentals and apply them to the basic problem.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Recognize and classify the basic Electrostatic theorems and laws and derive them.
2. Discuss the behavior of Electric fields in matter and Polarization concepts.
3. Classify the basic Magneto static theorems and laws and infer the magnetic properties of matter.
4. Summarize the concepts of electrodynamics and derive and discuss Maxwell's equations.
5. Students are expected to be familiar with Electromagnetic wave propagation.

UNIT-I

Preliminaries: Physical interpretation of gradient, divergence and curl. The Laplacian operator, vector relationship in rectangular, cylindrical and spherical polar coordinate systems, divergence and curl equations and its Integral forms, Stoke's Theorem, Green's Theorem, Dirac delta distribution. Field as derivative of potential, Helmholtz Theorems.

UNIT-II

Electrostatic Field: Coulomb's Law, electrostatic field, Laplace and Poisson's equation, divergence and curl of electrostatic field, scalar potential, Field equations in different coordinate systems, boundary conditions, Continuity equation and relaxation time, Energy stored due to accumulation of charges.

Magnetostatic Field: Lorentz force, Biot-Savart's law, Scalar and vector potentials. Divergence and curl of magnetic field, Ampere's law, Force and Torque equations, field equations in different coordinate systems. Boundary conditions, magnetic vector potential and flux, energy stored in a magneto static field.

UNIT-III

Dynamic electric and magnetic fields: Time varying fields and Faraday‘s law. Displacement current, Maxwell's correction to Ampere's law, relation between electric and Magnetic fields.

Poynting's Theorem and flow of power: Poynting‘s theorem and its equivalence to energy conservation law, Poynting‘s vector, power flow and relevance to power transmission.

UNIT-IV

Wave Equation: Maxwell's equations, Wave equations in free space and in conducting medium, Wave impedance. Wave propagation in Dielectrics, Propagation in Good Conductors, Skin Effect, Reflection of uniform Plane Waves at normal incidence, Plane Wave reflection at Oblique Incidence, Wave propagation in dispersive media.

Transmission Lines: Typical Transmission lines- Co-axial, Two Wire, Microstrip, Coplanar and Slot Lines, Transmission Line Parameters, Transmission Line Equations, Wave propagation in Transmission lines, low loss, lossless line, Distortionless line.

Suggestive Readings:

1. Matthew N.O. Sadiku, "Principles of electromagnetics" 4th edition, Oxford university Press, 2014.
2. Hayt Jr, William H., John A. Buck, and M. Jaleel Akhtar, "Engineering Electromagnetics (SIE)", McGraw-Hill Education, 2020.
3. Karl E. Longren, Sava V. Savov, Randy J. Jost., "Fundamentals of Electromagnetics with MATLAB", PHI (For MATLAB experiments)
4. D. C. Cheng, "Field and Wave Electromagnetics," Pearson Education (2001)

List of Experiments:

<u>Hardware based experiments</u>:

1. To study electric field patterns between two circular electrodes.
2. To study the electric field between parallel conductors.
3. To study Electric Field and Potential Inside the Parallel Plate Capacitor.
4. To study Capacitance and Inductance of Transmission Lines.
5. To study Magnetic Field Outside a Straight Conductor.
6. To study Magnetic Field of Coils.
7. To study Magnetic Inductions.

Simulation based experiments:

Write a program to:

1. Find gradient of a scalar field.
2. Find divergence of a vector field.
3. Find the curl of a vector field.
4. Transform
    a. spherical coordinates to cartesian coordinates.
    b. cylindrical coordinates to cartesian coordinates.
    c. cartesian coordinates to cylindrical coordinates.
    d. cartesian coordinates to spherical coordinates.
5. Represent electric field lines due to a point charge at origin.
6. Plot equipotential contours and electric field due to dipole.
7. Plot magnetic flux density due to current carrying wire.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Detailed Syllabus of Discipline Specific Elective (DSE) courses for B.Tech. (EE) – SEMESTER V

Utilization of Electric Power (DSE-03)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Utilization of Electric Power	4	3	1	0	Introduction to Electrical and Electronics Engineering, Power Transmission and Distribution

Course Hours: L-03, T-01, P-00

Course Objectives:

1. To introduce various electric drives and their applications.
2. To explain various techniques for designing indoor &outdoor lighting schemes.
3. To discuss different methods of electrical heating and electric welding.
4. To illustrate the fundamentals of electrolytic and electrometallurgical processes.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Recognize and classify the basic electric traction system.
2. Discuss the behavior of different lighting systems and their illuminance.
3. Classify the basic heating system used for industrial purposes.
4. Students are expected to be familiar with the different uses of electric systems in the industry.

UNIT-I

Electric Traction Drives: Advantages of electric drives, Characteristics of different mechanical loads, Parts of electric drives electric motors, close loop of electric drive system, Types of motors used in electric drive pulley drives etc., Examples of selection of motors for different types of domestic loads, Selection of drive for applications such as general workshop, textile mill, paper mill, steel mill, printing press, crane and lift etc.

UNIT-II

Illumination: Nature of light, visibility spectrum curve of relative sensitivity of human eye and wave length of light, Definition: Luminous flux, solid angle, luminous intensity, illumination, luminous efficiency, depreciation factor, coefficient of utilization, space to height ratio, reflection factor, glare, shadow, lux, Laws of illumination, Different type of lamps, construction and working of incandescent and discharge lamps – their characteristics, fittings required for filament lamp, mercury vapor lamp, fluorescent lamp, metal halide lamp, neon lamp, Main requirements of proper lighting; absence of glare, contrast and shadow, General ideas bout street lighting, flood lighting, monument lighting and decorative lighting, light characteristics etc.

UNIT- III

Electric Heating: Advantages of electrical heating, Heating methods: Resistance heating – direct and indirect resistance heating, electric ovens, their temperature range, properties of resistance heating elements, domestic water heaters and other heating appliances and thermostat control circuit, Induction heating; principle of core type and coreless induction furnace, Electric arc heating; direct and indirect arc heating, construction, working and applications of arc furnace.

UNIT -IV

Electric Welding: Advantages of electric welding, Welding method, Principles of resistance welding, types, Principle of arc production, electric arc welding, characteristics of arc; carbon arc, metal arc, hydrogen arc welding method of and their applications.

Electrical Circuits used in Refrigeration and Air Conditioning and Water Coolers: Principle of air conditioning, vapor pressure, refrigeration cycle, eco-friendly Refrigerants, Electrolytic Processes, Laws of electrolysis, process of electro-deposition clearing, operation, deposition of metals, polishing, buffing.

Suggestive Readings:

1. "Utilization of electrical energy" by E. O. Taylor and V. V. L. Rao, English Universities Press.
2. "Electrical Drives: Concept and applications" by Vedam Subrahmanyam" Tata McGraw Hill.
3. "Art and Science of Utilization of Electrical Energy" by H. Pratab, Dhanpat Rai & Co.

Economic Operations of Power System (DSE-3)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Power System Economics	4	3	1	0	Power Transmission and Distribution

Course Hours: L-03, T-01, P-00

Course Objectives:

5. To understand the fundamentals of power system economics.
6. To ensure cost-effective power generation while meeting system demands.
7. To develop skills in coordinating hydro and thermal power plants for optimal power generation.
8. To explore and apply various methods for optimal power flow (OPF) analysis, ensuring economic and secure operation of power systems.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

6. Demonstrate an understanding of power system economics, including the key principles of economic operation and optimization.
7. Analyze and solve the economic dispatch and unit commitment problems.
8. Apply hydro-thermal coordination techniques.
9. Solve optimal power flow (OPF) problems using various optimization techniques, ensuring economic, reliable, and secure operation of power systems.

UNIT-I
Introduction: Introduction to power system economics, Introduction to microeconomics, Introduction to economic operations, Evolution of Indian Power System, Control in Power Systems, Optimization Preliminaries.

UNIT-II
Economic Operation: Economic dispatch problem and methods of solutions, economic importance, characteristics of steam units, economic dispatch of thermal units and methods of solutions, problem considering and neglecting transmission losses, and iterative and non-iterative methods of solutions.
Unit Commitment: Definition, Constraints in Unit Commitment, Unit Commitment solution methods, Priority, List Methods, Dynamic Programming Solution, Economic dispatch versus Unit Commitment.

UNIT-III
Hydro-thermal coordination: Hydroelectric plant models, short-term hydrothermal scheduling problem, gradient approach, Hydro units in series, pumped storage hydro plants, hydro-scheduling using Dynamic programming and linear programming.

UNIT-IV
Optimal Power Flow: Introduction, Solution of OPF, gradient method, Newton's method, Linear Sensitivity analysis, linear programming method, Security Constrained OPF, Interior Point OPF, Bus Incremental Coats.

Suggestive Readings:

1. Allen J. Wood and Bruce F. Wollenberg, “Power Generation Operation and Control,” John Wiley & Sons, New York, 2016.
2. Elgerd O. I, “Electric Energy System Theory – an Introduction,” Tata McGraw Hill, New Delhi, 2013.
3. Robert H. Miller, James H. Malinowski, “Power System Operation,” Tata McGraw Hill, 2009.
4. J. C. Das, “Load Flow Optimization and Optimal Power Flow,” CRC press, 2017.
5. Abhijit Chakrabarti and Suita Halder, “Power System Analysis, Operation and Control,” PHI, 2010.

Detailed Syllabus of Discipline Specific Core (DSC) courses for B.Tech. (EE) – SEMESTER VI

Switchgear and Protection (DSC-16)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Switchgear and Protection	4	3	0	1	Power Transmission and Distribution, Power System Analysis

Course Hours: L-03, T-00, P-02

Course Objectives:

1. To get introduced to protective relays for power systems.
2. To be familiar with the protection schemes of generators and motors.
3. To know about how to protect transformers.
4. To design protection of transmission lines.
5. To be aware of the phenomenon of arcing and its interruption by circuit breakers.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Understanding Protective Relays and Instrument Transformers.
2. Knowledge of Protection Schemes for Electrical Equipment.
3. Transmission Line Protection and Safety Practices.
4. Circuit Breakers and Switchgear Selection and Application.

UNIT-I

Protective Relays, CTs and PTs: Classification **-** electromechanical, static, and numerical relays; construction, operating characteristic and their applications with limitations; over and under current, directional, differential, distance and other types of relay; constructions and characteristics of CTs and PTs, capacitance-voltage transformer.

UNIT-II

Protection of Generators and Motors: Differential Protection, protection of stator windings against short circuits, turn-to-turn faults and ground faults; rotor earth fault protection; protection against unbalanced loading, loss of excitation, loss of synchronism and prime mover failure; protection of motors (induction and synchronous) and bus bars.

Protection of Transformers: Protection against internal faults such as short circuits and turn-to-turn faults using differential and overcurrent relays, as well as protection for other abnormal conditions.

UNIT-III

Protection of Transmission lines: Over current protection, grading of overcurrent relays, distance protection, types of distance relays and their characteristics, carrier current protection; protection against surges, surge diverters, surge absorbers; use of ground wires on transmission lines; necessity of grounding system neutral and substation equipment, methods of grounding.

UNIT-IV

Switchgear: Types and applications of fuse and MCB; physics of arcing phenomenon and arc interruption, DC and AC circuit breaking, re-striking voltage and recovery voltage, rate of rise of recovery voltage, resistance switching, current chopping, interruption of capacitive current; circuit breakers and types – air-blast, air-break, oil, SF6 and vacuum circuit breaker, comparison of different circuit breakers, ratings and selection of circuit breakers.

Suggestive Readings:

1. J. J. Grainger, and W.D. Stevenson, "Power System Analysis", Tata McGraw-Hill, 2003.
2. Paul M. Anderson "Power System Protection" IEEE Press.
3. C L Wadhva, "Electrical Power System" Wiley Eastern Ltd., 3rd edition, 2000.
4. D.P. Kothari, and I.J. Nagrath "Modern Power System Analysis," Tata McGraw-Hill, 4th Edition.

List of Experiments:

1. Measure the high value of AC Voltage by a low-range AC Voltmeter and Potential Transformer.
2. Draw and study the operating characteristics of Miniature Circuit Breaker Type-B and Type-C.
3. Study the operating characteristics of the HRC fuse.
4. To study the IDMT Over the Current Relay & its applications.
5. To study and verify the operating characteristics of over current relay at various plug & time settings.
6. To study the performance of over voltage relay.
7. To study the performance of under voltage relay.
8. To study and verify the operating Characteristics of Earth Fault Relay with different plug settings.
9. Study and verify the operating characteristics of three phases over current and earth fault numeric relays with different time multiplier settings (TMS) and current Settings.
10. Unsymmetrical fault analysis of a single and three-phase transmission line.
11. Symmetrical fault analysis of three-phase transmission line.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Embedded System Technologies (DSC-17)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Embedded System Technologies	4	3	0	1	Fundamentals of Computer Programming, Analog and Digital Electronic Circuits

Course Hours: L-03, T-00, P-02

Course Objectives:

1. To understand the concepts of the Architecture of the 8085 microprocessors.
2. To understand the concepts of the Architecture of the 8051 microcontrollers.
3. To understand the design aspects of I/O and Memory Interfacing circuits.
4. To understand the architecture and programming of ARM processors.

Course Outcomes:

At the end of this course, students will have the ability to:

1. To design and implement programs on the 8085 microprocessor.
2. To design and implement programs on the 8051 microcontroller.
3. To design I/O circuits and Memory Interfacing circuits.
4. To design and develop components of the ARM processor.

UNIT-I

Microprocessors: 8085-architecture, operation, pin configuration and functions, bus organization, control signal generation for external operations- fetch, IO/M, read/write, machine cycles and bus timings. Addressing mode, instruction set, Overview/concept of peripheral interfacing devices- 8251, 8253, 8255 and 8279.

UNIT-II

Microcontrollers: 8051-architecture, operation, pin configuration and functions, memory organization, register, I/O ports, addressing modes, instruction sets, instruction classification. Assembly language programming, Interrupts in 8051. Timer/Counter programming for time delay generation and waveform generation. Interfacing with ADC, DAC, LEDs and seven-segment display.

UNIT-III

Introduction to Embedded Systems: Complex systems and microprocessors, Embedded system design process, Instruction sets preliminaries of ARM Processor, CPU: programming input and output supervisor mode, exceptions and traps, Co-processors, Memory system mechanisms, CPU performance.

UNIT-IV

Embedded Computing Platform Design and Optimization of CPU: Bus-Memory devices and systems, designing with computing platforms, platform level performance analysis, Components for embedded programs, Models of programs Assembly, linking and loading, compilation techniques-Program level performance analysis, Software performance optimization, Analysis and optimization of program size, Program validation and testing.

Suggestive Readings:

1. Ramesh S. Goankar, “8085 Microprocessors Architecture Application and Programming”, Penram International, 5th Edition.
2. Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay, “The 8051 Microcontroller and Embedded Systems: Using Assembly and C”, 2nd Edition, Pearson Education, 2011
3. Marilyn Wolf, “Computers as Components - Principles of Embedded Computing System Design”, 3rd Edition “Morgan Kaufmann Publisher (An imprint from Elsevier), 2012

List of Experiments:

1. To develop and execute a program on the 8085 microprocessor for arithmetic and logical operations relevant to electrical signal processing.
2. To implement data transfer and sorting algorithms on the 8086 microprocessor for efficient electrical system management.
3. To design and simulate a traffic light controller using the 8051 microcontroller and LED interfacing.
4. To program the 8051 microcontroller for generating pulse width modulation (PWM) signals for motor speed control.
5. To interface an ADC with the 8051 microcontroller to digitize an analog electrical signal and display the result.
6. To design and implement a real-time clock using timers on the 8051 microcontroller for power system monitoring.
7. To configure an ARM-based microcontroller for temperature monitoring using ADC and an LED-based alarm system.
8. To implement an ARM-based system for serial communication between electrical devices using UART.
9. To develop a program on the ARM processor for controlling a DC motor using timers and PWM signals.
10. To design an embedded system for energy measurement using an ARM microcontroller and interfacing it with an ADC and LED display.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Power Electronics (DSC-18)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Power Electronics	4	3	0	1	Introduction to Electrical and Electronics Engineering, Electrical Network Analysis, Mathematics-1

Course Hours: L-03, T-00, P-02

Course Objectives:

1. To outline the working of uncontrolled devices.
2. To introduce the basic theory of power semiconductor devices and passive components,
3. their practical application in power electronics.
4. To familiarize the operation principle of AC-DC, DC-DC, and DC-AC conversion circuits and their applications.
5. Analyzing power electronics circuits and understanding circuit operation by drawing output waveforms.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Explain the characteristics, functions, and applications of various power electronic devices such as SCR, Power Transistor, MOSFET, GTO, IGBT, and MCT.
2. Design and analyze single-phase and three-phase controlled rectifiers with different types of loads (R, R-L, R-L-E).
3. Understand the principles of chopper operation and design chopper circuits for various applications, including step-up and step-down choppers.
4. Analyze the operation of single-phase and three-phase inverters, and implement voltage control and harmonics reduction techniques.
5. Explain the principles and applications of AC voltage controllers and cycloconverters for various load types (R and RL).

UNIT-I

Introduction and Power Semiconductor Devices: Power Electronics: Scope and applications, Introduction to power electronics devices, SCR, Power Transistor, MOSFET, GTO, IGBT, MCT etc. Thyristor V-I and Gate Characteristics, Two transistor analogy of SCR, methods of triggering and commutation (A,B,C,D,E,F), ratings and protection of device, snubber circuits and safe operating area. Firing circuit.

UNIT-II

Phase Controlled Rectifiers: Principle of phase control, single phase half wave-controlled rectifiers with R, R-L, R-L-E load, single phase full wave-controlled converters, 2-pulse mid-point converters, 2-pulse half and fully controlled bridge converters with R, R-L, R-L-E load, Three phase uncontrolled & controlled rectifier, triggering schemes, flyback diode, effect of source inductance.

UNIT-III

DC – DC Converters: Basic Principle of step-down chopper operation with R-L Loads, control strategies-time ratio control and current limit control. Types of chopper circuits, four quadrant chopper, steady state time domain analysis of type a chopper, effect of source inductance, step up and step-down Chopper, chopper circuit design.

Inverters: Forced commutated inverters, single phase voltage source inverters, Half bridge inverter, full bridge inverter (with R and RL load), steady state analysis, voltage control in single phase inverters, 3-phase bridge inverters (with R and RL load) 120, 150 and 180 mode, pulse width modulated inverters, harmonics reduction techniques, current source inverter, inverter circuit design. (Voltage & frequency control).

UNIT-IV

AC Voltage Controllers and Cycloconverters: Principle of AC voltage controllers-phase control and integral cycle control, types of AC voltage controllers, single-phase and three-phase AC controllers with R and RL loads, fan and temperature control. **Cyclo-converter:** Principles of operation, advantages, disadvantages and applications of single/three cycloconverters on R and RL load.
Applications in power electronics: UPS, SMPS and Drives.

Suggestive Readings:

1. Lander C. W., “Power Electronics”, 3rd Ed., McGraw-Hill International Book Company
2. Mohan N., Undeland T. M. and Robbins W. P., “Power Electronics Converters, Applications and Design”, 3rd Ed., Wiley India.
3. Rashid M. H., “Power Electronics Circuits Devices and Applications”, 3rd Ed., Pearson Education.
4. Derek A. Paice “Power Electronic Converter Harmonics – Multipulse Methods for Clean Power”, IEEE Press, 1996.
5. P.C.Sen, “Modern Power Electronics ”, S. Chand and Co. Ltd., New Delhi, 2000.
6. Power Electronics”, P.S. Bimbhra, Khanna Pub.
7. Power Electronics Circuits and MATLAB simulations”, Alok Jain, Penram International Pub.(India) Pvt.Ltd.
8. Power Electronics Principles and Applications, Joseph vithayathil, McGraw Hill Education India P. Ltd
9. Power Electronics : Essentials & Applications, L. Umanand, Wiley India Pvt Ltd
10. “Power Electronics and Variable Frequency Drives”, Bimal K.Bose, IEEE Press.
11. “Power Electronics Systems: Theory and Design”, Jai P. Agrawal, Pearson Education Pvt.Ltd.

List of Experiments:

1. Characteristics of SCR, IGBT & Power MOSFET.
2. Analysis of Single-phase AC voltage controller with R & RL Loads.
3. Analysis of Single phase fully controlled bridge converter With R & RL Loads.
4. Analysis of Single phase IGBT inverter with R and R-L Loads.
5. Analysis of Three phase fully controlled bridge converter with R Load.
6. Analysis of Single-phase dual converter with RL load.
7. Analysis of Four quadrant operation of chopper with R-load.
8. Analysis of PWM control of Boost converter with R and R-L loads.
9. Simulation of Single-Phase ac to dc converter with LC filter in MATLAB.
10. Simulation of Single-phase inverter with current controlled PWM technique in MATLAB.
11. Analysis of single phase cyclo converter with R and R-L load.
12. Simulation of Single phase fully controlled PWM rectifier with R & RL loads using PSCAD.
13. Generation of PWM pulses using microcontroller kit.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Detailed Syllabus of Discipline Specific Elective (DSE) courses for B.Tech. (EE) – SEMESTER VI

Digital Control System (DSE-4)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Digital Control System	4	3	0	1	Introduction to Electrical and Electronics Engineering, Mathematics-I, Electrical Network Analysis/ Network Analysis and Synthesis

Course Hours: L-03, T-00, P-02

Course Objectives:

1. To acquire the knowledge of digital control systems.
2. To understand the modelling of discrete-time systems, time response analysis and stability analysis method.
3. To design digital control systems with deadbeat response.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Apply the concepts of signal processing and digital control.
2. Apply z-transformation for the digital control system.
3. Design and analyze the digital control system for various physical systems.

UNIT-I
Introduction to Digital Control Systems
Discrete-time system representation, mathematical modelling of the sampling process.

UNIT-II
Modelling Discrete-Time Systems by Pulse Transfer Function
Z-transform, mapping of s-plane to z-plane, pulse transfer function, related examples.

UNIT-III
Stability Analysis
Jury stability test, stability analysis using bi-linear transformation, related examples.
Time Response Analysis
Transient and steady-state responses, time response parameters of a second order system, related examples.

UNIT-IV
Design of Sampled Data Control Systems
Root locus method, controller design using root locus, Nyquist stability criteria, bode plot, lead compensator design, lag compensator design, lag-lead compensator design in the frequency domain.

Deadbeat Response Design
Design of digital control systems with deadbeat response, practical issues with deadbeat response design, sampled data control systems with deadbeat response.

Suggestive Readings:

1. Digital Control and State variable methods by M Gopal, Tata McGraw-Hill publishing company limited.
2. Discrete-Time Linear Systems: Theory and Design with Applications by G Gu, Springer Science & Business Media.
3. Discrete Time Control Systems by K Ogata.
4. Discrete Control Systems by Y Okuyama, Springer London.
5. Advanced Discrete-Time Control by K Abidi, J X Xu, Springer Singapore.

List of Experiments:

1. Discrete time state space modelling for the SISO system.
2. Discrete time state space modelling for MIMO system.
3. Time response analysis of the SISO discrete-time system.
4. Time response analysis of MIMO discrete-time system.
5. Stability analysis of SISO discrete-time system.
6. Stability analysis of MIMO discrete-time system.
7. Design of lead compensator for the discrete-time system.
8. Design of lag compensator for the discrete-time system.
9. Design of a Lag-lead compensator for a discrete-time system.
10. Design of digital control systems with deadbeat response.
11. Design of Root locus for the discrete-time system.
12. Implementation of Bode plot for discrete-time system.
13. Implementation of Nyquist criteria on discrete-time system.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Digital Signal Processing (DSE-4)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Digital Control Processing	4	3	1	0	Introduction to Electrical and Electronics Engineering, Mathematics-I, Electrical Network Analysis/ Network Analysis and Synthesis

Course Hours: L-03, T-01, P-00

Course Objectives:

1. To acquire knowledge of digital signal processing for various process controls, signal and signal processing.
2. To understand the time domain representation of digital signals.
3. To learn the concepts of digital filter designs.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Acquire knowledge of digital signal processing for various process controls.
2. Learn thoroughly signal and signal processing.
3. To formulate the time domain representation, transformation, filtered design etc. for their projects and research applications.

UNIT-I

Signal & Signal Processing: Classification of signals, typical signal processing operations, typical signal processing applications, why digital signal processing.

Time Domain Representation of Signals & Systems: Discrete-time signals, operations on sequences, the sampling process, discrete-time systems, time-domain characters tics of LTI discrete-time systems, state space representation of LTI discrete-time systems.

UNIT-II

Transformations: Domain representation of signals: the discrete-time Fourier transform, discrete Fourier transform, computation of the DFT of real sequences, linear convolution using the DFT, z-transform, inverse z-transform.

UNIT-III

Time Domain Representation of LTI Systems: Frequency response, transfer function. Digital two-pair stability test.

Digital Processing of Continuous Time – Signals: Sampling of continuous-time signals, analysis filter design, anti-aliasing filter design, and reconstruction filter design**.**

UNIT-IV

Digital Filter Structures

Block diagram representation, signal flow graph representation, equivalent structures, Basic FIR digital filter structures, Basic IIR filter structures, all-pass filters, and tunable structures**.**

Digital Filter Design: Preliminary conditions, impulse invariance method of IIR filter design, bilinear transform method of IIR filter design, design of filter IIR notch filters, FIR filter design based on truncated Fourier series, FIR filter design based on frequency sampling approach, computer-aided design of digital filters.

Suggestive Readings:

1. Digital Signal Processing by Sanjit K. Mitra, Tata McGraw Hill
2. Digital Filters: Analysis & Design by A. Antoniou, McGraw Hill book company
3. Digital Signal Processing by S.D. Sterms, Prentice Hall Inc

Department of Electrical Engineering
Faculty of Technology
University of Delhi

List of Discipline Specific Elective (DSE)/ Generic Elective (GE) courses offered for Minors / Specializations by the Department of Electrical Engineering in Third Year

1. **Minor in EE (Offered to ECE and CSE)**
    a. DSE-3/ GE-5: Electrical Measurement Technique
    b. DSE-4/ GE-6: Power Converters and Applications
2. **Minor/Specialization in Robotics and Automation (Offered to EE, ECE, and CSE)**
    a. DSE-3/ GE-5: Introduction to Robotics and Mechatronics
    b. DSE-4/ GE-6: Industrial Automation
3. **Minor/Specialization in Sustainable Energy Engineering (Offered to EE, ECE, and CSE)**
    a. DSE-3/ GE-5: Energy Conservation and Audit
    b. DSE-4/ GE-6: Energy Policies for Sustainable Development
4. **Minor/Specialization in Electric and Hybrid Vehicle (Offered to EE, ECE, and CSE)**
    a. DSE-3/ GE-5: Electrical Storage and Management System
    b. DSE-4/ GE-6: Power Electronics Converters and Drives for Electric Vehicles

Detailed Syllabus of Generic Elective (GE) courses offered for Minors / Specializations by Department of Electrical Engineering in SEMESTER V

Electrical Measurement Techniques (DSE-3/ GE-5)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Electrical Measurement Techniques	4	3	0	1	Introduction to Electrical and Electronics Engineering, Physics

Course Hours: L-03, T-00, P-02

Course Objectives:

1. To make students familiarize themselves with different types of error in measurements.
2. To analyze the constructions of different analog type measuring instruments.
3. To understand the working of different power, energy, resistance, inductance and capacitance measurement techniques.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Classify various measuring instruments used to measure electrical quantities.
2. Apply methods for the measurement of resistance, capacitance and inductance.
3. Choose the suitable current and potential transformers.
4. Measure and analyze the error free currents and voltages.

UNIT-I

Preliminaries: Concepts of Measurements & Measurement Systems: Introduction to measurement and instrumentation, S. I. system, methods of measurement, static and dynamic characteristics of instruments, definitions – true value, accuracy, error, precision, sensitivity, resolution etc.

Analog Instruments: Classification of analog instruments, principle of operation, operating forces, errors in ammeters and voltmeters. Permanent magnet moving coil, moving iron, dynamometer type, induction type, electrostatic type instruments.

UNIT-II

Potentiometers: Principle of D. C. potentiometer, direct reading potentiometers, accurate forms of potentiometers, A. C. potentiometer principle, polar and Co - ordinate type A. C. potentiometer, applications of A. C. and D. C. potentiometers.

Measurement of Power and Energy: Electrodynamometer type wattmeter, measurement of power in three phase circuits, three phase wattmeter, measurement of reactive power, energy meter for A.C. circuits, induction type energy meter.

UNIT-III

Measurement of Resistance: Measurement of low, medium & high resistances, insulation resistance measurement, localization of cable fault, Loop tests.

Measurement of Inductance and Capacitance: A. C. bridges for inductance measurement – Maxwell, Hays, Anderson and Owen bridges, capacitance measurement – De Sauty and Schering Bridge. Measurement of frequency by Wien's bridge.

UNIT-IV

Magnetic Measurements: Magnetic measurement using Ballistic Galvanometer, Grassot Flux meter, BH curve of magnetic material, separation of losses.

Instrument Transformers: Current and Potential transformers, ratio and phase angle errors, design considerations, numerical problem.

Suggestive Readings:

1. Introduction to Modern Electronic Instrumentation and Measurement Techniques: Helfrick and Cooper, Prentice Hall of India, 1997.
2. Instrumentation Measurement and Feedback: Jones, B. E., Tata McGraw-Hill, 1995.
3. Electrical Measurement and Measuring Instruments: Golding, E. W., Sir Issac Pitman & Sons., 3rd Edition.
4. A course in Electrical and Electronic Measurement and Instrumentation: A. K. Sawhney, Dhanpat Rai Publication.

List of Experiments:

1. Calibration of voltmeter and Ammeter.
2. Calibration of single phase A.C. Energy meter.
3. Three phase power measurement by two wattmeter method.
4. Measurement of reactive power using single wattmeter in three-phase circuit.
5. Measurement of percentage ratio error and phase angle of given C.T. by Silsbee's method.
6. Extension of instrument ranges using C.T. and P.T.
7. Measurement of resistance using Kelvin's Double Bridge.
8. Measurement of earth resistance using Meggar.
9. Measurement of self-inductance and Quality factor using Anderson Bridge.
10. Measurement of capacitance using Schering Bridge.
11. Measurement of voltage, current and resistance using DC potentiometer.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Introduction to Robotics and Mechatronics (DSE-3/ GE-5)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Introduction to Robotics and Mechatronics	4	3	0	1	Introduction to Electrical and Electronics Engineering

Course Hours: L-03, T-00, P-02

Course Objectives:

1. To acquire the knowledge of fundamentals of mechatronics.
2. Impart knowledge and information about product design.
3. Development and control of intelligent systems for all aspects of life.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. To understand the components of mechatronic system,
2. To design product and systems theoretically as well as practically with intelligence.

UNIT-I

Understanding Mechatronics : Mechatronics System, Evolution, Definitions of Mechatronics, Key Elements of Mechatronics, Mechatronics for all Civil, Metallurgical, Aerospace, Chemical, Architecture, Medical, Robotics, Defense, Agriculture, etc., Role of Mechanical, Electrical, Electronics, Computer Engineers in Intelligent Product and Process Design, Development and Control, Bio-mechatronics.

UNIT-II

Systems and Machines: System, Classification of System, Mechanistic System Classification Based on Input Energy, Mathematical Model and Function, Machine, Parts of Machine, Concepts of Machine, Classification of Machines based on Function and Size.

System Intelligence: Properties of Intelligent System, System Intelligence Levels, Human Intelligence System, Future Generation System Intelligence Level, Expressing System Intelligence.

UNIT-III

Sensor and Transducer: Sensors in Mechatronics System, Difference between Sensors and Transducers, Classification of Sensors, Based on Sensor Output Signal, Sensor Input Physical Parameters, Sensor Accuracy (Smart/Intelligent Sensor), Performance Terminology, Static Characteristics, Dynamic Characteristics.

Signal Conditioning Devices: Signal Conditioning Processes, Application of Signal Conditioning Devices in Mechatronics based on Their Characteristics such as Diode, Transistor, SCR, DIAC, TRIAC, Op-Amps, Signal Filtering, Circuit Protection, Signal Conversion, ADC and DAC, Logic Gates, Flip-Flops, Register, Counters.

Actuators: Actuators, Types of Actuators, Mechanical Actuation System (i.e. Linear-rotary, Rotary-linear Mechanism, Gear, Bearing, Pulley etc.). Electrical Actuation System (DC, AC, Stepper Motors), Pneumatic and Hydraulic Actuation System.

UNIT-IV

Controllers: Microprocessor, Microcontroller, PLC Controller & Their Architectures, Principles and Working Software Programs (Assembly/High Level), Interfacing Aspects, Application Examples.

Robotics and Automation:

Evolution of Robots, Definitions, Types of Motions, Function, Governing Laws, Classification, Features and Components of Robots, System Automation.

Suggestive Readings:

1. Mechatronics First edition by Tilak Thakur, published by Oxford University Press
2. Mechatronics, fourth edition by W Bolton. ISBN 978-81-317-3253-3
3. Dan Necsulescu Mechatronics published by Pearson Education (Singapore) Pvt. Ltd.
4. Book by H M T Limited, Mechatronics Tata McGraw Hill Publishing Company Limited, New Delhi.
5. Mechatronics Principles, Concepts & Applications by Nitaigour P Mahalik published by TMH

List of Experiments:

1. ***Experiment on Sensors & Transducers:***
   i. To study the characteristics of LVDT using linear displacement trainer Kit & compare with ideal characteristics.
   ii. To measure the strain of the metal strip using strain gauge trainer kit & compare with ideal characteristics.
   iii. To measure the angular displacement of resistive & capacitive transducer using angular displacement trainer kit & compare with ideal characteristics.
   iv. To obtain the characteristics of RTD, Thermistor, thermocouple with hot and cold junction thermal trainer kit & compare with ideal characteristics.
2. ***Experiments on Signal Conditioning***:
   i. PN Junction Diode
   ii. Zener Diode
   iii. Half wave rectifier
   iv. Full wave rectifier
3. ***Experiments on Digital devices:***
   i. Logic Gates (AND, OR, NAND, NOR etc)
   ii. Flip Flop (RS Flip Flop), D Flip Flop.
4. ***Experiments on Controller:***
   i. Study of microprocessors, microcontroller, programmable logic controller (PLC).
   ii. PLC interfacing of I/O and I/O addressing.
   iii. To perform any basic sequence programming using PLC.
5. ***Experiments on Actuators:***
   i. Study of mechanical, electrical, hydraulic/pneumatic actuators.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Energy Conservation and Audit (DSE-3/ GE-5)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Energy Conservation and Audit	4	3	1	0	Introduction to Electrical and Electronics Engineering, Mathematics-I

Course Hours: L-03, T-01, P-00

Course Objectives:

6. To understand basic principles of energy audit and management.
7. To describe lighting modification of existing systems.
8. To examine power factor improvement measures and energy instruments.
9. To analyze space conditioning systems.
10. To evaluate the energy management strategies.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

5. Ability to Conduct Energy Audits.
6. Understanding of Energy Management Principles.
7. Skill in Lighting and Power Systems Optimization.
8. Proficiency with Energy Measurement Instruments.
9. Economic and Financial Analysis of Energy Projects.

UNIT-I

Principles of Energy Audit and Management: Energy audit: Definitions, Concept, Types of audit, Energy index, Cost index, Pie charts, Sankey diagrams, Load profiles. Energy conservation schemes and energy saving potential, Principles of energy management: Initiating, planning, controlling, promoting, monitoring, reporting.

UNIT-II

Lighting Modification of existing systems, Replacement of existing systems, Priorities: Definition of terms and units, Luminous efficiency, Polar curve, Calculation of illumination level, Illumination of inclined surface to beam, Luminance or brightness, Types of lamps, Types of lighting, Electric lighting fittings (luminaries), Floodlighting, White light LED and conducting Polymers, Energy conservation measures.

UNIT-III

Power Factor and energy instruments: Power factor, Methods of improvement, Location of capacitors, Power factor with nonlinear loads, Effect of harmonics on Power factor. Energy Instruments: Watt-hour meter, Data loggers, Thermocouples, Pyrometers, Lux meters, Tong testers, Power analyzer.

Space Heating and Ventilation: Air–Conditioning (HVAC) and Water Heating: Introduction, Heating of buildings, Transfer of Heat, Space heating methods, Ventilation and air–conditioning, Cooling load, Electric water heating systems, Energy conservation methods.

UNIT-IV

Economic Aspects and Analysis Economics Analysis: Depreciation Methods, Time value of money, Rate of return, Present worth method, Replacement analysis, Life cycle costing analysis, Energy efficient motors (basic concepts).

Computation of Economic Aspects Calculation of simple payback method, Net present worth method, Power factor correction, Applications of life cycle costing analysis, Return on investment.

Suggestive Readings:

1. W.R. Murphy & G. Mckay, "Energy management," Butter worth Elsevier publications. 2012
2. John. C. Andreas, "Energy-Efficient Electric Motors," Taylor & Francis, 2018.
3. S C Tripathy, "Electric Energy Utilization and Conservation", Tata McGraw hill, 1991.
4. Paul o' Callaghan, "Energy management," Mc–Graw Hill, 1998.
5. W.C. Turner, "Energy management handbook," John wiley and sons.
6. K. V. Sharma, P. Venkataseshaiah, "Energy Management and Conservation," I K International Publishing House, 2020.

Electrical Storage and Management System (DSE-3/ GE-5)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Electrical Storage and Management System	4	3	0	1	Introduction to Electrical and Electronics Engineering, Physics, Introduction to Electric and Hybrid Vehicles

Course Hours: L-03, T-00, P-02

Course Objectives:

1. To understand the different types of energy storage systems.
2. To study the battery characteristics & parameters.
3. To model the types of batteries
4. To know the concepts of the battery management system and design the battery pack.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Analyse different types of batteries.
2. Describe the battery characteristics & parameters.
3. Apply the concepts of a battery management system and design the battery pack.
4. Explain battery testing, disposal, and recycling.

UNIT-I

Energy Storage Technologies: Classification of Storage Technologies by Energy type- Thermal Energy: Heat Storage; Chemical Energy: Organic and Non- Organic; Mechanical Energy: Kinetic and Potential Energy; Electrical Energy: Electrical Potential.

Energy Storage Systems (ESS) in Modern Electrical Systems: Lead-acid batteries, Nickel-cadmium batteries, Lithium-ion batteries, Sodium-sulphur batteries, Nickel metal hydride batteries, Capacitors and Supercapacitors. Solid state Batteries. Differences amongst different ESS.

UNIT-II

Typical ESS and Battery Chemistry: Electrodes, Electrolytes, Collectors, Thermal management, Packaging of battery pack Lithium-based batteries: Lithium manganese oxide, Lithium iron phosphate, Lithium nickel manganese cobalt oxide, Lithium nickel cobalt aluminium oxide and Lithium titanate; Silicon Batteries, Sodium-sulphur Batteries, Proton Batteries, Graphite Dual-Ion Batteries, Salt-water Batteries and Potassium-Ion Batteries**.**

The development cycle of Batteries: ESS sizing, Electrical, Mechanical and Thermal Design, BMS Software and Hardware development, Prototype development, System Validation, Lab Testing, Safety test and Certification.

UNIT-III

Battery Management Systems (BMS): Introduction to BMS, Objectives of the BMS: Discharging control, Charging control, State-of-Charge Determination, State-of-Health Determination, Cell Balancing; BMS topologies: Distributed Topology, Modular Topology and Centralized Topology, Firmware development, Certification, Aging.

UNIT-IV

Batteries for the EV application: Performance criterion for EV batteries, Energy density, Amp hour density, Energy efficiency, Cost, Operating temperature, number of life cycles, recharge and self-discharge rates and commercial availability, some reference batteries and extension to nonautomotive sectors.

Chemical & structure material properties for cell safety and battery design, battery testing, limitations for transport and storage of cells and batteries, Recycling, disposal and second use of batteries. Battery Leakage: gas generation in batteries, leakage path, leakage rates. Ruptures: Mechanical stress and pressure tolerance of cells, safety vents, Explosions: Causes of battery explosions, explosive process, Thermal Runway: High discharge rates, Short circuits, charging and discharging. Environment and Human Health impact assessments of batteries, General recycling issues and drivers, methods of recycling of EV batteries.

Suggestive Readings:

1. Alfred Rufer, "Energy Storage systems and components", CRC Press 2017
2. Tom Denton, "Automotive Electrical and Electronic Systems", 5th Edition, Routledge 2018.
3. Mehard Ehsani, Yiming Gao, Stefano longo and Kambiz Ebrahimi, "Modern Electric, Hybrid Electric, and Fuel Cell Vehicles", CRC Press, 3rd Edition. 2019.
4. Iqbal Husain, "Electric and Hybrid Vehicles: Design Fundamentals", CRC Press. 2021
5. K. T. Chau, "Energy Systems for Electric and Hybrid Vehicles," IET Transportation Series 2 2016.
6. Jiuchun Jiang and Caiping Zhang, "Fundamentals and Applications of Lithium–Ion Batteries in Electric Drive Vehicles," John Wiley & Sons 2015.
7. Pistoia, J.P. Wiaux, S.P. Wolsky, Used Battery Collection and Recycling, Elsevier, 2001.
8. Chris Mi, Abul Masrur& David Wenzhong Gao, Hybrid electric Vehicle- Principles & Applications with Practical Properties, Wiley, 2011.
9. Arno Kwade, Jan Diekmann, Recycling of Lithium-Ion Batteries: The LithoRec Way, Springer, 2018.
10. Ibrahim Dinçer, Halil S. Hamut and Nader Javani, Thermal Management of Electric Vehicle Battery Systems, JohnWiley& Sons Ltd., 2016.

List of Experiments:

1. Study the basic parameters of battery
2. Measure the charging voltage and current of a given battery.
3. Demonstrate any charging technique of lead acid battery/Lithium Ion battery.
4. Efficiency Analysis: Evaluate charge/discharge efficiency at different C-rates. State of Charge (SOC) Estimation: Perform tests to estimate SOC using voltage and current data.
5. Study of ratings of battery for e-cycle, 2W EVs, Erickshaws, E-CARsetc
6. Study the process of battery testing and measure the parameters of the battery.
7. Study and Demonstration of Battery Temperature Measurement / thermal safety issues (Thermocouple, Thermistor etc)
8. Battery pack design for given EV application (Testing Various series parallel combinations for given application)
9. Voltage and Current Ripple Testing: Study the effect of DC-DC converter integration on battery pack voltage and current quality. Fast Charging Analysis: Study the impact of fast charging on cell and pack performance.
10. Battery Recycling Efficiency: Study cell disassembly and material recovery for end-of-life batteries.
11. BMS Functional Testing and its integration with EV battery: Validate the basic functionalities of the Normal & Smart BMS, including voltage and current monitoring and how to perform its connection with EV Battery and other auxiliary systems.
12. Communication Protocol Testing (CAN Bus): Verify data communication between the BMS and external systems using CAN protocol.

13. CAN Data Analysis: Analyze and interpret CAN bus data from the BMS for insights into system performance. Data Logging and Monitoring: Record and analyze real-time data from the battery system during operation.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Detailed Syllabus of Generic Elective (GE) courses offered for Minors / Specializations by Department of Electrical Engineering in SEMESTER VI

Power Converters and Applications (DSE-4/ GE-6)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Power Converters and Applications	4	3	0	1	Introduction to Electrical and Electronics Engineering

Course Hours: L-03, T-00, P-02

Course Objectives:

1. To outline the working of uncontrolled devices.
2. To introduce the basic theory of power semiconductor devices and passive components and their practical application in power electronics.
3. To familiarize the operation principle of AC-DC, DC-DC, DC-AC conversion circuits and their applications.
4. Analyzing power electronics circuits and understanding circuit operation by drawing output waveforms.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Explain the characteristics, functions, and applications of various power electronic devices such as SCR, Power Transistor, MOSFET, GTO, IGBT, and MCT.
2. Design and analyze single-phase and three-phase controlled rectifiers with different loads (R, R-L, R-L-E).
3. Understand the principles of chopper operation and design chopper circuits for various applications, including step-up and step-down choppers.
4. Analyze the operation of single-phase and three-phase inverters and implement voltage control and harmonics reduction techniques.
5. Explain the principles and applications of AC voltage controllers and cycloconverters for various load types (R and RL).

UNIT-I

Solid State Power Devices: Principle of operation of SCR, dynamic characteristic of SCR during turn ON and turn OFF, parameters of SCR, dv/dt and di/dt protection, snubber circuit, commutation circuits, Heat sink design.

Modern Power Devices: Principle of operation of MOSFET, IGBT, GTO, MCT, SIT, SITH, IGCT, and their operating characteristics.

UNIT-II

Single-phase Converter: Half wave converter, 2-pulse midpoint converter, half-controlled and fully controlled bridge converters, input current and output voltage waveforms, the effect of load and source impedance, expressions for input power factor, displacement factor, harmonic factor and output voltage, the effect of the free-wheeling diode, triggering circuits.

UNIT-III

DC-DC Converters: Principle of operation of single quadrant chopper, continuous and discontinuous modes of operation; Voltage and current commutation, design of commutating components; Introduction to SMPS.

UNIT-IV

Inverters: Voltage source and current source inverters, Principle of operation of single-phase half bridge and full bridge voltage source inverters, voltage and current waveforms; Three-phase bridge inverter, 120^0 and 180^0 modes of operation, voltage and current waveforms with star and delta connected RL load; Voltage and frequency control of inverters.

Suggestive Readings:

1. Lander C. W., "Power Electronics", 3rd Ed., McGraw-Hill International Book Company
2. Mohan N., Undeland T. M. and Robbins W. P., "Power Electronics Converters, Applications and Design", 3rd Ed., Wiley India.
3. Rashid M. H., "Power Electronics Circuits Devices and Applications", 3rd Ed., Pearson Education.
4. Derek A. Paice "Power Electronic Converter Harmonics – Multipulse Methods for Clean Power", IEEE Press, 1996.
5. P.C.Sen, "Modern Power Electronics ", S. Chand and Co. Ltd., New Delhi, 2000.
6. Power Electronics", P.S. Bimbhra, Khanna Pub.
7. Power Electronics Circuits and MATLAB simulations", Alok Jain, Penram International Pub.(India) Pvt.Ltd.
8. Power Electronics Principles and Applications, Joseph vithayathil, McGraw Hill Education India P. Ltd
9. Power Electronics : Essentials & Applications, L. Umanand, Wiley India Pvt Ltd
10. "Power Electronics and Variable Frequency Drives", Bimal K.Bose, IEEE Press.
11. "Power Electronics Systems: Theory and Design", Jai P. Agrawal, Pearson Education Pvt.Ltd.

List of Experiments:

1. To study various static switches (SCR, TRIAC, DIAC, IGBT and MOSFET) and their control.
2. To study R and RC-based triggering circuits for thyristors.
3. Design a relaxation oscillator circuit using a Unijunction Transistor (UJT) to be used as a firing circuit for single-phase phase-controlled rectifiers.
4. To study the phase control of TRIAC using DIAC & RC circuits.
5. To study single-phase half-controlled rectifier configurations for R and RL loads.
6. To study single-phase half-controlled rectifier configurations for RLE loads.
7. To study single-phase fully controlled rectifier configurations for R and RL loads.
8. To study three-phase fully controlled rectifier configurations for R and RL loads.
9. To study the working of step-up chopper.
10. To study the working of 180 degree inverter circuit.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Industrial Automation (DSE-4/ GE-6)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Industrial Automation	4	3	0	1	Sensors and Transducers, Introduction to Robotics and Mechatronics

Course Hours: L-03, T-00, P-02

Course Objectives:

1. To introduce the importance of automation techniques for process industries.
2. To impart the role of PLC in industry automation.
3. To expose to various control techniques employed in process automation.
4. To develop automation systems for process industries.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Familiar with various automation technologies in process industries.
2. Understand various automation tools and methods in the process industry.
3. Implement various control and automation methods in process industries.
4. Familiar with various communication technologies in manufacturing and process industries.

UNIT-I

Introduction: Physical Process, Types of Industrial Processes, Industry Classification, Process Automation System, Needs Met by automation, Advantages of Automation, Steps of Automation, Process Signals, Main Components of Industrial Automation.

Types Of Automation Systems: Localized Process, Distributed Process, Supervisory Control and Data Acquisition.

UNIT-II

Programmable Logic Controller (PLC): Block diagram of PLC, Programming languages of PLC, Basic instruction sets, Design of alarm and interlocks, Networking of PLC, Overview of safety of PLC with case studies. Process Safety Automation: Levels of process safety through use of PLCs, Integrating Process safety PLC and DCS, Application of international standards in process safety control**.**

Fundamental PLC wiring diagram, relays, switches, transducers, sensors, seal-in circuits. Fundamentals of logic, Program scan, Relay logic, PLC programming languages, Digital logic gates, Boolean algebra PLC programming.

UNIT-III

Introduction to Computer based Industrial Automation: Direct Digital Control (DDC), supervisory control and data acquisition (SCADA) based architectures. SCADA for process industries includes understanding of RTUs, Pumping stations, Evacuation processes, Mass Flow Meters and other flow meters, Leak-flow studies of pipelines, Transport Automation.

SCADA Systems Software and Protocols: The components of a SCADA system, SCADA software package, Specialized SCADA Protocols, Error Detection, Distributed Network Protocols, New technologies in SCADA systems.

UNIT-IV

Distributed Control System (DCS): Local Control Unit (LCU) architecture, LCU Process Interfacing Issues, Block diagram and Overview of different LCU security design approaches, Networking of DCS. Introduction to communication protocols- Profibus, Field bus, HART protocols. Data gathering, Data analytics, Real-time analysis of data stream from DCS, Historian build, Integration of business inputs with process data, Leveraging RTU (as different from PLCs and DCS).
Automation System Functionalities and Application Areas: Major Functionalities like Data Acquisition, Data Supervision or Monitoring, Process Survey, Process Control, Process Studies, Human Interaction, Data Logging and History Generation, Data Exchange, Data Availability, Application Areas of Automation System.

Suggestive Readings:

1. John W. Webb and Ronald A. Reis, "Programmable Logic Controllers: Principles and Applications", 5th Edition, Prentice Hall Inc., New Jersey, 2003.
2. Krishna Kant, "Computer - Based Industrial Control", 2nd Edition, Prentice Hall, New Delhi, 2011.
3. Frank D. Petruzella, "Programmable Logic Controllers", 5th Edition, McGraw- Hill, New York, 2016.
4. Curtis D. Johnson, "Process Control Instrumentation Technology", 8th Edition, Pearson New International, 2013.
5. Lukas M. P., "Distributed Control Systems", Van Nostrand Reinhold Co., New York, 1986.

List of Experiments:

1. Feed forward and ratio controller design for real time process trainer.
2. Development of combinational and sequential logic application using minimum PLC languages.
3. Development of Ladder logic program for control of real time processes.
4. Development of SCADA for a control of real time processes.
5. Study of HART and Field bus protocol.
6. P&I diagram development using simulation software for complex processes.
7. Study of Distributed Control System and different instruction sets.
8. Development of Cascade, ratio and feedback controller using DCS simulation software.
9. Development of HMI and annunciator circuits using DCS simulation software.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Energy Policies for Sustainable Development (DSE-4/ GE-6)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Energy Policies for Sustainable Development	4	3	1	0	Introduction to Electrical and Electronics Engineering

Course Hours: L-03, T-01, P-00

Course Objectives:

1. Understand the Legal and Regulatory Framework
2. Analyze India's Energy Resources and Consumption Patterns
3. Examine the Global Energy Context
4. Evaluate Energy Policies and Security
5. Address Future Energy Challenges

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. In-depth Understanding of Energy Acts
2. Comprehensive Knowledge of Indian Energy Resources
3. Global Energy Awareness
4. Energy Policy and Security Expertise
5. Problem-Solving for Future Energy Needs

UNIT-I

Energy Conservation: Energy Conservation Act-2001 and its features, Electricity Act – 2003 and its features, Framework of Central Electricity Authority (CEA), Central & States Electricity Regulatory Commissions (CERC & ERCs), Role of MoP (Ministry of Power)-BEE (Bureau of Energy Efficiency).

UNIT-II

Indian Energy Scenario: Energy resources & Consumption, Commercial and non-commercial forms of energy, Fossil fuels, Renewable sources in India, Sector-wise energy Consumption, Impact of energy on the economy, Need for use of new and renewable energy sources-present status and future of nuclear and renewable energy-Energy, Policy Issues related, Fossil Fuels-Renewable, Energy-Power sector reforms, restructuring of energy supply sector, energy strategy for future.

UNIT-III

Global Energy Scenario: Role of energy in economic development and social transformation, Energy and GDP - GNP and its dynamics, Energy sources, overall Energy demand and availability, Energy consumption in various sectors and its changing pattern, Depletion of energy sources and impact economics on international relations.

UNIT-IV

Indian Energy Policy: Global Energy Issues, National & State Level Energy Issues, National & State Energy Policy, Industrial Energy Policy, Energy Security, Energy Vision, Energy Pricing and Impact of Global Variations, Energy Productivity (National & Sector wise productivity).

Global Energy Policy: International Energy Polices of G-8 Countries, G-20 Countries, OPEC Countries, EU Countries, International Energy Treaties (Rio, Montreal and Kyoto), INDO-US Nuclear Deal, Future Energy Options, Sustainable Development, Energy Crisis, Role of International Energy Agency.

Suggestive Readings:

1. Mohan Munasinghe, Peter Meier, "Energy Policy analysis and Modelling," Cambridge University Press1993
2. J. Goldemberg, T.B. Johansson, A.K.N. Reddy and R.H. Williams, "Energy for a Sustainable World," Wiley Eastern, 1990.
3. P. Meier and M. Munasinghe, "Energy Policy Analysis & Modeling," Cambridge University Press, 1993.
4. Charles E. Brown, "World Energy Resources," Springer 2002.
5. Resources, Charles E. Brown, "International Energy Outlook" EIA Annual Publication.
6. A.W. Culp, "Principles of Energy Conversion," McGraw Hill International edition, BEE Reference book: no. 1/2/3/4.
7. S Rao, "Energy Technology," Khanna Publishers.

Power Electronics Converters and Drives for Electric Vehicles (DSE-4/ GE-6)

(Credit Distribution and Pre-Requisites of the Course)

Course title	Credits	Credit distribution of the course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Power Electronics Converters and Drives for Electric Vehicles	4	3	0	1	Introduction to Electrical and Electronics Engineering, Electric Vehicle Motor

Course Hours: L-03, T-00, P-02

Course Objectives:

1. To provide a comprehensive understanding of semiconductor devices and their application in power electronics for electric vehicle (EV).
2. To understand the fundamentals of DC-DC converters, grid-connected converters, and their role in bidirectional energy flow and V2X (Vehicle-to-Everything) applications.
3. To introduce the principles and modelling of advanced motor drives such as SRM, BLDC, and PMSM, with emphasis on field-oriented and direct torque control.
4. To explore the applications of high-power and high-speed motor drives in EVs, including power converter design, special PWM techniques, and field-oriented control.

Course Outcomes:

At the end of this course, students will demonstrate the ability to:

1. Design and model DC-DC converters for grid-connected and bidirectional energy flow in EV systems, including isolated and non-isolated types.
2. Classify different types of EV chargers and understand their operational principles and impact on EV charging infrastructure.
3. Demonstrate proficiency in controlling induction motor drives using open-loop V/f, vector control techniques, and slip recovery methods for various applications.
4. Model and implement control techniques for switched reluctance motors (SRM), brushless DC (BLDC) motors, and permanent magnet synchronous motors (PMSM) in EV systems.
5. Apply advanced field-oriented control and PWM techniques for high-power and high-speed motor drives in electric vehicles, ensuring efficient energy conversion and performance optimization.

UNIT-I

Review of semiconductor devices; turn-on and turn-off characteristics; loss computation in semiconductor devices; basics of nonisolated/isolated DC-DC and grid connected converters; classification of EV chargers; modelling and control of bi-directional DC-DC converters; discussions on V2X applications.

UNIT-II

Induction Motor Drives: Basics of induction motor; open-loop v/f control; basic pulse width modulation techniques; vector control of IM drives of IM drives for different applications, VSI and CSI fed IM drives, vector controlled permanent magnet induction machines, slip recovery and stator emf injection method, vector control of wound rotor Induction machines.

UNIT-III

SRM, BLDC and PMSM Drives: Basics of switched reluctance motor, BLDC motor and PMSM motors; Basics modelling of SRM, BLDC and PMSM drives, Field oriented control and direct torque control of these drives.

UNIT-IV

High-power and High-speed EVs: Applications of High-power induction motor drives; power converter design; special PWM techniques for high-power applications; field-oriented control of high-power IM drives; applications of highspeed PMSM drives; power converter design and PWM techniques; field-oriented control of high-speed PMSM drives.

Suggestive Readings:

1. Ali Emadi, Advanced Electric Drive Vehicles, CRC Press (2015)
2. Iqbal Husain, Electric and Hybrid Vehicles – Design Fundamentals, Second Edition, CRC Press (2011).
3. W. Leonard, Control of Electric Drives, Springer Press (2007).
4. R Krishnan, Permanent Magnet Synchronous and Brushless DC Motor Drives", CRC Press (2010).
5. Berker B., James W. J. & A. Emadi, Switched Reluctance Motor Drives, CRC Press (2019).
6. Bin Wu, High-Power Converters and Ac Drives, IEEE WILEY Press (2017).
7. Bimal K. Bose, Modern Power Electronics and AC Drives,Prentice Hall PTR (2001).

List of Experiments:

1. To familiarize with the basic vector control of Permanent Magnet Synchronous Motor (PMSM) and Induction Motor (IM) drives with speed/torque control operation using a two-level DC-AC voltage source converter.
2. To measure and analyze the power, torque, and efficiency of a 4-wheeler EV chassis under varying operational conditions during a complete drive cycle.
3. To understand and measure energy flow in an EV power train during modes such as charging, Vehicle-to-Grid (V2G) feeding, motoring, and braking.
4. To operate an EV in all four quadrants (forward motoring, forward braking, reverse motoring, and reverse braking) using multiple motor drives and demonstrate necessary PWM control techniques.
5. To implement and demonstrate special synchronized PWM techniques for high-power and high-speed Induction Motor (IM) drives with field weakening capabilities.
6. To compute turn-on and turn-off losses in power semiconductor devices (e.g., IGBT, MOSFET) under varying load conditions.
7. To design and implement a bidirectional DC-DC converter and analyze its operation in charging and discharging modes.
8. To design and implement a bidirectional DC-DC converter and analyze its operation in charging and discharging modes.
9. To develop models for Induction Motor (IM) drives in both forward and reverse motoring modes.
10. To design a control system for SRM drives and study its performance under varying load conditions.
11. To model and control a PMSM drive using advanced control techniques such as field-oriented control (FOC).
12. To apply and analyze special PWM techniques for high-speed PMSM drives and study their performance.
13. To implement field-oriented control on high-power IM drives and analyze its effectiveness in EV applications.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercise.)

Department of Computer Science and Engineering

Faculty of Technology

University of Delhi

Detailed Course Structure and Curriculum of B.Tech. (CSE) Third Year

Department of Computer Science and Engineering
Faculty of Technology
University of Delhi

Course Structure of the B.Tech. (CSE) Third Year

Semester – V

S.No.	Course Domain	Course Title	Credits*			Total Credits
			L	T	P	
1.	DSC-13	Theory of Computation	3	0	1	4
2.	DSC-14	Artificial Intelligence and Machine Learning	3	0	1	4
3.	DSC-15	Computer Networks	3	0	1	4
4.	DSE-3	Select a course from the specified list of DSE-3				4
5.	GE-5	Select a course from the specified list of GE-5				4
6.	SEC or IAPC	Choose one SEC or Internship/Apprenticeship/Project/Community Outreach (IAPC)				2
Total Credits						22

Semester – VI

S.No.	Course Domain	Course Title	Credits*			Total Credits
			L	T	P	
1.	DSC-16	Cybersecurity	3	0	1	4
2.	DSC-17	Cloud Computing	3	0	1	4
3.	DSC-18	Software Project Management	3	0	1	4
4.	DSE-4	Select a course from the specified list of DSE-4				4
5.	GE-6	Select a course from the specified list of GE-6				4
6.	SEC or IAPC	Choose one SEC or Internship/Apprenticeship/Project/Community Outreach (IAPC)				2
Total Credits						22

**Credits*
L (01 Credit) is equivalent to 01 contact hour per week
T (01 Credit) is equivalent to 01 contact hour per week
P (01 Credit) is equivalent to 02 contact hours per week

Department of Computer Science and Engineering
Faculty of Technology
University of Delhi

Pool of DSEs offered by the department in Third Year

S.No.	Semester	DSE	Paper Title
1.	V	DSE - 3	Object Oriented Programming
2.			Computational Statistics and Probability
3.			Front-end Web Design and Development
4.			Discrete Structures
5.	VI	DSE - 4	Foundations of Data Analysis
6.			Computer Graphics
7.			Introduction to IoT
8.			Optimization Techniques
9.			Compiler Design

List of SEC/IAPC offered by the Department of Computer Science and Engineering in Third Year

S. No	Semester	SEC / IAPC	Course Title
1.	V	IAPC	Internship
2.	VI	SEC	Full Stack Application Development

Department of Computer Science and Engineering
Faculty of Technology
University of Delhi

Specializations/Minors offered by the Department of Computer Science and Engineering

Semester	GE	Minor (for ECE / EE)	Specializations / Minors (Open to CSE / ECE / EE)				
			Artificial Intelligence & Machine Learning	**Data Science**	**Software Engineering**	**Blockchain and Cybersecurity**	**Augmented Reality / Virtual Reality**
V	GE-5	Foundations of Computer Networks	Data Mining & Warehousing	Data Mining & Warehousing	Predictive Modeling	Blockchain Essentials	Human Computer Interaction
			Neural Networks	Advanced Data Analytics	Software Testing	Ethical Hacking and Advanced Cybersecurity	Game Development
VI	GE-6	Fundamentals of Software Engineering	AI for Image Processing	Health Data Analytics	Agile Software Development	Blockchain Applications	Advanced Game Development
			NLP: Techniques and Applications	Fundamentals of Time Series Analysis	Software Reliability	Cybersecurity with Blockchain	Augmented and Virtual Reality Systems Design

Detailed Syllabus of Discipline Specific Core (DSC) Courses for B.Tech. (CSE) - Semester V

THEORY OF COMPUTATION (DSC-13)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Theory of Computation	4	3L	1T	0P	-	-

Course Hours: L: 03 T: 01 P: 00

Course Outcomes
At the end of this course, students will be able to:
1. Understand fundamental models of computation and their capabilities (e.g., finite automata, pushdown automata, and Turing machines).
2. Analyze formal languages and classify them into regular, context-free, and recursively enumerable languages.
3. Construct formal proofs to show properties of languages and automata, and prove language equivalences or non-equivalences.
4. Reason about computational limits, including undecidability and complexity classes, to understand the boundaries of algorithmic solvability.

Course Objectives
1. Introduce students to the theoretical underpinnings of computation and formal language theory.
2. Enable students to model and analyze computational problems using mathematical frameworks such as automata and grammars.
3. Impart knowledge on the relationships between different classes of languages and the machines that recognize them.
4. Familiarize students with Turing machines and concepts of decidability, reducibility, and complexity.

UNIT I
Automata Theory: Introduction to automata theory: definitions and concepts - symbols, alphabets, strings, and languages, Finite automata - DFA and NFA, Minimization of DFA, Regular Expressions, Properties of regular languages: pumping lemma, closure properties.

UNIT II
Context-Free Languages and Pushdown Automata: Context-Free Grammars (CFGs): Definition, derivations, parse trees, ambiguity. Normal forms: Chomsky Normal Form (CNF) and Greibach Normal Form (GNF). Pushdown Automata (PDA): Definition and acceptance by empty stack and final state. Equivalence of PDAs and CFGs. Properties of Context-Free Languages: Closure properties, Pumping Lemma for CFLs. Applications of CFLs in compiler design and syntactic parsing.

UNIT III

Computability Theory: The Church-Turing thesis and models of computation, Turing machines: definition, design, and examples, Recursive and Recursively Enumerable Languages, Decidability: recognizable and decidable languages, The Halting problem and reductions, Non-computable functions and undecidable problems in computer science

UNIT IV
Complexity Theory: Introduction to complexity classes: P, NP, NP-complete, NP-hard, Polynomial time reductions and examples of NP-complete problems, Space complexity: PSPACE, NPSPACE, L, NL, and the Savitch's theorem, Time-space trade-offs and the class hierarchy, Complexity classes beyond NP: EXPSPACE, NEXP

Tutorial Component

1. Designing and simulating finite automata for simple problems.
2. Designing and simulating PDA.
3. Designing and simulating Turing machines.
4. Implementing algorithms for parsing regular expressions.
5. Implementing algorithms for parsing context-free grammars.
6. Solving decision problems and proving decidability.
7. Reductions among computational problems to prove NP-completeness.

List of tutorial tasks
Note: The course instructor will design tasks to complete the tutorial component of the course.

Suggested Readings

1. "Introduction to the Theory of Computation" by Michael Sipser
2. "An Introduction to Formal Languages and Automata" by Peter Linz
3. "Introduction to Automata Theory, Languages, and Computation" by John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman
4. "Computational Complexity: A Modern Approach" by Sanjeev Arora and Boaz Barak

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING (DSC-14)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Artificial Intelligence and Machine Learning	4	3L	0T	1P	-	Data Structures

Course Hours: L: 03 T: 00 P: 02

Course Outcomes

At the end of this course, students will be able to:

1. Understand the basic concepts of Artificial Intelligence, including problem-solving, search strategies, and heuristic methods.
2. Implement machine learning algorithms such as regression, decision trees, and evaluation metrics, and understand their performance.
3. Apply deep learning techniques using neural networks, CNNs, and RNNs to solve problems in image and sequence data.
4. Implement and evaluate reinforcement learning algorithms, including Q-learning and policy gradients, in real-world applications.
5. Understand and apply natural language processing techniques for text analysis and classification.
6. Utilize generative models like autoencoders and GANs for tasks involving data generation and image synthesis.

Course Objectives

1. Provide students with foundational knowledge of AI, focusing on search algorithms, problem-solving techniques, and applications.
2. Introduce machine learning algorithms, with an emphasis on both supervised and unsupervised learning methods.
3. Enable students to explore deep learning and neural networks for handling complex data, including images and text.
4. Equip students with practical skills in reinforcement learning and its applications in dynamic, real-world environments.
5. Introduce students to NLP concepts, such as text preprocessing, classification, and sentiment analysis.
6. Provide an understanding of generative models and their applications in image generation and anomaly detection.

UNIT I

Introduction to Artificial Intelligence (AI)*:* Definition and Scope of AI; History and Evolution of AI; AI Applications: Natural Language Processing, Robotics, Expert Systems, and Vision Systems; Search Strategies: Uninformed Search (BFS, DFS), Informed Search (Greedy, A*), and Heuristics; Problem-Solving Techniques: Constraint Satisfaction Problems, Game Playing, and Adversarial Search.

UNIT II

Machine Learning Fundamentals*:* Introduction to Machine Learning (ML): Types of Learning – Supervised, Unsupervised, and Reinforcement Learning; Linear Regression, Logistic Regression, and Decision Trees; Model

Evaluation Metrics: Confusion Matrix, Precision, Recall, F1 Score, and ROC-AUC; Overfitting and Underfitting; Regularization Techniques: Lasso and Ridge Regression.

UNIT III
***Deep Learning and Neural Networks**:* Introduction to Neural Networks; Perceptrons and Multilayer Perceptrons (MLP); Backpropagation and Gradient Descent Optimization; Convolutional Neural Networks (CNN): Architecture and Applications in Image Processing; Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM): Applications in Sequential Data and Text Processing.

UNIT IV
Advanced Topics and AI Applications: Reinforcement Learning: Markov Decision Processes, Q-Learning, and Policy Gradients; Natural Language Processing (NLP): Tokenization, Stemming, Lemmatization, and Text Classification; Generative Models: Autoencoders and Generative Adversarial Networks (GANs); Ethical Implications of AI: Bias, Fairness, and Accountability; Case Studies: AI in Healthcare, Autonomous Vehicles, and Financial Systems.

Practical Component

1. Implementing a basic AI agent for problem-solving using search algorithms like BFS and DFS.
2. Developing machine learning models for regression and classification using popular algorithms like Decision Trees, k-NN, and SVM.
3. Implementing and training a simple neural network using Python and TensorFlow or PyTorch.
4. Designing and testing deep learning models, including CNNs, for image classification tasks.
5. Implementing reinforcement learning algorithms like Q-learning and testing them in a simulated environment.
6. Developing a natural language processing pipeline for text classification, sentiment analysis, or spam detection.
7. Building and testing generative models (e.g., GANs) for image generation tasks.

List of Experiments
Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Suggested Readings

1. “Artificial Intelligence: A Modern Approach” by Stuart Russell and Peter Norvig (3rd Edition)
2. “Pattern Recognition and Machine Learning” by Christopher Bishop
3. “Deep Learning” by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
4. “Machine Learning: A Probabilistic Perspective” by Kevin P. Murphy
5. “Python Machine Learning” by Sebastian Raschka
6. “Reinforcement Learning: An Introduction” by Richard S. Sutton and Andrew G. Barto
7. “Natural Language Processing with Python” by Steven Bird, Ewan Klein, and Edward Loper

COMPUTER NETWORKS (DSC-15)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Computer Networks	4	3L	0T	1P	-	Operating System

Course Hours: L: 03 T: 00 P: 02

Course Outcomes

Upon completion of this course, students will be able to:

1. Understand and explain the various layers of computer networks and the protocols used in each layer.
2. Analyze network performance using key metrics and troubleshoot networking issues.
3. Demonstrate the ability to design and implement network solutions.
4. Identify and describe the latest trends in computer networking, including software-defined networking (SDN), Internet of Things (IoT), and 5G technologies.
5. Apply networking concepts through practical experiments in simulated environments.

Course Objectives

1. To introduce the fundamental concepts of computer networks and communication systems.
2. To provide an understanding of the network architecture, protocols, and layers.
3. To examine various types of networks and their components, including physical media, protocols, and network devices.
4. To familiarize students with current trends and technologies in networking such as SDN, IoT, and 5G.
5. To enhance problem-solving and analytical skills through hands-on experiments in network simulation and implementation.

UNIT I
Introduction to Computer Networks: Overview of computer networks and their importance, Network topologies, types, and models, OSI Reference Model, TCP/IP reference model, Addressing, Network Devices, Introduction to software-defined networking (SDN)
Physical layer: transmission media, signaling, modulation and demodulation, switching

UNIT II
Data link layer: design issues, framing, error detection and correction, Elementary data link protocols, Sliding window protocols, MAC Protocols
Network Layer: Network layer design issues, Shortest path routing, Link-State Routing algorithm, Distance-Vector Routing algorithm; Logical Addressing and Internet Protocols: IPv4 and IPv6, Address Mapping, Error Reporting ICMP, Network devices: routers, switches, firewalls

UNIT III
Transport Layer: Process to process Delivery, Elements of Transport Protocol, Congestion control, Internet Transport Protocol: UDP and TCP

Application layer protocols: HTTP, DNS, FTP, SMTP, Electronic Mail

UNIT IV

Network Security: Security Services, Communication security, email security, web security

Emerging Technologies: SDN architecture, and applications, Wireless and mobile networks: WiFi, LTE, 5G, Networking IoT devices and protocols like MQTT and CoAP, Network virtualization and cloud computing, Blockchain in networking

Practical Component

1. Setting up a small network and configuring network devices.
2. Analyzing network traffic using packet sniffing tools.
3. Simulating routing protocols using network simulation software (Simulate and visualize the working of RIP, OSPF, and BGP in a network environment).
4. Setting up and configuring a simple LAN using switches and routers
5. Simulate the three-way handshake process of TCP and implement error control and flow control.
6. Implementing a basic secure network communication protocol.
7. Overview and configuration of 5G network components in a simulator.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Suggested Readings

1. "Data Communications and Networking" by Behrouz A. Forouzan
2. "Computer Networks" by Andrew S. Tanenbaum and David J. Wetherall (5th Edition, Pearson)
3. "Computer Networking: A Top-Down Approach" by James Kurose and Keith Ross (6th Edition, Pearson)
4. "Data and Computer Communications" by William Stallings (10th Edition, Pearson)
5. "Network Security Essentials: Applications and Standards" by William Stallings

Detailed Syllabus of Discipline Specific Elective (DSE) courses for B.Tech. (CSE) – Semester V (DSE-3)

OBJECT-ORIENTED PROGRAMMING (DSE-3)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Object-Oriented Programming	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes
At the end of this course, students will be able to:
1. Understand and apply basic OOP concepts.
2. Implement advanced OOP concepts such as abstract classes, interfaces, exception handling, generic programming, inner classes, lambda expressions, reflection, and annotations in software development projects.
3. Apply creational, structural, and behavioral design patterns in real-world software development scenarios.
4. Use OOP principles to design and develop small to medium-scale software projects.

Course Objectives
1. Provide students with a thorough understanding of the basic principles of object-oriented programming, including classes, objects, inheritance, polymorphism, encapsulation, and the use of constructors and destructors.
2. Equip students with advanced OOP concepts such as abstract classes, interfaces, exception handling, generic programming, inner classes, lambda expressions, reflection, and annotations.
3. Introduce students to various design patterns, their real-world applications, and the impact of anti-patterns, emphasizing creational, structural, and behavioral patterns.
4. Enable students to apply OOP principles in software engineering using languages such as Java and C++, exploring advanced features and best practices, and analyzing case studies of large-scale OOP-based software systems.

UNIT I
OOP Concepts: Fundamentals of OOP: Classes, Objects, Methods, Inheritance: Single, Multiple, Multilevel, Hierarchical, Polymorphism: Compile-time and Runtime, Encapsulation and Data Hiding, Constructors and Destructors.

UNIT II
Advanced OOP Concepts: Abstract Classes and Interfaces, Exception Handling and Assertions, Generic Programming, Inner Classes and Lambda Expressions, Reflection and Annotations.

UNIT III

Design Patterns: Creational Patterns: Singleton, Factory, Builder, Prototype, Structural Patterns: Adapter, Composite, Proxy, Flyweight, Behavioral Patterns: Strategy, Command, Observer, Iterator, Real-world applications of design patterns, Anti-patterns and their impact.

UNIT IV

OOP in Software Development: OOP principles in software engineering, OOP in Java: Advanced features and libraries, OOP in C++: STL, Templates, Operator Overloading, Case studies: OOP in large-scale software systems, Best practices in OOP

Practical Component

1. Development of small to medium scale OOP projects
2. Implementation of various design patterns in projects
3. Object-oriented analysis and design exercises
4. Case studies: Analyzing OOP-based software

Suggested Readings

1. "Object-Oriented Analysis and Design with Applications" by Grady Booch, Robert A. Maksimchuk, Michael W. Engle
2. "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

COMPUTATIONAL STATISTICS AND PROBABILITY (DSE-3)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Computational Statistics and Probability	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes

At the end of this course, students will be able to:

1. Understand and apply probability theory.
2. Analyze random variables and distributions.
3. Execute point and interval estimation, perform linear and multiple regression analysis, conduct ANOVA and non-parametric tests, and analyze time series data for forecasting purposes.
4. Apply Monte Carlo methods and simulations, understand and use queueing theory (M/M/1, M/M/c), model decision processes using Markov chains, analyze system reliability and survival, and utilize statistical learning and data mining techniques.

Course Objectives

1. Provide students with a thorough understanding of the basic concepts of probability, including conditional probability, random variables, and key probability distributions, along with expectations and moment generating functions.
2. Equip students with knowledge of joint, marginal, and conditional distributions, functions of random variables, the Central Limit Theorem, sampling distributions, and hypothesis testing methods.
3. Enable students to perform point and interval estimation, regression analysis, ANOVA, non-parametric tests, and time series analysis, and to understand their applications in data analysis.
4. Familiarize students with practical applications of computational statistics in computer science, including Monte Carlo simulations, queueing theory, Markov chains, reliability theory, and statistical learning.

UNIT I

Probability Theory: Basic concepts of probability, Conditional probability and Bayes' theorem, Discrete and continuous random variables, Probability distributions: Binomial, Poisson, Normal, Expectation, Variance, and Moment generating functions.

UNIT II

Random Variables and Distributions: Joint, Marginal, and Conditional distributions, Functions of random variables, Central Limit Theorem and its implications, Sampling distributions and estimators, Hypothesis testing: Z-test, T-test, Chi-square test.

UNIT III

Statistical Methods: Point and Interval estimation, Regression analysis: Linear and Multiple regression, Analysis of Variance (ANOVA), Non-parametric tests, Time series analysis and forecasting.

UNIT IV

Applications in Computer Science: Monte Carlo methods and simulations, Queueing theory: M/M/1, M/M/c queues, Markov chains and decision processes, Reliability theory and survival analysis, Statistical learning and data mining.

Practical Component

1. Statistical analysis using R or Python
2. Data visualization techniques
3. Simulations for probabilistic models
4. Case studies: Application of statistics in computer science

Suggested Readings

1. "Probability and Statistics for Engineering and the Sciences" by Jay L. Devore
2. "Introduction to Probability and Statistics for Engineers and Scientists" by Sheldon M. Ross

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

FRONT-END WEB DESIGN AND DEVELOPMENT (DSE-3)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Front-end Web Design and Development	4	0L	0T	4P	-	-

Course Hours: L: 00 T: 00 P: 04

Course Outcomes

At the end of this course, students will be able to:

1. Create and style web pages using HTML and CSS.
2. Develop interactive web pages using JavaScript.
3. Apply responsive design principles using media queries and responsive frameworks. Develop mobile-first designs, ensure accessibility and adherence to web standards, and achieve cross-browser compatibility.
4. Build applications using front-end frameworks and libraries.

Course Objectives

1. Provide students with a comprehensive understanding of HTML5 and CSS3, covering essential elements, attributes, semantic HTML, selectors, the box model, and layout techniques including Flexbox and Grid Layout. Additionally, introduce responsive web design principles and CSS preprocessors like SASS and LESS.
2. Equip students with the basics of JavaScript, including syntax, variables, and data types, as well as advanced concepts like ES6 features. Enable students to manipulate the DOM, handle events, and make asynchronous web requests using AJAX and the Fetch API.
3. Teach students the principles of responsive web design, including media queries and responsive frameworks, the mobile-first design approach, accessibility standards, and cross-browser compatibility.
4. Familiarize students with popular front-end frameworks and libraries, including React.js, Vue.js, and Angular. Teach the basics of building single-page applications (SPAs) and utilizing UI design frameworks like Bootstrap.

UNIT I

HTML and CSS: HTML5: Elements, Attributes, Semantic HTML, CSS3: Selectors, Box Model, Flexbox, Grid Layout, Responsive Web Design Principles, CSS Preprocessors: SASS, LESS.

UNIT II

JavaScript and DOM Manipulation: JavaScript Basics: Syntax, Variables, Data Types, DOM Manipulation: Selectors, Events, Event Listeners, ES6 Features: Arrow Functions, Promises, Modules, AJAX and Fetch API for Asynchronous Web Requests.

UNIT III

Responsive Design: Media Queries and Responsive Frameworks, Mobile-First Design Approach, Accessibility and Web Standards, Cross-Browser Compatibility.

UNIT IV

Frameworks and Libraries: Introduction to React.js: Components, State, Props, Working with Bootstrap for UI Design, Introduction to Vue.js and Angular, Building Single Page Applications (SPAs).

Practical Component

1. Designing and developing a responsive website
2. Implementing interactive features using JavaScript and frameworks
3. Project: Develop a complete front-end for a web application

Suggested Readings

1. "HTML and CSS: Design and Build Websites" by Jon Duckett
2. "Eloquent JavaScript: A Modern Introduction to Programming" by Marijn Haverbeke
3. "Learning React: A Hands-On Guide to Building Web Applications Using React and Redux" by Kirupa Chinnathambi

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

DISCRETE STRUCTURES (DSE-3)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Discrete Structures	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes

At the end of this course, students will be able to:

1. Understand and apply concepts of sets, relations, and functions.
2. Apply basic counting techniques in combinatorial problems, understand and work with different types of graphs and their representations, perform graph traversals, and analyze properties of trees and planar graphs. Utilize graph coloring techniques and network models in practical scenarios.
3. Understand and apply the properties of groups, subgroups, rings, and fields. Work with polynomials and understand their applications. Analyze and utilize lattices and Boolean algebra in various contexts.
4. Analyze and construct propositional and predicate logic statements, use logical connectives and truth tables to determine tautologies and contradictions, understand logical equivalence and normal forms. Simplify Boolean functions and apply them in digital logic design.
5. Participate in hands-on exercises and projects that involve set theory, combinatorics, graph theory, algebraic structures, and logic.

Course Objectives

1. Provide students with a comprehensive understanding of set theory, relations, and functions, including their definitions, properties, and operations.
2. Equip students with basic counting techniques and the fundamentals of graph theory, including graph types, representations, traversals, trees, planar graphs, graph coloring, and network models.
3. Introduce students to algebraic structures such as groups, rings, and fields, along with their properties and applications.
4. Familiarize students with propositional and predicate logic, logical connectives, truth tables, tautologies, contradictions, logical equivalence, and normal forms.

UNIT I

Sets, Relations, and Functions: Set Theory: Definitions, Operations, Venn Diagrams, Relations: Types, Properties, Closure, Functions: Types, Composition, Inverse, Counting: Pigeonhole Principle, Permutations and Combinations, Mathematical Induction and Recursion.

UNIT II

Combinatorics and Graph Theory: Basic Counting Techniques, Graphs: Types, Representations, Traversals, Trees: Properties, Binary Trees, Tree Traversals, Planar Graphs, Graph Coloring, Network Models and Algorithms.

UNIT III

Algebraic Structures: Groups: Definitions, Properties, Subgroups, Rings and Fields: Definitions, Properties, Polynomials and their Applications, Lattices and Boolean Algebra.

UNIT IV

Logic and Boolean Algebra: Propositional and Predicate Logic, Logical Connectives, Truth Tables, Tautologies, and Contradictions, Logical Equivalence, Normal Forms, Boolean Functions, Simplification of Boolean Functions, Applications in Digital Logic Design.

Practical Component

1. Problem-solving sessions using combinatorial techniques
2. Implementing graph algorithms
3. Boolean function simplification using software tools

Suggested Readings

1. "Discrete Mathematics and Its Applications" by Kenneth H. Rosen
2. "Concrete Mathematics: A Foundation for Computer Science" by Ronald L. Graham, Donald E. Knuth, and Oren Patashnik
3. "Discrete Mathematical Structures with Applications to Computer Science" by J. P. Trembly& P. Manohar
4. "Elements of Discrete Mathematics" by C. L. Liu

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Detailed Syllabus of Discipline Specific Core (DSC) Courses for B.Tech. (CSE) - Semester VI

CYBERSECURITY (DSC-16)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Cybersecurity	4	3L	0T	1P	-	Operating Systems, Computer Networks

Course Hours: L: 03 T: 00 P: 02

Course Outcomes
At the end of this course, students will be able to:
1. Understand the principles of cybersecurity and identify various types of cyber threats.
2. Apply encryption techniques and cryptographic protocols to secure communication channels.
3. Design and implement secure system architectures to mitigate risks.
4. Utilize tools and techniques to perform vulnerability assessments and penetration testing.
5. Analyze and respond to security incidents effectively to protect digital assets.

Course Objectives
1. Provide a comprehensive understanding of fundamental cybersecurity principles, threats, and countermeasures.
2. Equip students with the skills to analyze and design secure systems and networks.
3. Familiarize students with encryption, authentication, and cryptographic protocols.
4. Develop critical thinking and problem-solving skills in securing digital environments against cyber threats.
5. Enable hands-on experience with security tools and frameworks for assessing and improving cybersecurity postures.

UNIT I
Introduction to Cybersecurity: Definition and importance of cybersecurity, core principles (confidentiality, integrity, availability), and frameworks like NIST and ISO 27001. Types of cyber threats: phishing, malware, ransomware, and vulnerabilities. Basic cryptographic techniques: encryption, hashing, and digital signatures. Overview of compliance and regulatory frameworks such as GDPR and HIPAA. Importance of security policies in organizational contexts.

UNIT II
Network Security: Network security mechanisms: firewalls, intrusion detection/prevention systems (IDS/IPS), and virtual private networks (VPNs). Secure network design principles: subnetting, VLANs, and demilitarized zones (DMZ). Vulnerabilities in network protocols (e.g., TCP/IP) and secure protocols like HTTPS and SSH. Wireless security principles, including WPA3 and defenses against common wireless attack vectors.

UNIT III

System and Application Security: Operating system security: user access control, privilege escalation, and securing Windows/Linux systems. Secure software development lifecycle (SDLC) practices to prevent vulnerabilities like SQL injection and buffer overflows. Web application security using OWASP Top 10 guidelines. Endpoint security management: anti-malware solutions, patch management, and best practices for safeguarding devices.

UNIT IV

Advanced Topics in Cybersecurity: Incident response lifecycle: detection, containment, eradication, recovery, and forensics. Security challenges in emerging technologies, including cloud computing, Internet of Things (IoT), and blockchain. Ethical hacking and penetration testing techniques using tools like Metasploit and Nmap. Future trends in cybersecurity: artificial intelligence in threat detection and the implications of quantum computing on cryptography.

Practical Component

1. Configuring and managing firewalls to enforce network security policies.
2. Performing network scanning and penetration testing using tools like Nmap and Metasploit.
3. Implementing Intrusion Detection Systems (IDS) to monitor network traffic for suspicious activities.
4. Encrypting and decrypting sensitive data using algorithms such as AES or RSA.
5. Conducting vulnerability assessments of web applications and fixing identified issues.
6. Simulating and responding to Distributed Denial of Service (DDoS) attacks.
7. Developing and analyzing secure authentication mechanisms using multi-factor authentication (MFA).

Suggested Readings

1. "Computer Security: Principles and Practice" by William Stallings and Lawrie Brown (Pearson)
2. "Cybersecurity Essentials" by Charles J. Brooks, Christopher Grow, Philip Craig, and Donald Short (Wiley)
3. "The Art of Deception" by Kevin Mitnick and William L. Simon (Wiley)
4. "Practical Malware Analysis" by Michael Sikorski and Andrew Honig (No Starch Press)

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

CLOUD COMPUTING (DSC-17)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Cloud Computing	4	3L	0T	1P	-	Computer Networks

Course Hours: L: 03 T: 00 P: 02

Course Outcomes

At the end of this course, students will be able to:

1. Understand fundamental concepts of cloud computing, including service and deployment models, and evaluate the benefits and limitations of cloud-based solutions.
2. Explore and manage virtualized resources, storage services, and networking components in the cloud, as well as identify and implement appropriate security measures.
3. Design, develop, and deploy cloud-native applications using modern tools, containerization, and serverless architectures.
4. Leverage advanced cloud services for big data analytics, machine learning, blockchain, and emerging cloud-based innovations.

Course Objectives

1. Introduce the fundamental principles, models, and services of cloud computing and its evolving ecosystem.
2. Provide hands-on knowledge of infrastructure management, virtualization, cloud networking, and security best practices.
3. Enable students to design, implement, and manage cloud-native applications through containers, orchestration, and serverless paradigms.
4. Familiarize students with advanced cloud services, including big data analytics, machine learning, and blockchain on the cloud.
5. Foster an understanding of cutting-edge and future cloud technologies, including quantum and edge computing, and their impact on modern IT landscapes.

UNIT I

Cloud Computing Fundamentals: Introduction to Cloud Computing: Definition, characteristics, evolution. Service Models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS). Deployment Models: Public, Private, Hybrid, and Community Clouds. Economic and Organizational Considerations: Cost-benefit analysis, scalability, agility. Challenges and Risks: Vendor lock-in, data privacy and security, regulatory compliance, SLA issues.

UNIT II

Cloud Infrastructure and Management: Virtualization Concepts: Hypervisors, VMs, containers, resource abstraction and pooling. Cloud Storage Services and Databases: Object storage, block storage, file storage, NoSQL databases, and distributed storage considerations. Cloud Networking: Virtual Private Clouds (VPCs), Virtual Private Networks (VPNs), Content Delivery Networks (CDN), load balancing. Cloud Security and Compliance: Identity and

Access Management (IAM), data encryption, ISO / IEC security standards for cloud, compliance frameworks (GDPR, HIPAA). Resource Management: Autoscaling, monitoring, logging, cost optimization tools.

UNIT III

Developing and Deploying Cloud Applications: Cloud-Native Application Design: 12-factor apps, microservices architecture, cloud design patterns. Development and Deployment Models: CI/CD pipelines, DevOps and DevSecOps practices in the cloud. Containers and Orchestration: Docker fundamentals, Kubernetes architecture and resource management, Helm charts. Serverless Architectures: Functions as a Service (FaaS).

UNIT IV

Advanced Cloud Technologies and Trends: Big Data Analytics in the Cloud: Data lakes, data warehouses, scalable analytics platforms (e.g., AWS EMR, Google BigQuery). Machine Learning and AI Services: Pre-built ML APIs, custom model training and deployment, AutoML, edge inference.

Practical Component

1. Configuring and deploying a simple cloud-based application.
2. Implementing a containerized application with Docker and deploying it using Kubernetes.
3. Utilizing cloud services for storage, computing, and networking in a project.
4. Developing a serverless application that integrates with cloud-based AI services.

Suggested Readings

1. "Cloud Computing: Concepts, Technology & Architecture" by Thomas Erl, Ricardo Puttini, and Zaigham Mahmood
2. "AWS Documentation" - https://docs.aws.amazon.com/
3. "Kubernetes: Up and Running" by Kelsey Hightower, Brendan Burns, and Joe Beda
4. "Mastering Cloud Computing: Foundations and Applications Programming" by Rajkumar Buyya, Christian Vecchiola, S.Thamarai Selvi.
5. "Cloud Computing Design Patterns" by Thomas Erl, Robert Cope, Amin Naserpour.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

SOFTWARE PROJECT MANAGEMENT (DSC-18)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Software Project Management	4	3L	0T	1P	-	Software Engineering

Course Hours: L: 03 T: 01 P: 00

Course Outcomes
Upon completion of this course, students will be able to:
1. Understand the software project management process and lifecycle.
2. Apply project estimation techniques and create project plans.
3. Manage project schedules, resources, and risks.
4. Use modern tools and methodologies for software project management, including Agile and Scrum.
5. Communicate effectively and lead software development teams.

Course Objectives
1. To introduce students to the concepts, processes, and tools of software project management.
2. To develop an understanding of how to plan, monitor, and control software projects.
3. To learn about risk management, quality management, and resource allocation in a software project.
4. To explore modern software project management practices, including agile methodologies.

UNIT I
Introduction to Software Project Management: Fundamentals of project management, Software Project life cycle and phases, Role of a project manager, Factors leading to project success, Project failure reasons, Key performance indicators (KPIs), Introduction to project management methodologies (Agile, Waterfall, PRINCE2)

UNIT II
Software Project Planning and Estimation: Scope management and work breakdown structure (WBS); Estimation Techniques: Function Point Analysis (FPA), COCOMO model, Use Case Point; Project Scheduling , Gantt charts, PERT charts, Critical Path Method (CPM), Resource allocation and budgeting, Risk management strategies

UNIT III
Project Monitoring and Control: Leadership and team management skills, Communication strategies within a project team, Monitoring project progress and performance metrics, Quality assurance and control in project management, Project closure and post-mortem analysis

UNIT IV
Advanced Topics in Software Project Management: Agile Methodologies, Scrum Framework, Software Project Management Tools like JIRA, Trello, Microsoft Project, OpenProject, Taiga, ProjectLibre, OrangeScrum,Redmine, Remote project management, Virtual teams, Collaborative tools, Impact of AI and Machine Learning in project management

Tutorial Component

1. Create a project charter for a hypothetical software project using *OpenProject* or similar software.
2. Perform stakeholder analysis and create a work breakdown structure (WBS) using *Taiga* or similar software.
3. Use *OrangeScrum* (open-source agile project management software) or similar software to estimate project costs and resources.
4. Create a project schedule using Gantt charts for a software project using *ProjectLibre* or similar software.
5. Perform risk analysis for a software project and develop a risk management plan using *OpenProject.*
6. Use project management software *Taiga* or similar software to monitor project progress and track milestones for an Agile project, integrating KPIs and Earned Value Analysis (EVA).
7. Simulate a project change request process and perform configuration management using version control tools (e.g., Git).
8. Create a project control plan for managing changes using the *Redmine* tool, including a change control board setup.
9. Set up an Agile Scrum project using *Taiga*, define roles (Product Owner, Scrum Master, Development Team), create product backlog, and manage sprints.
10. Use *Redmine* to manage an Agile project, set up sprint backlogs, track tasks, and manage communication with stakeholders.

List of tutorial tasks

Note: The course instructor will design tasks to complete the tutorial component of the course.

Suggested Readings

1. Software Project Management by Bob Hughes, Mike Cotterell (7th Edition, McGraw-Hill)
2. Software Engineering Project Management by Richard H. Thayer (Wiley)
3. Agile Project Management with Scrum by Ken Schwaber (Microsoft Press)
4. The Art of Project Management by Scott Berkun (O'Reilly Media)
5. Modern Software Engineering by David L. Parnas (Springer)

Detailed Syllabus of Discipline Specific Elective (DSE) courses for B.Tech. (CSE) – Semester VI (DSE-4)

FOUNDATIONS OF DATA ANALYSIS (DSE-4)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Foundations of Data Analysis	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes
At the end of this course, students will be able to:
1. Understand the significance of Data Analytics.
2. Apply descriptive and inferential statistics.
3. Perform regression analysis (linear and logistic) and apply classification techniques.
4. Conduct time series analysis and forecasting, and understand the fundamentals of model validation and selection.
5. Participate in hands-on projects that involve data collection, cleaning, analysis, and interpretation.

Course Objectives
1. Provide students with a comprehensive understanding of data analytics, its importance across various domains, and the different types of data.
2. Equip students with knowledge of descriptive and inferential statistics, exploratory data analysis (EDA), and data visualization techniques.
3. Introduce students to the basics of predictive analytics, including regression analysis (linear and logistic), classification techniques, time series analysis, forecasting, and model validation and selection.
4. Provide hands-on experience with data analytics tools such as R and Python, and their libraries.

Unit 1: Introduction to Data Analytics
Overview of data analytics and its significance in various domains, Types of data: structured, unstructured, semi-structured, Data analytics lifecycle: data collection, cleaning, analysis, interpretation, Case studies demonstrating the impact of data analytics.

Unit 2: Data Analysis Techniques
Descriptive statistics: measures of central tendency and variability, Inferential statistics: hypothesis testing, confidence intervals, Introduction to exploratory data analysis (EDA), Data visualization techniques and tools.

Unit 3: Introduction to Predictive Analytics
Basics of regression analysis: linear and logistic regression, Overview of classification techniques, Time series analysis and forecasting basics, Introduction to model validation and selection.

Unit 4: Data Analytics Tools and Applications

Introduction to data analytics software: R, Python and its libraries, Data analytics in business decision-making, Ethical considerations in data analytics, Emerging trends in data analytics.

Practical Component

1. Performing exploratory data analysis using a dataset and presenting findings.
2. Implementing linear regression and logistic regression models on real-world data.
3. Creating data visualizations to interpret and communicate data insights.
4. Conducting a basic time series analysis project.

Suggested Readings

1. "Data Science for Business" by Foster Provost and Tom Fawcett
2. "Python for Data Analysis" by Wes McKinney
3. "Naked Statistics: Stripping the Dread from the Data" by Charles Wheelan

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

COMPUTER GRAPHICS (DSE-4)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Computer Graphics	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes
At the end of this course, students will be able to:
1. Understand and apply foundational computer graphics concepts.
2. Create and manipulate 2D and basic 3D graphics.
3. Develop and render complex 3D models.
4. Conceptualize and develop a complex 3D scene, integrating advanced graphics techniques and user interactions.

Course Objectives
1. Provide students with a comprehensive understanding of the basic principles and applications of computer graphics, including graphics systems, hardware, coordinate systems, and transformations.
2. Equip students with knowledge and skills in basic drawing algorithms, 2D transformations, clipping and windowing techniques, color models, filling algorithms, and an introduction to 3D graphics including transformations and projections.
3. Enable students to create complex 3D models, apply texture mapping and material design, utilize lighting and shading models, implement hidden surface removal algorithms, and understand real-time rendering techniques including shader programming and GPU utilization.
4. Familiarize students with keyframe animation, motion capture, rigging, skeletal animation, facial animation, lip syncing, physics-based animation, and procedural animation techniques.

Unit 1: Foundations of Computer Graphics
Introduction to computer graphics and applications, Graphics systems and hardware, Coordinate systems and transformations, Introduction to OpenGL and graphics libraries.

Unit 2: 2D and Basic 3D Graphics
Basic drawing algorithms: lines, circles, and polygons, 2D transformations and animations, Clipping and windowing techniques, Color models and filling algorithms, Introduction to 3D graphics, 3D transformations and projections.

Unit 3: Advanced 3D Modeling and Rendering
Complex 3D Modeling Techniques, Texture Mapping and Material Design, Lighting and Shading Models, Hidden surface removal algorithms, Real-Time Rendering: Algorithms, Shader Programming, and GPU Utilization.

Unit 4: Animation, Rigging, and Advanced Techniques
Keyframe Animation and Motion Capture, Rigging and Skeletal Animation, Facial Animation and Lip Syncing, Physics-Based Animation and Procedural Animation Techniques, Advanced Graphics Project: Conceptualizing a Complex 3D Scene, Integrating Techniques, Implementing Interactions, and Project Presentation.

Practical Component

1. Implementing 2D and 3D transformations and animations.
2. Developing a simple 3D model and applying texture mapping.
3. Creating basic and advanced shaders for real-time rendering.
4. Conducting a project on complex 3D scene development.
5. Applying animation and rigging techniques to a 3D character.

Suggested Readings

1. "Computer Graphics: Principles and Practice" by John F. Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, and Kurt Akeley
2. "Fundamentals of Computer Graphics" by Peter Shirley, Steve Marschner
3. "Real-Time Rendering" by Tomas Akenine-Möller, Eric Haines, Naty Hoffman

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

INTRODUCTION TO IOT (DSE-4)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Introduction to IoT	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes

At the end of this course, students will be able to:

1. Demonstrate a thorough understanding of the basic components of IoT, its communication models, and networking protocols.
2. Set up and configure IoT development boards like Raspberry Pi, Arduino, and ESP8266.
3. Apply various IoT connectivity technologies such as Wi-Fi, Bluetooth, Zigbee, and LoRa.
4. Understand the application of IoT in different domains including smart homes, industrial IoT, healthcare, and smart cities.

Course Objectives

1. Provide students with a comprehensive understanding of the basics of IoT.
2. Equip students with practical knowledge of IoT development boards, sensors, actuators, and programming languages.
3. Enable students to understand and apply various IoT connectivity technologies, data transmission methods, data protocols, and secure communication techniques.
4. Familiarize students with different applications of IoT in various domains such as smart homes, industrial IoT, healthcare, and smart cities.

Unit 1: IoT Fundamentals

Overview of IoT: Definition, History, and Applications, Basic Components of IoT: Sensors, Actuators, and Controllers, IoT Communication Models: Device-to-Device, Device-to-Cloud, and Device-to-Gateway, IoT Networking Protocols: MQTT, CoAP, HTTP, and WebSocket.

Unit 2: IoT Hardware and Software

Introduction to IoT Development Boards: Raspberry Pi, Arduino, ESP8266, etc., Setting up a basic IoT environment using Raspberry Pi/Arduino, Sensors and Actuators: Types and Use Cases, IoT Programming: Basics of Python and C/C++ for IoT, Interfacing Sensors with IoT Boards, Building IoT applications to collect and display sensor data.

Unit 3: IoT Communication and Networking

IoT Connectivity Technologies: Wi-Fi, Bluetooth, Zigbee, and LoRa, Data Transmission in IoT: Cloud Services and Data Storage, IoT Data Protocols: JSON, XML Secure Communication in IoT: Encryption and Authentication, Interfacing IoT systems with cloud.

Unit 4: IoT Applications and Case Studies

Smart Home and Building Automation, Industrial IoT (IIoT) and Smart Manufacturing, Healthcare and Wearable IoT Devices, Smart Cities and Infrastructure, Case studies of end-to-end IoT applications (e.g., smart home system, health monitoring device).

Practical Component

1. Weekly Lab Sessions: Hands-on experience with IoT development boards and sensors.
2. Mini Projects: Regular small projects to implement different IoT functionalities.
3. Final Project: A comprehensive project to design and develop a complete IoT system.
4. Case Study Analysis: Reviewing real-world IoT applications and their implementations.

Suggested Readings

1. "Internet of Things: A Hands-On Approach" by Arshdeep Bahga and Vijay Madisetti
2. "The Internet of Things: Enabling Technologies, Platforms, and Use Cases" by Pethuru Raj and Anupama C. Raman
3. "Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry" by Maciej Kranz

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

OPTIMIZATION TECHNIQUES (DSE-4)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Optimization Techniques	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes

At the end of this course, students will be able to:

1. Demonstrate a thorough understanding of optimization principles, types, and mathematical foundations.
2. Apply linear and non-linear programming techniques.
3. Solve discrete optimization problems using heuristic methods.
4. Utilize advanced optimization techniques in real-world applications.

Course Objectives

1. Provide students with a comprehensive understanding of the basics of optimization, including its definition, types, importance, and mathematical foundations.
2. Equip students with knowledge of linear and non-linear programming methods, including the Simplex method, duality, sensitivity analysis, gradient descent, Newton's method, and constrained optimization techniques.
3. Introduce students to discrete optimization techniques such as integer programming, branch and bound, cutting planes, and combinatorial optimization problems.

Unit 1: Introduction to Optimization

Basics of Optimization: Definition, Types, and Importance, Mathematical Foundations: Linear Algebra and Calculus Review, Formulating Optimization Problems: Objective Function, Constraints, Types of Optimization Problems: Linear, Non-linear, Integer, and Combinatorial, Solving basic optimization problems using Python (SciPy).

Unit 2: Linear and Non-Linear Programming

Linear Programming: Simplex Method, Duality, Sensitivity Analysis, Non-Linear Programming: Gradient Descent, Newton's Method, Constrained Optimization: Lagrange Multipliers, KKT Conditions, Implementation of linear and non-linear optimization algorithms using Python (SciPy, CVXPY).

Unit 3: Discrete Optimization and Heuristics

Integer Programming: Branch and Bound, Cutting Planes, Combinatorial Optimization: Traveling Salesman Problem, Knapsack Problem, Heuristic Methods: Genetic Algorithms, Simulated Annealing, Tabu Search, Solving discrete optimization problems using Python (PuLP, DEAP).

Unit 4: Advanced Optimization Techniques and Applications

Multi-Objective Optimization: Pareto Optimality, Weighted Sum Method, Metaheuristics: Particle Swarm Optimization, Ant Colony Optimization, Real-World Applications: Scheduling, Network Optimization, Machine Learning, Developing optimization solutions for real-world problems using Python and specialized libraries (e.g., PyGMO).

Practical Component

1. Hands-on practice with optimization algorithms and tools.
2. A small project to apply optimization techniques to various problems.
3. A comprehensive project to formulate and solve a complex optimization problem.
4. Reviewing and analyzing optimization problems in real-world scenarios.

Suggested Readings

1. "Introduction to Operations Research" by Frederick S. Hillier and Gerald J. Lieberman
2. "Linear and Nonlinear Programming" by David G. Luenberger and Yinyu Ye
3. "Optimization in Operations Research" by Ronald L. Rardin
4. "Practical Optimization: Algorithms and Engineering Applications" by Andreas Antoniou and Wu-Sheng Lu

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

COMPILER DESIGN (DSE-4)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Compiler Design	4	3L	0T	1P	-	-

Course Outcomes

At the end of this course, students will be able to:

1. Understand the phases and components of a compiler, including lexical analysis, syntax analysis, and code generation.
2. Design and implement lexical analyzers using finite state machines and regular expressions.
3. Apply different parsing techniques such as top-down parsing, bottom-up parsing, and LR parsing to construct efficient parsers.
4. Develop and implement syntax-directed translation schemes to generate intermediate code from high-level languages.
5. Use appropriate data structures, such as symbol tables, for managing information during the compilation process.
6. Implement error detection and recovery techniques to handle common syntax and semantic errors during compilation.
7. Optimize code through various techniques, such as loop optimization and DAG representation.

Course Objectives

1. Provide students with a comprehensive understanding of compiler components and their functionalities.
2. Introduce students to the design and implementation of lexical analyzers using finite state machines and regular expressions.
3. Teach various parsing techniques and help students develop efficient parsers for different programming languages.
4. Introduce syntax-directed translation schemes and help students generate intermediate code for high-level programming languages.
5. Familiarize students with the implementation of symbol tables and memory management during the compilation process.
6. Equip students with the skills to detect and recover from errors during the compilation process.
7. Teach optimization techniques to improve the efficiency of the generated code.

Unit 1: Introduction to Compilers

Definition and Role of a Compiler; Phases of a Compiler: Lexical Analysis, Syntax Analysis, Semantic Analysis, Intermediate Code Generation, Optimization, and Code Generation; Compiler Passes: Single-pass and Multi-pass Compilers; Overview of Finite State Machines (FSM) and Regular Expressions in Lexical Analysis; Implementation of Lexical Analyzers and Lexical Analyzer Generators; Introduction to Lex – A Lexical Analyzer Generator.

Unit 2: Syntax Analysis

Role of Syntax Analysis in Compilers; Context-Free Grammars (CFG) and Backus-Naur Form (BNF); Parse Trees and Abstract Syntax Trees; Parsing Techniques: Top-Down Parsing (Recursive Descent, Predictive Parsing) and

Bottom-Up Parsing (Shift-Reduce Parsing, Operator Precedence Parsing); LR Parsers: SLR, Canonical LR, and LALR Parsing Techniques; Constructing Parsing Tables; Introduction to YACC – A Parser Generator.

Unit 3: Syntax-Directed Translation and Intermediate Code Generation

Syntax-Directed Definitions and Translation Schemes; Implementation of Syntax-Directed Translators; Intermediate Representations: Postfix Notation, Syntax Trees, Three-Address Code, Quadruples, and Triples; Translation of Programming Constructs: Assignment Statements, Boolean Expressions, Control Flow, Array References, Procedure Calls, and Case Statements; Runtime Environments and Storage Allocation Strategies.

Unit 4: Code Optimization and Code Generation

Need for Optimization; Basic Blocks and Flow Graphs; Loop Optimization Techniques; Data Flow Analysis for Optimization; DAG Representation of Basic Blocks; Peephole Optimization; Code Generation Techniques: Instruction Selection, Register Allocation, and Instruction Scheduling; Challenges in Code Generation: Target Machine Architecture and Runtime Considerations.

Practical Component

1. Implementing a lexical analyzer using regular expressions and finite state machines (FSM).
2. Developing a simple parser using recursive descent parsing for basic arithmetic expressions.
3. Constructing and testing an LR parser using tools like YACC or Bison.
4. Implementing syntax-directed translation schemes to convert expressions into intermediate code.
5. Designing a symbol table management system for tracking variables, functions, and scopes.
6. Developing a code generator that produces three-address code from a simple high-level programming language.
7. Implementing an optimization pass to perform dead code elimination or constant folding on intermediate code.

Suggested Readings

1. "Compiler Design: Principles, Techniques and Tools", 2nd Ed., Prentice-Hall, 2006 by AV Aho, MS Lam, R Sethi, JD Ullman.
2. "Modern Compiler Implementation in Java", Cambridge University Press, 2002 by AW Appel, J Palsberg.
3. "Modern Compiler Implementation in C", Cambridge University Press, 2004 by AW Appel, M Ginsburg.
4. "Engineering a Compiler", 2nd Ed., Morgan Kaufmann, 2011 by K Cooper, L Torczon.
5. "Compiler Construction: Principles and Practice", Cengage Learning, 1997 by KC Louden.
6. "Modern Compiler Design", Wiley, 2000 by D Grune, H Bal, C Jacobs, K. Langendoen.
7. "Programming Language Pragmatics", 3rd Ed., Morgan Kaufmann, 2009 by Michael L Scott.
8. "Advanced Compiler Design and Implementation", Morgan Kaufmann/Elsevier(India), 2003 by S Muchnick.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Department of Computer Science and Engineering
Faculty of Technology
University of Delhi

List of Discipline Specific Elective (DSE) / Generic Elective (GE) courses offered for Minors / Specializations by the department in Third Year

1. **Minor in CSE (Offered to ECE and EE)**
    a. DSE-3 / GE-5: Foundations of Computer Networks
    b. DSE-4 / GE-6: Fundamentals of Software Engineering

2. **Minor/Specialization in Artificial Intelligence and Machine Learning (Offered to CSE, ECE and EE)**
    a. DSE-3 / GE-5: Data Mining & Warehousing
    b. DSE-3 / GE-5: Neural Networks
    c. DSE-4 / GE-6: AI for Image Processing
    d. DSE-4 / GE-6: NLP: Techniques and Applications

3. **Minor/Specialization in Data Science (Offered to CSE, ECE and EE)**
    a. DSE-3 / GE-5: Data Mining & Warehousing
    b. DSE-3 / GE-5: Advanced Data Analytics
    c. DSE-4 / GE-6: Health Data Analytics
    d. DSE-4 / GE-6: Fundamentals of Time Series Analysis

4. **Minor/Specialization in Software Engineering (Offered to CSE, ECE and EE)**
    a. DSE-3 / GE-5: Predictive Modeling
    b. DSE-3 / GE-5: Software Testing
    c. DSE-4 / GE-6: Agile Software Development
    d. DSE-4 / GE-6: Software Reliability

5. **Minor/Specialization in Blockchain and Cybersecurity (Offered to CSE, ECE and EE)**
    a. DSE-3 / GE-5: Blockchain Essentials
    b. DSE-3 / GE-5: Ethical Hacking and Advanced Cybersecurity
    c. DSE-4 / GE-6: Blockchain Applications
    d. DSE-4 / GE-6: Cybersecurity with Blockchain

6. **Minor/Specialization in Augmented Reality / Virtual Reality (Offered to CSE, ECE and EE)**
    a. DSE-3 / GE-5: Human Computer Interaction
    b. DSE-3 / GE-5: Game Development
    c. DSE-4 / GE-6: Advanced Game Development
    d. DSE-4 / GE-6: Augmented and Virtual Reality Systems Design

Department of Computer Science and Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Discipline Specific Elective (DSE) / Generic Elective (GE) courses offered for Minors / Specializations by the department in Semester V

Foundations of Computer Networks (DSE-3/GE-5)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Foundations of Computer Networks	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes
Upon completion of this course, students will be able to:
1. Understand and explain the various layers of computer networks and the protocols used in each layer.
2. Analyze network performance using key metrics and troubleshoot networking issues.
3. Demonstrate the ability to design and implement network solutions.
4. Apply networking concepts through practical experiments in simulated environments.

Course Objectives
1. To introduce the fundamental concepts of computer networks and communication systems.
2. To provide an understanding of the network architecture, protocols, and layers.
3. To examine various types of networks and their components, including physical media, protocols, and network devices.

UNIT I
Networking Fundamentals: Introduction to the concepts of computer networking, Types of networks: LAN, WAN, MAN, PAN, Networking topologies, protocols, and models, Introduction to the OSI and TCP/IP models, Basics of IP addressing and subnetting

UNIT II
Data Link Layer and Network Layer: Functions of the data link layer: framing, error control, flow control, MAC protocols: CSMA/CD, CSMA/CA, Network layer functionalities: routing, addressing, packet forwarding, Introduction to IPv4 and IPv6, Basic concepts of routing protocols (RIP, OSPF, BGP)

UNIT III
Transport Layer and Application Layer: Principles of transport layer protocols: TCP and UDP, Mechanisms of congestion control and error handling, Application layer protocols and services: HTTP, DNS, SMTP, FTP, Principles of network security: encryption, firewalls, VPNs

UNIT IV

Network Management and Emerging Technologies: Basics of network management: SNMP, network configuration, and troubleshooting, Wireless networking technologies: WiFi, Bluetooth, LTEIntroduction to network virtualization and SDN (Software Defined Networking), Future trends in networking: IoT, 5G, cloud networking

Practical Component

1. Configuring a small network with routers and switches.
2. Performing network traffic analysis using packet sniffing tools.
3. Setting up and configuring a VPN.
4. Implementing basic secure network communication protocol configuration on a firewall.

Suggested Readings

1. "Networking All-in-One For Dummies" by Doug Lowe
2. "CCNA Routing and Switching 200-125 Official Cert Guide Library" by Wendell Odom
3. "Data Communications and Networking" by Behrouz A. Forouzan
4. "Computer Networks" by Andrew S. Tanenbaum and David J. Wetherall (5th Edition, Pearson)
5. "Computer Networking: A Top-Down Approach" by James Kurose and Keith Ross (6th Edition, Pearson)
6. "Data and Computer Communications" by William Stallings (10th Edition, Pearson)
7. "Network Security Essentials: Applications and Standards" by William Stallings

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Data Mining & Warehousing (DSE-3/GE-5)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Data Mining & Warehousing	4	3L	1T	0P	-	Database Management System

Course Hours: L: 03 T: 01 P: 00

Course Outcomes

At the end of this course, students will be able to:

1. Understand the core concepts, architectures, and data modeling techniques of data warehousing, and relate them to OLAP operations and data cube technology.
2. Apply data preprocessing methods, association rule mining, classification, and clustering algorithms to discover actionable insights from large, complex datasets.
3. Evaluate various clustering techniques, including partitioning, density-based, and hierarchical methods, and assess their effectiveness in segmenting data.

Course Objectives

1. Impart a thorough understanding of data warehousing concepts, architectures, and data modeling methods, including star and snowflake schemas, as well as the ETL processes and OLAP operations.
2. Familiarize students with fundamental data mining concepts and techniques, enabling them to handle and analyze large datasets effectively.
3. Highlight the importance of mining complex data types (text, web, time series) and familiarize students with modern tools and techniques required to extract meaningful patterns and knowledge.
4. Expose students to real-world business applications, case studies, and current trends in data warehousing and data mining.

UNIT I

Introduction to Data Warehousing: Concepts and architecture of a data warehouse, Data modeling for data warehouses: star schema, snowflake schema, ETL processes: extraction, transformation, loading, OLAP in data warehouse, Data Cube Technology, From Data Warehousing to Data Mining

UNIT II

Data Mining Concepts: Overview of data mining and knowledge discovery in databases, Data preprocessing: cleaning, normalization, transformation, Association rule mining: Apriori, FP-growth

Data Mining Techniques:Classification techniques: decision trees, Naive Bayes, KNN, SVM, random forests, Prediction, Classifier Performance measures, Clustering methods:

UNIT III

Cluster Analysis in Data Mining: Types of Data in Cluster Analysis. A Categorization of Major Clustering Methods, Partitioning Methods, Density Based Methods, Grid Based Methods; Model Based Clustering Methods, k-means clustering, hierarchical clustering, DBSCAN, Outlier Analysis

Mining Complex Data: Text mining and natural language processing, Mining Time Series and Sequence Data, Web mining: content mining, structure mining, usage mining, Ethical issues and privacy concerns in data mining

UNIT IV

Real-world Applications of Data Mining and Warehousing: Business intelligence and decision support systems, Data mining in retail, finance, healthcare, and social media, Recommender systems and market basket analysis, Case studies on successful data mining projects, Future trends in data mining and data warehousing

List of tutorial tasks

Note: The course instructor will design tasks to complete the tutorial component of the course.

Suggested Readings

1. "Data Mining: Concepts and Techniques" by Jiawei Han, Micheline Kamber, and Jian Pei
2. "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling" by Ralph Kimball and Margy Ross
3. "Data Mining Introductory & Advanced Topics" by M.H. Dunham, Pearson Education
4. "Data Warehousing Fundamentals" by P. Ponnian, John Wiley.
5. "Master in Data Mining" by M. Berry, G. Linoff, John Wiley
6. "Big Data: Principles and Best Practices of Scalable Realtime Data Systems" by Nathan Marz and James Warren

Neural Networks (DSE-3/GE-5)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Neural Networks	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes
At the end of this course, students will be able to:

1. Understand the architecture and functioning of neural networks, including perceptrons and multi-layer perceptrons.
2. Implement neural network models using backpropagation and gradient descent optimization techniques.
3. Apply Convolutional Neural Networks (CNNs) to solve problems related to image classification and object detection.
4. Design and implement Recurrent Neural Networks (RNNs) for sequence-based tasks like time-series forecasting and language modeling.
5. Develop deep learning models using frameworks like TensorFlow or PyTorch.
6. Evaluate the performance of neural network models through various metrics and techniques, such as cross-validation.
7. Understand the challenges associated with training deep learning models and techniques for handling them, including regularization and dropout.

Course Objectives

1. Introduce students to the fundamental concepts of neural networks and deep learning.
2. Teach students to design, implement, and train simple neural network architectures.
3. Equip students with practical experience in building CNNs and applying them to image processing tasks.
4. Introduce RNNs and demonstrate their use in sequence prediction tasks.
5. Provide hands-on experience in using deep learning frameworks and libraries for model implementation.
6. Teach students how to evaluate and improve the performance of neural network models.
7. Address the common challenges faced in training deep learning models, including overfitting and optimization.

Unit 1: Introduction to Neural Networks
Overview of neural networks and their role in artificial intelligence. Biological inspiration and comparison to artificial neurons. Basic structure of neural networks: perceptrons, activation functions (sigmoid, ReLU, tanh), and feedforward architectures. Fundamentals of supervised and unsupervised learning. Applications of neural networks in classification, regression, and clustering tasks.

Unit 2: Training Neural Networks
Backpropagation and gradient descent for weight optimization. Loss functions (mean squared error, cross-entropy) and performance metrics. Hyperparameter tuning, including learning rate, batch size, and epochs. Regularization techniques to prevent overfitting: L1/L2 regularization, dropout, and early stopping. Introduction to optimizers like SGD, Adam, and RMSprop. Challenges in training: vanishing and exploding gradients.

Unit 3: Advanced Architectures and Concepts

Deep neural networks (DNNs) and their layers: convolutional layers, pooling layers, and fully connected layers. Introduction to convolutional neural networks (CNNs) for image processing and recurrent neural networks (RNNs) for sequence prediction. Understanding long short-term memory (LSTM) and gated recurrent unit (GRU) networks. Transfer learning and pre-trained models. Generative adversarial networks (GANs) for data synthesis.

Unit 4: Applications and Future Directions

Practical applications of neural networks: computer vision, natural language processing, robotics, and healthcare. Neural networks in recommendation systems, autonomous vehicles, and financial forecasting. Introduction to reinforcement learning and neural networks' role in AI decision-making. Ethical considerations and challenges in deploying neural network models. Future trends: spiking neural networks and neuromorphic computing.

Practical Component

1. Implementing a simple neural network from scratch to solve a classification problem (e.g., XOR problem).
2. Developing and training a Convolutional Neural Network (CNN) for image classification using TensorFlow/Keras.
3. Implementing a Recurrent Neural Network (RNN) for sequence prediction tasks like time-series forecasting.
4. Fine-tuning a pre-trained neural network model for transfer learning tasks.
5. Developing an image generation model using Generative Adversarial Networks (GANs).
6. Evaluating and improving neural network performance using techniques such as dropout and data augmentation.
7. Experimenting with hyperparameter optimization and model evaluation to enhance the performance of deep learning models.

Suggested Readings

1. "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
2. "Transformers for Natural Language Processing" by Denis Rothman
3. "Attention Is All You Need" by Vaswani et al., seminal paper introducing transformers
4. "GPT-3: Language Models are Few-Shot Learners" by Brown et al., detailing the architecture and capabilities of GPT-3

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Advanced Data Analytics (DSE-3/GE-5)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Advanced Data Analytics	4	3L	0T	1P	-	Fundamentals of Data Analytics

Course Hours: L: 03 T: 00 P: 02

Course Outcomes
At the end of this course, students will be able to:

1. Understand the fundamental concepts and techniques of data analytics, including data preprocessing, statistical modeling, and exploratory data analysis.
2. Develop predictive analytics models using various machine learning algorithms and evaluate their performance for informed decision-making.
3. Apply advanced analytics techniques such as clustering, dimensionality reduction, time series forecasting, and text mining to extract meaningful insights from complex datasets.
4. Understand and implement big data analytics principles and tools to handle large-scale and unstructured data efficiently.

Course Objectives

1. Equip students with a strong foundation in data preprocessing, statistical analysis, and exploratory data analytics.
2. Enable the application of advanced modeling, machine learning, and big data techniques for deriving actionable insights from diverse data sources.
3. Instill practical understanding, and future-oriented perspectives by examining real-world applications and emerging trends.

Unit 1: Data Analytics Fundamentals
Introduction to Data Analytics: Role, importance, and applications in various sectors (finance, healthcare, marketing, etc.). Data Types and Quality: Structured, semi-structured, and unstructured data; issues of missing values, noise, outliers. Data Preprocessing: Cleaning, normalization, dimensionality reduction (overview), and feature engineering. Statistical Foundations: Probability distributions, hypothesis testing, confidence intervals, correlation, and ANOVA. Exploratory Data Analysis (EDA) and Visualization: Summary statistics, histograms, box plots, scatter plots, heat maps, and dashboards using tools like Matplotlib, Seaborn, Plotly, Tableau.

Unit 2: Predictive Analytics and Modeling
Regression Techniques: Linear regression (simple, multiple), regularization methods (Ridge, Lasso), logistic regression for classification. Tree-based Models: Decision trees, ensemble methods (random forests, gradient boosting). Support Vector Machines (SVM) and Neural Networks: Introduction to kernel methods, perceptrons, feedforward networks. Model Evaluation: Train-test splits, cross-validation, confusion matrix, ROC curves, AUC, precision, recall, F1-score. Model Selection and Tuning: Hyperparameter optimization (grid search, random search, Bayesian optimization). Formulation of predictive tasks from business problems, Translation of business problems into analytical models.

Unit 3: Advanced Analytics Techniques

Unsupervised Learning: Clustering methods (k-means, hierarchical clustering, DBSCAN), principal component analysis (PCA) for dimensionality reduction. Time Series Analysis and Forecasting: ARIMA, seasonal decomposition, exponential smoothing, and introduction to LSTM-based forecasting. Text Mining and Sentiment Analysis: Text preprocessing (tokenization, stemming, lemmatization), topic modeling, sentiment classification. Hadoop ecosystem, Spark for large-scale analytics, NoSQL databases.

Unit 4: Data Analytics in Practice

Ethical Considerations: Privacy, bias, fairness, transparency, and interpretability in analytical models. Industry-Specific Applications, financial analytics, healthcare analytics. Case Studies: Examination of real-world projects showcasing the end-to-end data analytics lifecycle.

Practical Component

1. Conducting exploratory data analysis using a popular data analysis tool or programming language.
2. Building and evaluating predictive models for a given dataset.
3. Implementing clustering techniques on a dataset to uncover patterns.
4. Implementing time series forecasting models on a time-series dataset.
5. Extract insights from textual data by performing sentiment analysis and topic modeling.
6. Work with a domain-specific dataset to build an end-to-end analytics solution - data ingestion, preprocessing, modeling, and insights presentation.

Suggested Readings

1. "Data Mining: Concepts and Techniques" by Jiawei Han, Micheline Kamber, Jian Pei
2. "Applied Predictive Modeling" by Max Kuhn, Kjell Johnson
3. "Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking" by Foster Provost and Tom Fawcett
4. "Practical Statistics for Data Scientists: 50 Essential Concepts" by Peter Bruce and Andrew Bruce
5. "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython" by Wes McKinney

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Predictive Modeling (DSE-3/GE-5)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Predictive Modeling	4	3L	1T	0P	-	Software Engineering

Course Hours: L: 03 T: 01 P: 00

Course Outcomes

By the end of the course, students will be able to:

1. Understand the principles and significance of predictive modeling in software engineering.
2. Design and implement predictive models for software engineering problems using empirical methods.
3. Evaluate predictive models using statistical and performance metrics.
4. Apply machine learning and data-driven approaches to improve software quality and project outcomes.

Course Objectives

1. To understand the fundamentals of predictive modeling in the context of software engineering.
2. To explore the application of machine learning techniques to predict software quality attributes.
3. To evaluate the performance of predictive models using appropriate metrics.
4. To introduce recent trends and advancements in predictive modeling for software development and maintenance.

UNIT I

Fundamentals of Predictive Modelling in Software Engineering: Introduction to predictive modeling in software engineering, Overview of empirical methods in software engineering research, Data collection and preprocessing for predictive modeling, Software engineering metrics: product, process, and resource metrics, Applications of predictive modeling in defect prediction, effort estimation, and maintenance

UNIT II

Machine Learning Techniques for Predictive Modelling: Supervised learning techniques: Linear regression, logistic regression, and decision trees, Support Vector Machines (SVMs) and ensemble methods (e.g., Random Forests, Gradient Boosting), Unsupervised learning techniques: Clustering and dimensionality reduction, Neural networks and deep learning for software engineering data, Feature selection and engineering for improving model performance, Case studies: Defect prediction using classification models

UNIT III

Model Evaluation and Validation: Metrics for predictive model evaluation: Accuracy, precision, recall, F1-score, Mean Absolute Error (MAE), Mean Squared Error (MSE), and RMSE, Cross-validation techniques (e.g., k-fold, leave-one-out), Handling imbalanced datasets in software engineering (e.g., SMOTE), Overfitting and underfitting in predictive models, Tools and frameworks for predictive modeling in Python (e.g., scikit-learn, Weka)

UNIT IV

Recent Trends and Advanced Topics: Explainable AI (XAI) in software predictive modeling, Predictive modeling for agile and DevOps practices, Effort estimation using advanced ML models (e.g., Bayesian models, transformer architectures), Applications in software evolution and maintenance (e.g., predicting change impact), Emerging trends: Transfer learning and meta-learning for software engineering data

Tutorial Component

1. Hands-on experiments with defect prediction datasets (e.g., PROMISE repository).
2. Feature selection and extraction from software metrics data.
3. Implementation of predictive models using Python or Weka.
4. Evaluation of model performance using cross-validation and statistical metrics.
5. Group activity: Developing a predictive model for a software engineering problem (e.g., effort estimation).

List of tutorial tasks

Note: The course instructor will design tasks to complete the tutorial component of the course.

Suggested Readings

1. "Empirical Software Engineering" by Ruchika Malhotra
2. "Data Mining and Predictive Analytics" by Daniel T. Larose and Chantal D. Larose
3. Research papers and case studies on predictive modeling in software engineering

Software Testing (DSE-3/GE-5)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Software Testing	4	3L	0T	1P	-	Software Engineering

Course Hours: L: 03 T: 00 P: 02

Course Outcomes

By the end of the course, students will be able to:

1. Explain key software testing concepts, principles, and processes.
2. Design test plans, test cases, and scenarios for different types of software applications.
3. Use automated testing tools to improve efficiency and accuracy.
4. Analyze software quality using appropriate metrics and standards.
5. Collaborate in teams to conduct testing and quality assurance activities effectively.
6. Understand where key testing concepts apply within the context of unified processes.

Course Objectives

1. To understand fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
2. To Identify a number of test styles and techniques and assess their usefulness in a particular context.
3. To discuss various software testing issues and solutions in software.
4. To learn how to plan a test project, design test cases and data, conduct testing operations, manage software problems and defects.

UNIT I

Fundamentals of Software Testing: Introduction to software testing and its importance, Testing Process, Limitations of Testing, Testing activities, Testing levels: unit testing, integration testing, system testing, acceptance testing, debugging, Performance testing, load testing, stress testing, regression testing

UNIT II

Verification Testing: Verification Methods, SRS Verification, Software Design Document Verification, Code Reviews, User Documentation Verification, Software Project Audits**.** White-box testing, black-box testing, and gray-box testing

Functional Testing: Boundary Value Analysis, Equivalence Class Testing, Decision Table Based Testing, Cause Effect Graphing Technique**.**

UNIT III

Structural Testing: Path testing, DD-Paths, Cyclomatic Complexity, Graph Metrics, Data Flow Testing, Mutation testing. Object Oriented Testing: Class Testing, GUI Testing.

Software Testing Tools: Defect life cycle and defect tracking tools (e.g., JIRA, Bugzilla), Fundamentals of automation testing, Benefits and challenges of automation testing, Methodology to evaluate automated testing, Testing Tools, Java Testing Tools, JMetra, JUNIT Cactus and ot

UNIT IV

Advanced Topics in Software Testing: Security testing, Usability testing, Testing in Agile and DevOps environments, Emerging trends: AI/ML in testing, Mobile app testing, Cloud-based testing

Practical Component

1. Developing and executing test cases for a software application.
2. Executing black-box and white-box testing techniques
3. Implementing automated tests using a testing framework.
4. Conducting performance testing on a web application.
5. Participating in a team project to manage and document the testing process for a software release.

Suggested Readings

1. Yogesh Singh, “Software Testing”, 1st Ed., Cambridge University Press.
2. Paul C. Jorgenson, Software Testing A Craftsman’s approach, CRC Press.
3. "Software Testing: Principles and Practices" by Srinivasan Desikan and Gopalaswamy Ramesh
4. Louise Tamres, “Software Testing”, Pearson Education Asia.
5. "The Art of Software Testing" by Glenford J. Myers, Corey Sandler, and Tom Badgett
6. "Agile Testing: A Practical Guide for Testers and Agile Teams" by Lisa Crispin and Janet Gregory

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Blockchain Essentials (DSE-3/GE-5)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Blockchain Essentials	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes

At the end of this course, students will be able to:

1. Understand the fundamental principles and cryptographic underpinnings of blockchain technology, and differentiate between various types of blockchain architectures.
2. Evaluate and select appropriate blockchain platforms and frameworks, and develop, deploy, and test smart contracts and decentralized applications (DApps).
3. Analyze real-world use cases of blockchain across multiple industries, identify regulatory and ethical challenges, and propose secure and compliant blockchain solutions.

Course Objectives

1. Introduce the foundational concepts, architecture, and cryptographic components of blockchain technology.
2. Provide hands-on exposure to blockchain platforms, enabling students to design, code, and deploy smart contracts and decentralized applications.
3. Examine practical industry applications of blockchain in finance, supply chain, healthcare, and government, and understand associated regulatory and ethical considerations.

Unit 1: Blockchain Fundamentals

Introduction to blockchain technology and its significance, Blockchain architecture: blocks, chains, consensus mechanisms, Types of blockchains: public, private, consortium, Cryptography in blockchain: hash functions, public key cryptography

Unit 2: Blockchain Platforms and Development

Overview of popular blockchain platforms: Ethereum, Hyperledger, Bitcoin, Smart contracts: definition, development, and deployment, Development tools and environments for blockchain, Building decentralized applications (DApps)

Unit 3: Blockchain in Industry

Blockchain applications in finance: cryptocurrencies, ICOs, DeFi, Blockchain for supply chain management, healthcare, and government, Regulatory and ethical considerations of blockchain technology, Security challenges and solutions in blockchain systems

Unit 4: Emerging Trends in Blockchain

Advances in consensus mechanisms: Proof of Stake, Delegated Proof of Stake, Proof of Authority, Scalability solutions: sidechains, sharding, layer 2 protocols, Interoperability among blockchain systems, Future directions and potential impacts of blockchain technology

Practical Component

1. Creating a simple smart contract and deploying it on a blockchain platform.
2. Developing basic DApp(s) using Ethereum and Solidity.
3. Analyzing blockchain use cases and proposing a solution for a real-world problem.

Suggested Readings

1. "Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications" by Lorne Lantz and Daniel Cawrey
2. "Blockchain Basics: A Non-Technical Introduction in 25 Steps" by Daniel Drescher
3. "Blockchain Applications: A Hands-On Approach" by Arshdeep Bahga and Vijay Madisetti

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Ethical Hacking and Advanced Cybersecurity (DSE-3/GE-5)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Ethical Hacking and Advanced Cybersecurity	4	3L	0T	1P	-	Fundamentals of Cybersecurity

Course Hours: L: 03 T: 00 P: 02

Course Outcomes

At the end of this course, students will be able to:

1. Understand and apply the principles of ethical hacking and penetration testing.
2. Detect and exploit vulnerabilities in a controlled and responsible manner.
3. Utilize advanced tools and frameworks for ethical hacking and network defense.
4. Conduct forensic investigations and malware analysis to identify the root causes of breaches.
5. Formulate and implement advanced cybersecurity strategies to protect critical infrastructure.

Course Objectives

1. Introduce ethical hacking techniques and practices to identify vulnerabilities in systems and networks.
2. Provide insights into advanced cybersecurity measures, including malware analysis and threat hunting.
3. Equip students with tools for ethical hacking, such as Kali Linux, Metasploit, and Burp Suite.
4. Develop ethical responsibility and compliance with legal frameworks in cybersecurity.
5. Enable hands-on experience in simulating and defending against sophisticated cyber-attacks.

Unit 1: Fundamentals of Ethical Hacking

Definition, Scope, and Importance of Ethical Hacking; Understanding Cyber Laws and Ethical Hacking Ethics; Types of Hackers (White Hat, Black Hat, Grey Hat); Phases of Ethical Hacking (Reconnaissance, Scanning, Gaining Access, Maintaining Access, Covering Tracks); Introduction to Ethical Hacking Tools such as Kali Linux, Metasploit, and Wireshark.

Unit 2: Network and System Security

Overview of Network Security: Firewalls, IDS/IPS, VPNs, Proxy Servers; Vulnerability Assessment and Penetration Testing (VAPT) Techniques and Tools; Network Hacking: Packet Sniffing, Spoofing, and Man-in-the-Middle (MITM) Attacks; System Security Concepts: Privilege Escalation, Backdoors, and Trojans; Introduction to Wireless Network Security and WPA/WPA2 Cracking.

Unit 3: Application and Web Security

Understanding Web Application Attacks: SQL Injection, Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF); Securing Web Applications: OWASP Top 10 Framework; Exploring Secure Coding Practices; Hacking APIs and Web Services; Introduction to Mobile Application Security.

Unit 4: Advanced Cybersecurity Concepts

Incident Detection and Response: Monitoring and Analyzing Cyber Attacks; Cryptographic Techniques: Encryption Standards, Digital Signatures, and Certificates; Advanced Persistent Threats (APTs): Detection and Mitigation

Strategies; Exploring Blockchain Security and Zero-Trust Architecture; Overview of Emerging Threats in AI and IoT.

Practical Component

1. Conducting reconnaissance using tools like Maltego to gather intelligence on a target system.
2. Exploiting vulnerabilities in a controlled lab environment to understand the ethical hacking lifecycle.
3. Developing exploits using custom scripts and testing them in a sandboxed environment.
4. Performing secure code analysis to identify common vulnerabilities like SQL injection or XSS.
5. Building and deploying honeypots to monitor and log potential attack attempts.
6. Simulating ransomware attacks and crafting effective incident response strategies.
7. Conducting security audits of cloud platforms to ensure compliance with security standards.

Suggested Readings

1. "The Hacker Playbook 3: Practical Guide To Penetration Testing" by Peter Kim
2. "Hacking: The Art of Exploitation" by Jon Erickson
3. "Penetration Testing: A Hands-On Introduction to Hacking" by Georgia Weidman
4. Hacking: The Art of Exploitation by Jon Erickson (No Starch Press)
5. Penetration Testing: A Hands-On Introduction to Hacking by Georgia Weidman (No Starch Press)
6. The Web Application Hacker's Handbook by Dafydd Stuttard and Marcus Pinto (Wiley)
7. Metasploit: The Penetration Tester’s Guide by David Kennedy, Jim O’Gorman, Devon Kearns, and Mati Aharoni (No Starch Press)

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Human Computer Interaction (DSE-3/GE-5)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Human Computer Interaction	4	3L	1T	0P	-	-

Course Hours: L: 03 T: 01 P: 00

Course Outcomes
At the end of this course, students will be able to:

1. Understand the fundamental theories, principles, and concepts of HCI, and apply user-centered design methodologies to create intuitive interfaces.
2. Employ interaction design principles, prototyping techniques, and accessibility guidelines to design effective interfaces for various platforms and devices.
3. Plan, conduct, and analyze usability evaluations to assess user experience, iteratively refining designs based on qualitative and quantitative feedback.

Course Objectives

1. Introduce students to the core principles, history, and cognitive frameworks underlying human-computer interaction.
2. Provide hands-on experience in interaction design, prototyping, and the implementation of user-centered and accessible interfaces.
3. Train students in usability evaluation methods, enabling them to measure and improve the quality of user experiences through iterative design.

Unit 1: Introduction to HCI
Fundamentals of HCI: Human capabilities, interaction paradigms, and interface metaphors. History and Evolution: From command-line interfaces to GUIs, multi-touch, and beyond. User-Centered Design (UCD): Key concepts, user involvement, requirement gathering, and iterative development. Cognitive Models and UI Design: Mental models, GOMS, cognitive load, and perception principles. Evaluating User Experience (UX): Metrics, qualitative and quantitative measures, satisfaction, efficiency, and effectiveness.

Unit 2: Designing Interactive Systems
Interaction Design Basics: Affordances, signifiers, feedback loops, constraints, and mapping. Design Thinking and Prototyping: Design research, ideation techniques, low-fi and hi-fi prototyping, wireframing, and storyboarding. User Interface Design Principles: Consistency, visibility, learnability, error prevention, standards. Designing for web, mobile, and desktop. Accessibility and Inclusive Design: WCAG guidelines, designing for diverse users, assistive technologies, and inclusive user scenarios. Tools and Technologies: UI/UX design software (Sketch, Figma, Adobe XD), prototyping tools, style guides, and design systems.

Unit 3: Usability Evaluation and Testing
Usability concepts and heuristics, Planning and conducting usability tests, Quantitative and qualitative methods for evaluating HCI, Analyzing and interpreting test results, Iterative design and usability engineering

Unit 4: Advanced Topics in HCI

Emotional design and user engagement, Voice user interfaces and conversational design, Augmented reality (AR) and virtual reality (VR) in HCI, Future trends in HCI: AI, IoT, and wearable technologies, Ethical considerations in HCI design and research

Tutorial Component

1. Conduct interviews or surveys to identify user needs, create user personas, and draft preliminary requirements.
2. Develop low-fidelity wireframes and interactive prototypes for a given application scenario using tools like Figma or Sketch.
3. Apply Nielsen's heuristics to evaluate a prototype or an existing interface, identify usability issues, and propose improvements.
4. Conduct a small-scale usability test with participants on a prototype, use think-aloud protocols, record observations, and summarize findings.
5. Experiment with a simple voice interface (using existing frameworks) or explore a basic AR scenario, observing user interaction.
6. Developing an accessible web or mobile application.

Suggested Readings

1. "Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability" by Steve Krug
2. "The Design of Everyday Things" by Don Norman
3. "About Face: The Essentials of Interaction Design" by Alan Cooper, Robert Reimann, David Cronin, and Christopher Noessel
4. "Interaction Design: Beyond Human-Computer Interaction" by Jenny Preece, Helen Sharp, Yvonne Rogers

Game Development (DSE-3/GE-5)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Game Development	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes

At the end of this course, students will be able to:

1. Understand the fundamental concepts, history, and processes involved in developing video games, from initial design to final deployment.
2. Apply game design principles to create engaging game worlds, narratives, levels, and characters suitable for various platforms and genres.
3. Develop and implement core gameplay functionalities using modern game engines, programming languages, graphics, physics, and AI techniques.

Course Objectives

1. Introduce the foundational principles of game development, including game genres, platforms, and industry-standard tools.
2. Cultivate skills in game design, storytelling, character creation, and level design, emphasizing user experience and engagement.
3. Provide technical proficiency in programming, graphics rendering, physics simulation, and AI, enabling students to implement game features effectively.

Unit 1: Introduction to Game Development

Overview of Game Development Process: Conceptualization, prototyping, production, polishing, and publishing. History and Evolution of Video Games: Arcade classics, console generations, PC gaming evolution, mobile and VR innovations. Game Genres and Platforms: Action, adventure, RPG, simulation, strategy, mobile vs. console vs. PC, VR/AR platforms. Introduction to Game Design Principles: Player-centered design, core gameplay loops. Game Engines and Tools: Introduction to Unity, Unreal Engine; asset creation tools (Blender, Maya); version control systems (Git).

Unit 2: Game Design and Storytelling

Elements of game design: mechanics, dynamics, aesthetics, Narrative and storytelling in games, Character development and world-building, Level design principles, Designing for different platforms: PC, consoles, mobile, VR/AR

Unit 3: Game Programming

Introduction to programming languages used in game development (C#, C++ etc), Game engine architecture and components, Graphics programming: rendering, shaders, particle systems, Physics and collision detection, AI in games: pathfinding, decision-making, NPC behaviors

Unit 4: Game Production and Post-Production

Project management in game development: Agile and Scrum methodologies, Quality assurance, testing strategies, and debugging, Marketing and monetization strategies for games, Post-launch support and community management, Emerging trends and future directions in game development

Practical Component

1. Designing a game concept and creating a design document.
2. Developing a simple game using a chosen game engine.
3. Implementing key game mechanics and programming AI behaviors.
4. Testing and debugging the game, followed by a presentation of the project.

Suggested Readings

1. "The Art of Game Design: A Book of Lenses" by Jesse Schell
2. "Rules of Play: Game Design Fundamentals" by Katie Salen and Eric Zimmerman
3. "Game Programming Patterns" by Robert Nystrom
4. "Unity in Action: Multiplatform Game Development in C#" by Joe Hocking
5. "Learning C++ by Creating Games with Unreal Engine 4" by Sharan Volin

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Department of Computer Science and Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Discipline Specific Elective (DSE) / Generic Elective (GE) courses offered for Minors / Specializations by the department in Semester VI

Fundamentals of Software Engineering (DSE-4/GE-6)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Fundamentals of Software Engineering	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes
At the end of this course, students will be able to:
1. Understand the fundamental concepts, principles, and models of software engineering and apply them to plan, develop, and manage software projects.
2. Analyze and specify software requirements using structured techniques and UML, ensuring that both functional and non-functional requirements are met.
3. Employ software design principles, coding standards, and quality measures to produce maintainable and efficient software.
4. Estimate project costs, identify and manage risks, and apply appropriate testing methodologies to ensure the reliability and quality of software systems.
5. Comprehend the processes of software maintenance, configuration management, and stay informed about emerging trends to adapt to evolving industry practices.

Course Objectives
1. Introduce students to the fundamental concepts of software engineering, including SDLC models, software metrics, and project management fundamentals.
2. Enable students to elicit, analyze, and document software requirements, and to understand various software design paradigms and coding best practices.
3. Familiarize students with project planning, scheduling, cost estimation models, and risk management strategies to improve software development effectiveness.
4. Develop an understanding of software testing techniques and levels, ensuring software reliability, performance, and adherence to specifications.

UNIT I
Introduction to Software Engineering: Definition and importance of software engineering, Software characteristics, Software components, Software applications, Software Engineering Principles, Software metrics and measurement,

monitoring and control, software development life cycle (SDLC) models: Waterfall, Agile, Spiral, DevOps, Software project management fundamentals

UNIT II

Software Requirements and Design: Requirements elicitation techniques, requirements analysis, requirements specification, Functional and non-functional requirements, Introduction to Unified Modeling Language (UML)
Software Design and Development: Programming paradigms: procedural, object-oriented, functional, logic; Code quality, Cohesiveness and Coupling, coding standards, and code reviews

UNIT III

Software Project Planning: Project planning and Project scheduling. Software Metrics: Size Metrics like LOC, Token Count, Function Count. Cost estimation using models like COCOMO. Risk management activities.
Testing: Verification and validation, Level of testing: Unit, Integration Testing, Top down and bottom up integration testing, Alpha and Beta testing, functional testing, structural testing.

UNIT IV

Software Maintenance and Advanced Topics: Software maintenance and evolution, Software configuration management, Risk management in software projects, Emerging trends in software engineering

Practical Component

1. Participating in a team project to develop a software application following a chosen SDLC model.
2. Conducting requirements analysis and creating a UML diagram for a system.
3. Implementing a software module and performing unit testing.
4. Reviewing peers' code for quality and adherence to coding standards.

Suggested Readings

1. K. K. Aggarwal & Yogesh Singh, “Software Engineering”, 2 ndEd., New Age International.
2. Software Engineering" by Ian Sommerville
3. R. S. Pressman, “Software Engineering – A practitioner’s approach”, 3rd ed., McGraw Hill Int. Ed.
4. "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides
5. "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin
6. "The Pragmatic Programmer: Your Journey to Mastery" by Andrew Hunt and David Thomas

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

AI for Image Processing (DSE-4/GE-6)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
AI for Image Processing	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Objectives

1. Provide a solid foundation in the concepts and techniques of image processing using AI.
2. Equip students with skills to develop and optimize image recognition and analysis algorithms.
3. Introduce state-of-the-art neural networks like CNNs for image classification, segmentation, and object detection.
4. Develop an understanding of deep learning frameworks and libraries like TensorFlow and PyTorch.
5. Encourage innovative applications of AI in fields such as healthcare, autonomous systems, and media.

Course Outcomes

At the end of this course, students will be able to:

1. Understand the principles of AI and its application in image processing tasks.
2. Design and implement neural networks for image classification, segmentation, and detection.
3. Utilize advanced techniques like transfer learning and data augmentation for improved model performance.
4. Analyze and interpret image datasets for various industrial and research applications.
5. Develop innovative AI solutions for real-world image processing challenges.

Unit 1: Fundamentals of Image Processing

Introduction to Digital Image Processing; Image Representation: Pixels, Resolution, and Color Models; Basics of Image Enhancement Techniques: Filtering, Histogram Equalization, and Contrast Adjustment; Image Transformations: Fourier Transform, Wavelets, and Principal Component Analysis (PCA); Introduction to OpenCV and Image Manipulation Libraries in Python.

Unit 2: Artificial Intelligence and Computer Vision

Overview of AI Techniques for Image Processing; Edge Detection and Object Localization; Image Segmentation Techniques: Thresholding, Clustering, and Region-based Methods; Feature Extraction: SIFT, SURF, and ORB; Deep Learning for Vision Tasks: Introduction to Convolutional Neural Networks (CNNs) and Transfer Learning.

Unit 3: Advanced Image Processing with Deep Learning

Understanding Image Classification and Object Detection Models; Pre-trained Models: VGG, ResNet, Inception, and YOLO; Semantic Segmentation using U-Net and Mask R-CNN; Generative Models for Images: GANs and Variational Autoencoders (VAEs); Image Style Transfer and Super-Resolution Techniques.

Unit 4: Applications of AI in Image Processing

Medical Image Processing: Detection and Diagnosis (CT, MRI, X-rays); Facial Recognition Systems and Emotion Detection; Autonomous Vehicles: Lane Detection, Obstacle Detection, and Tracking; Satellite Image Analysis:

Remote Sensing and Land Use Classification; Ethical Considerations in AI-based Image Processing: Bias, Privacy, and Security Concerns.

Practical Component

1. Implementing basic image preprocessing techniques like resizing, normalization, and augmentation.
2. Designing and training convolutional neural networks (CNNs) for image classification tasks.
3. Developing edge detection algorithms using Sobel, Prewitt, or Canny methods.
4. Implementing object detection models using YOLO, SSD, or Faster R-CNN frameworks.
5. Performing image segmentation using U-Net or Mask R-CNN for medical or satellite imagery.
6. Building style transfer applications to modify images using neural networks.
7. Creating and testing face recognition systems using feature extraction and deep learning models.

Suggested Readings

1. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods (Pearson)
2. Deep Learning for Computer Vision with Python by Adrian Rosebrock (PyImageSearch)
3. Programming Computer Vision with Python by Jan Erik Solem (O'Reilly Media)
4. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (MIT Press)

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

NLP: Techniques and Applications (DSE-4/GE-6)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
NLP: Techniques and Applications	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes

At the end of this course, students will be able to:

1. Understand the foundational principles of Natural Language Processing (NLP), including language modeling and basic text processing techniques.
2. Apply advanced NLP methodologies, such as Transformer architectures and large-scale pretrained language models, to solve complex language tasks.
3. Evaluate and optimize NLP systems using state-of-the-art ML approaches, including few-shot and zero-shot learning, prompt engineering, transfer learning, and in-context learning.

Course Objectives

1. Introduce students to the fundamental concepts and linguistic foundations of NLP.
2. Equip students with knowledge and skills in modern NLP, including deep neural architectures (RNNs, LSTMs, Transformers) and pretrained large language models (BERT, GPT, T5).
3. Train students to select and apply advanced ML techniques such as few-shot learning, transfer learning, and prompt engineering, enabling them to rapidly adapt NLP models to diverse tasks and domains.
4. Expose students to recent developments, including instruction-tuned language models, multi-modal NLP, reinforcement learning for improved model alignment.
5. Encourage critical thinking about the ethical implications of advanced NLP, ensuring responsible development and deployment of AI-driven language systems.

Unit 1: Fundamentals of NLP

Introduction to Natural Language Processing: Scope, challenges, and real-world applications. Linguistic Essentials: Morphology, syntax, semantics, pragmatics, and their relevance to NLP tasks. Text Processing Techniques: Tokenization, stemming, lemmatization, POS tagging, chunking, and dependency parsing. Traditional Language Modeling: N-grams, smoothing techniques, basic probabilistic approaches. Vector Representations: Bag-of-Words, TF-IDF, distributional semantics, introduction to embeddings (Word2Vec, GloVe).

Unit 2: Deep Learning for NLP

Neural Architectures: RNNs, LSTMs, GRUs for sequence modeling, sequence-to-sequence frameworks. Transformers and Attention Mechanisms: Self-attention, encoder-decoder frameworks, and the Transformer architecture. Large Language Models (LLMs): BERT, GPT, T5, and other pretrained models; fine-tuning techniques and transfer learning. Evaluation Metrics and Model Improvement: BLEU, ROUGE, METEOR, perplexity, accuracy; model interpretability and error analysis. Introduction to Prompt Engineering: Harnessing pretrained models through prompts and in-context learning.

Unit 3: Advanced NLP Techniques

Information Extraction and Semantic Understanding: Advanced NER, relation extraction, event detection, coreference resolution, semantic role labeling. Document-Level NLP: Topic modeling (LDA), advanced text classification, sentiment analysis, and stance detection. Few-shot and Zero-shot Learning: Exploiting pretrained models to perform tasks with minimal task-specific data. Advanced Prompting and Instruction-Tuning: Instruction-based interfaces (e.g., InstructGPT), guiding model behavior through task descriptions, Chain-of-Thought prompting for improved reasoning. Optimization and Model Efficiency: Model compression, quantization, distillation, and adapting LLMs for on-device and real-time NLP tasks.

Unit 4: Multimodal NLP and Ethics in NLP

Multi-modal NLP: Integrating text with images, video, and audio (CLIP, PaLM-E etc), bridging language with visual and other sensory modalities. Reinforcement Learning for Alignment: Reinforcement Learning from Human Feedback (RLHF) to refine model behavior, reduce harmful outputs, and improve user alignment.Ethical and Responsible AI in NLP: Bias detection and mitigation, fairness in language models, handling misinformation, and differential privacy techniques.

Practical Component

1. Implement tokenization, POS tagging, and a simple N-gram language model.
2. Train and evaluate RNN/LSTM-based language models, compare perplexity against baseline N-gram models.
3. Fine-tune a pretrained Transformer (BERT or DistilBERT) on a sentiment analysis or QA task.
4. Use GPT or T5 variants with prompt engineering for zero-shot and few-shot text classification.
5. Experiment with instruction-tuned models, apply chain-of-thought prompting to improve reasoning in a QA scenario.
6. Implement a simple text-image retrieval or captioning task using a vision-language model (e.g., CLIP).
7. Apply model distillation techniques to reduce model size and latency on a chosen NLP task.

Suggested Readings

1. "Speech and Language Processing" by Dan Jurafsky and James H. Martin, Prentice Hall.
2. "Natural Language Processing" by Jacob Eisenstein, MIT Press.
3. "Neural Network Methods for Natural Language Processing" by Yoav Goldberg, Morgan & Claypool.
4. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (T5)" by Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu, Journal of Machine Learning Research (JMLR).
5. "OpenAI and DeepMind Research Papers (GPT, BERT, PaLM etc) and recent ACL/NAACL/EMNLP findings" by various authors.
6. "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
7. "Transformers for Natural Language Processing" by Denis Rothman
8. "Attention Is All You Need" by Vaswani et al., seminal paper introducing transformers
9. "GPT-3: Language Models are Few-Shot Learners" by Brown et al., detailing the architecture and capabilities of GPT-3

Health Data Analytics (DSE-4/GE-6)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Health Data Analytics	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes
At the end of this course, students will be able to:
1. Understand the fundamental concepts, data sources, and regulatory frameworks that define health data analytics.
2. Employ statistical and machine learning techniques to preprocess, analyze, and derive insights from diverse healthcare datasets.
3. Integrate advanced methods, such as deep learning, NLP, and genomic data analysis, to address complex clinical decision-making challenges.

Course Objectives
1. Introduce the healthcare data landscape, including EHRs, clinical, imaging, genomic, and sensor data, as well as relevant compliance standards (HIPAA, GDPR).
2. Develop proficiency in data preprocessing, exploratory data analysis, and the application of machine learning models for predicting clinical outcomes and improving care quality.
3. Provide hands-on experience with advanced analytic techniques and interpret model results for informed decision-making.
4. Highlight the importance of privacy, security, and ethical considerations in health analytics, ensuring responsible and patient-centered AI applications.
5. Explore cutting-edge trends, technologies, and frameworks for health analytics and their potential to shape the future of healthcare delivery.

Unit 1: Introduction to Health Data Analytics
Healthcare Data Landscape: Electronic Health Records (EHRs), claims data, medical imaging, wearable sensor data, and genomic data. Stakeholders and Systems: Providers, payers, patients, regulatory bodies, and healthcare IT infrastructure. Healthcare Regulations and Standards: HIPAA, GDPR, FHIR; data integration and interoperability challenges. Data Quality and Preprocessing: Handling missing data, normalization, de-identification, data cleaning techniques. Basic Healthcare Statistics: Epidemiological measures, clinical outcome metrics, and public health indicators.

Unit 2: Techniques for Health Data Preprocessing and Analysis
Exploratory Data Analysis (EDA): Visualizing patient cohorts, stratifying populations by disease burden and risk factors. Feature Engineering: Encoding clinical variables, temporal data handling, and deriving risk scores. Predictive Modeling: Applying linear/logistic regression, decision trees, random forests, and gradient boosting for disease prediction and outcome forecasting. Machine Learning (ML) in Healthcare: Introduction to deep learning architectures (CNNs for imaging analysis, RNNs for temporal EHR data), model explainability. Evaluation Metrics: Sensitivity, specificity, PPV, NPV, AUROC, and their clinical significance.

Unit 3: Advanced Topics in Health Analytics

Advanced ML Techniques: Transfer learning, few-shot and zero-shot learning for rare diseases or small datasets. Natural Language Processing: Extracting insights from clinical notes, entity recognition (conditions, medications), summarization of patient records. Genomic data analysis. Federated Learning and Privacy-Preserving Analytics: Training models across distributed datasets while maintaining patient confidentiality.

Unit 4: Population Health Analytics and Imaging Analytics

Population Health Management: Identifying high-risk populations, preventive care strategies, reducing readmissions, and cost containment. Imaging Analytics: Radiomics, computer-aided diagnosis, pathology image interpretation, AI-driven diagnostics. AI-driven clinical decision support systems (CDSS).

Practical Component

1. Use data preprocessing on a synthetic EHR dataset, handle missing values, and perform basic statistical analysis.
2. Build and evaluate a logistic regression or random forest model to predict a healthcare-related metric.
3. Extract medical conditions or medications from clinical notes.
4. Integrate genomic data with clinical variables (if datasets are available).
5. Model Deployment and Interpretability - Create a simple dashboard for clinicians, integrate model explanations.

Suggested Readings

1. "Healthcare Data Analytics" by Chandan K. Reddy, Charu C. Aggarwal.
2. "Health Analytics: Gaining the Insights to Transform Health Care" by Jason Burke, Wiley.
3. "Bioinformatics and Functional Genomics" by Jonathan Pevsner.
4. "Healthcare Analytics for Quality and Performance Improvement" by Trevor L. Strome
5. Research articles and review papers published in leading journals on recent AI-driven DSS implementations and genomic data-driven studies.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Fundamentals of Time Series Analysis (DSE-4/GE-6)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Fundamentals of Time Series Analysis	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Outcomes

By the end of the course, students will be able to:

1. Identify and describe the structure and components of time series data.
2. Apply statistical and computational techniques for time series modeling.
3. Perform forecasting and evaluate the performance of time series models.
4. Solve domain-specific problems using time series analysis tools and techniques.

Course Objectives

1. To understand the fundamental principles and methods of time series analysis.
2. To equip students with the skills to analyze, model, and forecast time series data.
3. To introduce tools and libraries used for implementing time series models in Python.
4. To explore applications of time series analysis in various domains.

UNIT I

Introduction to Time Series Analysis: Definition and characteristics of time series data, Types of time series (e.g., stationary, non-stationary, seasonal), Components of time series: trend, seasonality, cycle, and noise, Exploratory Data Analysis (EDA) for time series, Data preparation and pre-processing for time series (e.g., scaling, transformations)

UNIT II

Statistical Methods for Time Series: Stationarity and differencing, Autoregressive models (AR), Moving Average models (MA), and ARMA models, ARIMA and SARIMA models for seasonal time series, Model selection criteria: AIC, BIC, Diagnostic checking: Residual analysis and model adequacy tests

UNIT III

Advanced Time Series Modeling: Exponential Smoothing methods (e.g., Simple, Holt's, Holt-Winters), State-space models and Kalman filters, Vector Autoregressive (VAR) and Vector Error Correction Models (VECM), Machine Learning for time series: Random Forests, Gradient Boosting, and LSTMs (Long Short-Term Memory networks), Evaluating forecasting accuracy: MSE, RMSE, MAPE

UNIT IV

Applications and Case Studies: Applications in finance (e.g., stock price prediction), retail (demand forecasting), and healthcare (disease trend analysis), Time series analysis in climate data and energy forecasting, Case study: Forecasting using Python libraries (statsmodels, Prophet, etc.), Challenges in real-world time series data (e.g., missing values, irregular intervals), Emerging trends: Anomaly detection in time series etc

Practical Component

1. Exploratory Analysis and Visualization (Time series decomposition using Python libraries (e.g., statsmodels, Plotting ACF and PACF)
2. Modeling and Forecasting (Implementing ARIMA/SARIMA models on real datasets, Experimenting with exponential smoothing techniques (e.g., Holt-Winters)
3. Machine Learning for Time Series (Developing LSTM models for time series forecasting, Comparing ML-based approaches with traditional models)
4. Real-World Applications (Case studies in finance, retail, and healthcare, Using Prophet and other frameworks for quick forecasting solutions)
5. Handling missing data and irregular intervals and Conducting anomaly detection experiments in time series data

Suggested Readings

1. "Introduction to Time Series and Forecasting" by Peter J. Brockwell and Richard A. Davis
2. "Time Series Analysis and Its Applications" by Robert H. Shumway and David S. Stoffer
3. "Practical Time Series Analysis" by Aileen Nielsen
4. Python libraries: statsmodels, Prophet, scikit-learn, and TensorFlow

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Agile Software Development (DSE-4/GE-6)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Agile Software Development	4	3L	0T	1P	-	-

Course Hours: L: 03 T: 00 P: 02

Course Objectives

1. To understand the core principles, values, and practices of Agile Software Development.
2. To explore Agile methodologies such as Scrum, Kanban, and Extreme Programming (XP).
3. To develop skills for Agile planning, estimation, and iterative development.
4. To emphasize the role of testing and quality assurance in Agile frameworks.

Course Outcomes

By the end of the course, students will be able to:

1. Explain the Agile philosophy and its relevance to modern software development.
2. Apply Agile methodologies to plan, execute, and manage software projects.
3. Demonstrate skills in iterative development, sprint planning, and backlog management.
4. Integrate continuous testing and delivery practices within Agile frameworks.
5. Utilize Agile tools for collaboration, tracking, and reporting.

Unit I

Introduction to Agile Development: Agile Manifesto and principles, Comparison of Agile with traditional software development approaches (e.g., Waterfall), Benefits and challenges of Agile development, Overview of Agile methodologies: Scrum, Kanban, XP, Lean Software Development, Agile team roles and responsibilities (e.g., Product Owner, Scrum Master, Team Member)

Unit II

Agile Practices and Frameworks: Scrum framework:Sprint planning, daily stand-ups, sprint reviews, and retrospectives, Product backlog and sprint backlog management, Kanban: Visualizing workflows and limiting work-in-progress (WIP), Extreme Programming (XP): Pair programming, test-driven development (TDD), refactoring, Agile estimation and planning: Story points, velocity, and release planning

Unit III

Agile Testing and Quality Assurance: Principles of Agile testing: Continuous testing and early defect detection, Test-driven development (TDD) and behavior-driven development (BDD), Exploratory testing in Agile, Automated testing in Agile: Tools and strategies, Role of CI/CD (Continuous Integration/Continuous Deployment) in Agile testing, Managing technical debt and maintaining code quality

Unit IV

Tools, Trends, and Case Studies: Agile tools for project management and collaboration: JIRA, Trello, Azure DevOps, Agile metrics and reporting: Burn-down charts, burn-up charts, velocity tracking, DevOps and its integration with Agile, Case studies of successful Agile implementations (e.g., Spotify Model), Recent trends: Scaled Agile Framework (SAFe), Agile in distributed teams, AI in Agile

Practical Component

1. Create a mind map or discussion board reflecting how the Agile Manifesto applies to real-world scenarios.
2. Write user stories for a sample project. Prioritize and create a product backlog.
3. Simulate a Sprint Planning session. Divide a product backlog into sprints, estimating tasks using story points or hours
4. Role-play a Scrum team. Conduct a mock Daily Scrum, Sprint Review, and Sprint Retrospective.
5. Use an open source tool or physical sticky notes to manage a Kanban board. Practice limiting work in progress (WIP).
6. Practice effort estimation techniques for sample tasks.
7. Collaboratively solve a simple coding problem using pair programming techniques.
8. Work on a small project across multiple sprints. Include sprint planning, execution, review, and retrospective.
9. Use Agile project management tools to manage a project.
10. Simulate stakeholder meetings to gather feedback on deliverables after a sprint.

Suggested Readings

1. *Agile Software Development: The Cooperative Game* by Alistair Cockburn
2. *Essential Scrum: A Practical Guide to the Most Popular Agile Process* by Kenneth S. Rubin
3. *Test-Driven Development: By Example* by Kent Beck

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Software Reliability (DSE-4/GE-6)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Software Reliability	4	3L	1T	0P	-	Software Engineering

Course Hours: L: 03 T: 01 P: 00

Course Objectives

1. To provide a thorough understanding of software reliability concepts and their significance.
2. To equip students with the ability to model, predict, and improve software reliability.
3. To apply Software Reliability Growth Models in Software Development
4. To emphasize the Application of Software Reliability Models
5. To introduce the use of tools and techniques for reliability analysis and testing.

Course Outcomes

1. Define and explain the principles and importance of software reliability in modern software systems.
2. Apply reliability metrics and models to evaluate and predict the performance of software systems..
3. Design strategies for improving software reliability through fault prevention, detection, and tolerance.
4. Utilize real-world data and case studies to develop a practical understanding of reliability challenges and solutions

Unit I: Fundamentals of Software Reliability

Definition and importance of software reliability, Software faults, errors, and failures, Reliability vs. availability, Software reliability in the software development lifecycle (SDLC), Factors affecting software reliability, Introduction to probabilistic approaches in reliability

Unit II:

Software Reliability Metrics & Models: Reliability metrics: Mean Time To Failure (MTTF), Mean Time Between Failures (MTBF), failure rate, Basic reliability models: Exponential and Weibull models, Musa, Jelinski-Moranda models, Fault injection and debugging concepts, Limitations of reliability models

Unit III:

Non Homogeneous Poisson Process Models: Musa models- Basic Execution time, Logarithmic Poisson Execution time models - Goel – Okumoto model, Yamada delayed S-shaped model, Imperfect debugging models

Reliability in Testing and Management: Testing Strategies for Reliability Assurance, Stress and Fault Tolerance Testing, Managing Reliability in Development and Deployment, Real-World Data Analysis: Using failure data to predict software performance. Tools: Overview of Software Failure and Reliability Assessment Tool (SFRAT)

Unit IV:

Recent Trends in Software Reliability: AI and machine learning in reliability prediction, Reliability in cloud-based and distributed systems, Reliability in IoT and embedded systems, DevOps practices for enhanced reliability, Emerging tools and platforms for reliability engineering.

Tutorial Component

1. Case Study Discussion: Analyze real-world examples of software failures (e.g., Therac-25, Ariane 5, or Toyota's recall due to software bugs), Identify the causes of faults, errors, and failures in the chosen case study. Discuss how reliability could have been improved.
2. SDLC and Reliability: Map reliability considerations to each phase of the SDLC. Create a checklist of reliability practices for design, development, and testing.
3. Fault and Failure Classification: Use sample software logs to classify errors as faults, failures, or bugs. Analyze the impact of these faults on system reliability.
4. Introduction to Probabilistic Approaches: Perform simple probabilistic calculations for failure rates using given data. Discuss the importance of probability in reliability prediction.

Suggested Readings

1. John D. Musa, Anthony Iannino, Kazuhira Okumoto, “Software Reliability – Measurement, Prediction, Application, Series in Software Engineering and Technology.
2. Michael Lyu, “Handbook of Software Reliability Engineering”, IEEE Computer Society Press
3. John D. Musa, “Software Reliability Engineering”, Tata McGraw Hill, 1999.
4. Patrick P. O'Connor, “Practical Reliability Engineering” , 4th Edition, John Wesley & sons, 2003.
5. Xie M, “Software Reliability Modelling”, World Scientific, Singapore, 1991.
6. Research papers published in relevant international journals

List of tutorial tasks

Note: The course instructor will design tasks to complete the tutorial component of the course.

Blockchain Applications (DSE-4/GE-6)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Blockchain Applications	4	3L	0T	1P	-	Blockchain Essentials

Course Hours: L: 03 T: 00 P: 02

Course Outcomes
At the end of this course, students will be able to:

1. Explain the foundational concepts of blockchain technology and evaluate its applications across diverse sectors such as finance, supply chain, and healthcare.
2. Critically assess the design, functionality, and risks associated with decentralized finance (DeFi) platforms, stablecoins, and digital currencies.
3. Explore and implement blockchain-based solutions for identity management, governance, and digital assets (NFTs), considering regulatory, ethical, and environmental impacts.
4. Identify emerging trends in blockchain technology.
5. Propose, design, and analyze use-case-driven blockchain solutions to address real-world problems, ensuring security, transparency, and scalability.

Course Objectives

1. Introduce students to the fundamental principles of blockchain technology, its underlying cryptographic mechanisms, and consensus models.
2. Familiarize students with practical blockchain applications in finance (DeFi, CBDCs), supply chain management, healthcare, and digital identity systems.
3. Equip students with an understanding of smart contracts, NFTs, and DAOs, enabling them to design secure and reliable decentralized applications.

Unit 1: Introduction to Blockchain Applications
Foundational blockchain concepts, Blockchain Applications, Introduction to Decentralized Finance (DeFi), Smart contracts and their use in lending, borrowing, and staking, Stablecoins and Central Bank Digital Currencies (CBDCs), Risks and challenges of DeFi (scalability, security, fraud), Cryptocurrency ecosystems: Wallets, exchanges, and trading platforms., Case studies: Bitcoin as a payment system, Ethereum's smart contract use cases., Case studies of successful implementations (e.g., Bitcoin, Ethereum, Hyperledger).

Unit 2: Blockchain in Supply Chain and Healthcare
Supply Chain Applications: Ensuring transparency and traceability in supply chains, Use cases: Counterfeit prevention, product provenance, Integration with IoT and AI, Case studies: IBM Food Trust, VeChain.
Healthcare Applications: Secure data sharing and patient data privacy, Blockchain in drug traceability and clinical trials, Case studies: MediBloc, Medicalchain.

Unit 3: Digital Identity, NFTs, and Governance
Identity Management: Decentralized Identity (DID) and self-sovereign identity, Blockchain for Know Your Customer (KYC) processes, Case studies: Sovrin Network, Microsoft ION.

Non-Fungible Tokens (NFTs): Token standards (ERC-721, ERC-1155), Use cases: Art, gaming, and digital collectibles, Legal, ethical, and environmental concerns.
Governance Applications: Decentralized Autonomous Organizations (DAOs), Blockchain-based voting systems and transparency in elections, Case studies: Aragon DAOs, blockchain voting experiments.

Unit 4: Emerging Trends
Emerging Trends in Blockchain:Layer 2 solutions: Lightning Network, Optimistic Rollups., Interoperability solutions: Polkadot, Cosmos, Blockchain and AI integration, Quantum computing and blockchain security.

Practical Component

1. Create and deploy a simple smart contract for a financial application (e.g., lending/borrowing on Ethereum).
2. Prototype a product tracking system using blockchain (e.g., Hyperledger or Ethereum).
3. Build a secure medical record-sharing app using blockchain.
4. Create a DID system using tools like uPort or Sovrin.
5. Mint and trade an NFT on Ethereum or Polygon.
6. Simulate DAO governance and build a blockchain-based voting system.
7. Choose an industry-specific problem and develop a blockchain-based solution, such as decentralized crowdfunding platform, blockchain certificate verification system, A supply chain traceability application etc

Suggested Readings

1. Blockchain Basics: A Non-Technical Introduction in 25 Steps by Daniel Drescher
2. "Blockchain Applications: A Hands-On Approach" by Arshdeep Bahga and Vijay Madisetti
3. DeFi and the Future of Finance by Campbell R. Harvey, Ashwin Ramachandran, and Joey Santoro.
4. Blockchain and the Supply Chain: Concepts, Strategies, and Practical Implementation by Nick Vyas.
5. Blockchain in Healthcare: Innovations that Empower Patients, Connect Professionals, and Improve Care by Malaya Zeal.
6. The NFT Handbook: How to Create, Sell and Buy Non-Fungible Tokens by Matt Fortnow and QuHarrison Terry.
7. Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications by Imran Bashir
8. Bitcoin and Cryptocurrency Technologies by Arvind Narayanan, Joseph Bonneau, et al

List of Experiments
Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Cybersecurity with Blockchain (DSE-4/GE-6)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Cybersecurity with Blockchain	4	3L	0T	1P	-	Fundamentals of Cybersecurity, Blockchain Essentials

Course Hours: L: 03 T: 00 P: 02

Course Outcomes
At the end of this course, students will be able to:

1. Understand the integration of blockchain principles with traditional cybersecurity frameworks and evaluate how decentralization, consensus mechanisms, and cryptographic constructs enhance security.
2. Design and implement decentralized identity and access control solutions, ensuring secure key management and robust verification of credentials.
3. Identify and mitigate vulnerabilities in smart contracts and blockchain applications, utilizing auditing, testing, and formal verification tools to ensure reliability and correctness.
4. Employ advanced techniques for data privacy and confidentiality within blockchain ecosystems, leveraging zero-knowledge proofs, secure multi-party computation, and other privacy-preserving methods.
5. Analyze the security implications of various tokenized assets (e.g., stablecoins, security tokens) and integrate privacy-enhancing technologies to develop secure, scalable, and trustworthy blockchain-based solutions.

Course Objectives

1. Provide an in-depth understanding of blockchain security fundamentals and their interplay with cybersecurity principles.
2. Equip students with skills for implementing secure decentralized identity frameworks and managing cryptographic keys in blockchain environments.
3. Enable students to identify, analyze, and remediate vulnerabilities in smart contracts, applying secure coding standards and formal verification techniques.
4. Introduce data privacy challenges in blockchain applications and familiarize students with cutting-edge cryptographic tools for maintaining confidentiality and integrity.

Unit 1: Foundations of Blockchain-Enhanced Security
Revisiting Cybersecurity Fundamentals in a Blockchain Context: CIA triad, threat landscape, and how blockchain properties reinforce or challenge these concepts. Blockchain Security Architecture: Decentralization, consensus mechanisms (PoW, PoS), tamper-evident ledgers, and how these attributes counter traditional cyber threats. Threat Modeling and Risk Assessment in Blockchain Networks: Identifying attack surfaces (51% attacks, Sybil attacks), securing communication channels, and secure key management.

Unit 2: Identity and Access Control with Blockchain
Decentralized Identity Management (DID): Self-sovereign identity, verifiable credentials, DID frameworks (e.g., Sovrin), and blockchain authentication. Access Control Mechanisms: Permissioned vs. permissionless blockchains,

role-based and attribute-based access control models implemented on DLTs. Key Management and Security: Private key handling, hardware security modules (HSM), multi-signature schemes, and hierarchical deterministic wallets.

Unit 3: Secure Smart Contracts and Blockchain Applications
Smart Contract Vulnerabilities: Common exploits (reentrancy, arithmetic overflows, access control flaws), vulnerability scanning tools, and secure coding best practices. Security Auditing and Formal Verification: Automated auditing tools (Mythril, Slither) and formal verification methods (Solidity, Vyper etc) to ensure correctness.

Unit 4: Advanced Techniques, Data Privacy
Data Privacy in Blockchain Applications: Techniques for handling sensitive data on-chain, zero-knowledge proofs (ZKPs), state channels, and confidential transactions. Stablecoins, Security Tokens, and Digital Assets: Analyzing token standards and their security implications. Privacy-Preserving Frameworks: Homomorphic encryption, secure multi-party computation (MPC), and differential privacy methods integrated with blockchain.

Practical Component
1. Secure Key Management and DID Setup
2. Smart Contract Vulnerability Analysis
3. Use automated vulnerability scanning tools (Mythril, Slither) to identify potential issues and apply remediations.
4. Formal Verification of Smart Contracts:
5. Experiment with zero-knowledge proof frameworks to implement a confidential transaction.
6. Integrate a simple state channel or sidechain to enhance privacy and scalability.
7. Perform a threat model and risk assessment for a hypothetical blockchain application scenario. Identify attack surfaces and propose mitigation strategies, documenting the security design.

Suggested Readings

1. "Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications" by Lorne Lantz and Daniel Cawrey
2. "Ultimate Blockchain Security Handbook: Advanced Cybersecurity Techniques and Strategies for Risk Management, Threat Modeling, Pen Testing, and Smart Contract Defense for Blockchain" by Taha Sajid
3. "Hands-On Blockchain for Python Developers" by Arjuna Sky Kok, for practical coding approaches
4. Research papers from reputed journals and conferences on blockchain security, smart contract verification, and privacy-preserving techniques
5. Online resources and documentation for decentralized identity frameworks (e.g., Sovrin, W3C DID), auditing tools (MythX, Slither), and ZKP libraries

List of Experiments
Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Advanced Game Development (DSE-4/GE-6)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Advanced Game Development	4	3L	0T	1P	-	Game Development

Course Hours: L: 03 T: 00 P: 02

Course Outcomes

At the end of this course, students will be able to:

1. Implement advanced rendering techniques, including PBR, ray tracing, and custom shader pipelines, to achieve high-fidelity visual experiences.
2. Integrate complex physics simulations into games and apply performance profiling and optimization techniques to ensure smooth, responsive gameplay.
3. Architect scalable multiplayer systems, employ secure networking protocols, and utilize CI/CD pipelines and data analytics tools to streamline production and enhance player retention.
4. Develop immersive XR applications, adapt games for multiple platforms and cloud infrastructures, and ensure high-quality user experiences across diverse devices and environments.

Course Objectives

1. Equip students with the knowledge to create visually sophisticated games using modern rendering technologies and custom shader development.
2. Enable students to design and implement realistic physics simulations while maintaining optimal performance through profiling and optimization strategies.
3. Provide students with a comprehensive understanding of multiplayer architectures, secure network communication, automated workflows, and data-driven decision-making in game production.
4. Familiarize students with advanced XR development techniques, cross-platform portability considerations.

Unit 1: Advanced Graphics Techniques

High-Fidelity Rendering Techniques: Physically Based Rendering (PBR), global illumination, ray tracing, and advanced post-processing effects. Shader Programming and Custom Pipelines: Writing custom shaders, and integrating special effects.

Unit 2: Physics Simulations and Performance Assessment

Complex Physics Simulations: Soft-body physics, realistic fluid and cloth simulations, advanced collision algorithms, advanced simulations. Performance Profiling and Optimization: GPU/CPU profiling, LOD strategies, batching, culling, and memory optimization.

Unit 3: Multiplayer, Networking, and Production Pipelines

Multiplayer Game Architectures: Client-server models, peer-to-peer, dedicated servers, load balancing, matchmaking, and latency compensation. Network Protocols and Security: TCP/UDP trade-offs, prediction and interpolation, cheat detection, secure data transmission. Continuous Integration and Deployment (CI/CD): Automated builds, testing pipelines, asset management, version control best practices for large teams. Data Analytics and Telemetry: Collecting in-game analytics, using A/B testing, telemetry-driven updates, player retention analysis.

Unit 4: XR Development and Cross-platform portability

Extended Reality (XR) Development: Advanced VR/AR techniques, haptic feedback, spatial audio, user comfort, and input methods. Cross-Platform and Cloud Gaming: Porting to multiple platforms (consoles, PC, mobile), leveraging cloud gaming infrastructures.

Practical Component

1. Implement ray tracing or advanced post-processing in a chosen game engine, profile and optimize for performance.
2. Build a small-scale multiplayer prototype, handle network synchronization issues, and add basic matchmaking.
3. Integrate a continuous integration pipeline with automated builds and tests, experiment with asset versioning and rollbacks.
4. Create a simple VR or AR experience emphasizing interaction fidelity, comfort, and immersion.
5. Add telemetry events, analyze player data logs,

Suggested Readings

1. "Game Engine Architecture" by Jason Gregory
2. "Real-Time Rendering" by Tomas Akenine-Moller et al.
3. Latest GDC (Game Developers Conference) and SIGGRAPH proceedings, relevant research papers, and industry white papers on advanced topics in rendering, AI, and XR.
4. Documentation of industry-standard physics engines.

List of Experiments

Note: The course instructor will design experiments/mini-projects to complete the practical component of the course.

Augmented and Virtual Reality Systems Design (DSE-4/GE-6)

CREDIT DISTRIBUTION, ELIGIBILITY AND PRE-REQUISITES OF THE COURSE

Course title	Credits	Credit distribution of the course			Eligibility criteria	Prerequisite of the course (if any)
		Lecture	Tutorial	Practical		
Augmented and Virtual Reality Systems Design	4	3L	0T	1P	-	Foundations of Augmented and Virtual Reality

Course Hours: L: 03 T: 00 P: 02

Course Outcomes
At the end of this course, students will be able to:
1. Understand the basic principles and technologies behind augmented and virtual reality systems.
2. Design and implement interactive AR/VR applications using modern hardware and software tools.
3. Apply knowledge of 3D modeling, rendering, and computer vision to create immersive virtual environments.
4. Develop AR applications that integrate the physical world with digital content through sensors and mobile devices.
5. Analyze the use of AR/VR in various industries, including gaming, education, healthcare, and retail.
6. Understand the challenges and limitations of AR/VR technologies and develop strategies to overcome them.
7. Evaluate the effectiveness of AR/VR in enhancing user experience and engagement.

Course Objectives
1. Provide students with a foundational understanding of augmented and virtual reality technologies and their applications.
2. Equip students with the skills to develop AR/VR applications using appropriate hardware platforms and software frameworks.
3. Introduce students to advanced topics such as 3D rendering, gesture recognition, and immersive experiences.
4. Help students design and create interactive AR experiences using real-world sensors and mobile applications.
5. Enable students to analyze the diverse applications of AR/VR in various fields and industries.
6. Discuss the ethical and social implications of AR/VR technologies and their influence on user behavior.
7. Prepare students to overcome technical challenges in AR/VR systems and deliver high-quality user experiences.

Unit 1: Introduction to AR and VR Systems
Overview of Augmented Reality (AR) and Virtual Reality (VR); Key Differences and Applications; Hardware and Software Components: Head-mounted Displays, Sensors, and Input Devices; Basics of 3D Coordinate Systems and Transformations; Introduction to AR/VR Development Tools: Unity, Unreal Engine, and ARKit/ARCore.

Unit 2: AR/VR Interaction and User Experience Design

Design Principles for Immersive Interfaces; Gesture Recognition, Eye-tracking, and Motion Capture; Techniques for Creating Intuitive AR/VR Experiences; Haptic Feedback Systems; Evaluation of User Experience (UX) in AR/VR Environments; Case Studies of Successful AR/VR Applications.

Unit 3: AR/VR Development Techniques

3D Modeling and Scene Design for AR/VR; Integration of Real and Virtual Content: Rendering and Registration; AR/VR Content Development Using Unity/Unreal Engine; Marker-based and Markerless AR Systems; Optimizing Performance for Real-time Interaction in AR/VR.

Unit 4: Advanced Topics and Applications in AR/VR

Mixed Reality and the Future of Immersive Technologies; Use of AI in AR/VR: Scene Understanding and Object Recognition; Applications in Healthcare, Education, and Entertainment; Challenges and Ethical Considerations in AR/VR Systems: Privacy, Security, and Accessibility; Emerging Trends: Spatial Computing and Metaverse Development.

Practical Component

1. Creating and deploying basic AR applications using mobile devices and AR frameworks like ARCore or ARKit.
2. Developing a virtual reality environment using Unity and Oculus Rift or HTC Vive.
3. Designing an interactive 3D virtual world and implementing object manipulation in VR.
4. Building an AR application that uses real-time tracking to place virtual objects in the physical world.
5. Implementing gesture recognition for interacting with AR/VR environments.
6. Conducting usability testing of AR/VR applications and optimizing them for improved user experiences.
7. Evaluating the impact of VR and AR applications in education, healthcare, or entertainment through small prototype projects.

Suggested Readings

1. Augmented Reality: Principles and Practice by Dieter Schmalstieg and Tobias Hollerer (Pearson)
2. Virtual Reality by Steven M. LaValle (Cambridge University Press)
3. Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile by Tony Parisi (O'Reilly Media)
4. Augmented Reality: Where We Will All Live by Jon Peddie (Springer)
5. Programming 3D Applications with HTML5 and WebGL by Tony Parisi (O'Reilly Media)
6. 3D User Interfaces: Theory and Practice by Doug Bowman, Ernst Kruijff, Joseph LaViola, and Ivan Poupyrev (Addison-Wesley)
7. Unity Virtual Reality Projects by Jonathan Linowes (Packt Publishing)

Annexure-3

Faculty of Technology
University of Delhi

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

(Course Structure and Curriculum of B.Tech. (ECE) Third Year)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Course Structure and Curriculum of B.Tech. (ECE) Third Year

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Course Structure of B. Tech (ECE) Third Year
Third Year

Semester – V

S. No.	Course Domain	Course Title	Credits*			Total Credits
			L	T	P	
1.	DSC-13	Control Systems Engineering	3	0	1	4
2.	DSC-14	Digital Signal Processing	3	0	1	4
3.	DSC-15	Analog Communication Systems	3	0	1	4
4.	DSE-3	Select a course from the specified list of DSE - 3				4
5.	GE-5	Select a course from the specified list of GE - 5				4
6.	SEC or IAPC	Choose one SEC or Internship/Apprenticeship/Project/Community Outreach (IAPC)				2
Total Credits						22

Semester – VI

S. No.	Course Domain	Course Title	Credits*			Total Credits
			L	T	P	
1.	DSC-16	Hands – On CMOS VLSI Design	3	0	1	4
2.	DSC-17	Embedded Systems and Applications	3	0	1	4
3.	DSC-18	Digital Communication Systems	3	0	1	4
4.	DSE-4	Select a course from the specified list of DSE - 4				4
5.	GE-6	Select a course from the specified list of GE - 6				4
6.	SEC or IAPC	Choose one SEC or Internship/Apprenticeship/Project/Community Outreach (IAPC)				2
Total Credits						22

**Credits*
L (01 Credit) is equivalent to 01 contact hour per week
T (01 Credit) is equivalent to 01 contact hour per week
P (01 Credit) is equivalent to 02 contact hours per week

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Pool of DSEs offered by the Department

S. No.	Semester	DSE	Course Title
1.	V	DSE - 3	Spread Spectrum Communication
2.			Network Technologies and Interfacing
3.			Digital Image Processing
4.	VI	DSE - 4	Wireless Sensor Networks
5.			Artificial Intelligence in Electronics
6.			Coding and Data Compression Techniques
7.			Electromagnetic Compatibility: Principles and Applications

List of SECs offered by the Department

S. No.	Semester	Course Title
1.	V	IAPC
2.	VI	Radar and Antenna Workshop

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Specialization and Minors offered by the Department

Semester	DSE/ GE	ECE Minor (Open only for CSE/ EE)	Specializations for ECE/ Minors for EE and CSE			
			Telecommunication Networks	VLSI Technology and System Design	IoT System Design	Computer Vision
V	DSE-3/ GE-5	Communication Architecture	Wireless Communication and Mobile Networks	Advanced Digital VLSI Circuits and Physical Design	Introduction to Security of Cyber-Physical Systems	Intelligent Imaging
VI	DSE-4/ GE-6	Antenna and Wave Propagation	Satellite Communication	CMOS Analog IC Design	Ubiquitous Sensing, Computing and Communication	Deep Learning for Image Analysis
		Elements of Wireless Communication	Cognitive Radio & Networks	Current Mode Analog VLSI Circuits	Introduction to Embedded Systems for IoT	Deep & Reinforcement Learning

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Discipline Specific Core (DSC) Courses for B. Tech. (ECE) – Semester V

Control Systems Engineering (DSC - 13)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Control Systems Engineering	4	3	0	1	Signals and Systems

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Understand core concepts of linear, nonlinear, time-invariant, and time-varying control systems.
- Develop skills in modeling physical systems using differential equations, transfer functions, and block diagrams.
- Analyze system behavior in both time and frequency domains.
- Apply stability criteria and root-locus techniques for system analysis.
- Design and tune controllers (P, PI, PID) and compensators (lag, lead, lead-lag).
- Utilize state-space methods to analyze controllability and observability.

Course Outcomes:
After completing the course, the students should be able to:
1. Classify and model various control systems.
2. Evaluate time response and steady-state errors.
3. Assess stability using Routh-Hurwitz and root-locus methods.
4. Interpret frequency response plots (Bode, Nyquist) for stability and performance.
5. Design appropriate controllers and compensators to meet specifications.
6. Apply state-space techniques to analyze and design advanced control systems.

Unit - I
Introduction to Control System: Linear, Non Linear, Time Varying and Linear Time Invariant System, Mathematical Modelling of Electrical and Mechanical Systems, Differential Equations of Physical Systems, Transfer Functions, Block Diagram and Signal Flow Graphs. Feedback and Non feedback Systems. Reduction of Parameter Variations by use of Feedback Control Over System Dynamics. Feedback Control of Effects of Disturbance.
Control Systems and Components DC and AC Servomotors, Synchro Error Detector, Tacho Generator and, Stepper Motors etc.

Unit - II
Time Response Analysis: Standard Test Signals, Time Response of First-order Systems, Time Response of Second-Order Systems, Steady-State Error and Error Constants, Effect of Adding a Pole/ Zero to a System, Design Specifications of Second-Order Systems and Performance Indices. The Concept of Stability, Necessary Conditions for Stability, Hurwitz Stability Criterion, Routh Stability Criterion and relative Stability Analysis. The Root Locus Concept, Construction of Root Loci, Root Contours, Systems with Transportation Lag, Sensitivity of the Roots of the Characteristic equation.

Unit - III

Frequency Response Analysis: Correlation Between Time and Frequency Response, Bode Plots, Polar Plots. Stability in Frequency Domain: Mathematical Preliminaries, Nyquist Stability Criterion, Calculation of Gain Margin and Phase Margin, Assessment of Relative Stability Using Nyquist Criterion and Closed-Loop Frequency Response.

Unit - IV

P, PI and PID Control Action and Their Effect, Compensator and Controller Design: Design of Lag, Lead, Lead Lag, Feedback compensator, Preliminary Considerations of Classical Design, Realization of Basic Compensators, Cascade Compensation in Time Domain Cascade Compensation in Frequency Domain, Tuning of PID Controllers.

Control Systems Analysis in State Space: State-Space Representations of Transfer-Function Systems, Solving the Time-Invariant State Equation, Controllability, Observability.

Suggested Readings

1. Control Systems Engineering by Norman S. Nise (Wiley)
2. Modern Control Engineering by Katsuhiko Ogata (Pearson)
3. Modern Control Systems by Richard C. Dorf and Robert H. Bishop (Pearson)
4. Feedback Control of Dynamic Systems by Gene Franklin, J. Da Powell, and Abbas Emami-Naeini (Pearson)
5. Control System Engineering by I J Nagrath (New Age Publishers)
6. Automatic Control Systems by B. C. Kuo (Wiley)

List of Experiments (Hardware and Software using Matlab/Simulink)

1. Transient and Steady-State Analysis: analyze responses of first- and second- order systems using MATLAB.
2. PID Controller Design: Implement and tune PID controllers.
3. Utilize tuned PID controller for DC motors using MATLAB.
4. PID Controller Design: Implement PID control for a DC-DC converter using Arduino.
5. State-Space Analysis and Design: Perform state feedback control design and implementation using Raspberry Pi.
6. Analysis of a SISO and MIMO system
7. Frequency Response Analysis: Generate and interpret Bode, Nyquist, and Nichols plots using MATLAB.
8. Frequency Response Analysis using Bode Plot of a given transfer function using Hardware.
9. Design of Lag Compensator and Lead Compensator using MATLAB and Hardware.
10. Design of Lag and Lead Compensator using MATLAB and Hardware.

(Note: Course instructor may add/update new experiments in addition to the above suggested practical exercises.)

Digital Signal Processing (DSC - 14)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Digital Signal Processing	4	3	0	1	Signals and Systems

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- To understand the fundamentals of digital signal processing and its applications in real-world systems.
- To learn techniques for analyzing and designing digital filters and systems.
- To gain hands-on experience with modern tools and platforms for implementing DSP algorithms.
- To explore advanced topics such as adaptive filtering, multi-rate signal processing.
- To equip students with skills to solve industry-relevant problems using DSP techniques.

Course Outcomes:

After completing the course, the students should be able to:

1. Analyze and process discrete-time signals.
2. Design and implement digital filters for various applications.
3. Apply multi-rate signal processing techniques in communication systems.
4. Work with adaptive signal processing algorithms for noise cancellation and predictive analytics.

Unit - I

Fundamentals of DSP: Introduction to DSP and its applications, Discrete-Time Signals and Systems: Sampling, quantization, aliasing, Discrete-Time Fourier Transform (DTFT) and its properties, Linear Time-Invariant Systems: Convolution and stability analysis. Wavelet Transform: Continuous wavelets, discrete wavelets and its advantages.

Unit - II

Digital Filter Design (for both FIR and IIR): Design of Linear Phase FIR Filters: Window methods, frequency sampling methods, IIR Filters: Butterworth, Chebyshev, and elliptic filters, Filter realization: Direct form, cascade form, and parallel form, Applications in image compression.

Unit - III

Advanced DSP Techniques: Fast Fourier Transform (FFT) and its applications, Multi-rate Signal Processing: Decimation, interpolation, and polyphase structures, Adaptive Signal Processing: Adaptive Filtering – Kalman Filters, Wiener Filters, LMS and RLS algorithms.

Unit - IV

Real-Time DSP Applications: Real-Time Processing: Signal reconstruction, anti-aliasing filters and its applications for signal processing. Introduction to Digital Signal Processors: Fixed Point and Floating-Point processors, architectures. TMS 320C54XX and TMS320C67XX Architecture, Memory, Addressing Modes.

Suggested Readings

1. Digital Signal Processing by John G. Proakis and Dimitris K. Manolakis (Pearson)
2. Discrete-Time Signal Processing by Alan V. Oppenheim and Ronald W. Schafer (Pearson)
3. Digital Signal Processing: Principles, Algorithms, and Applications by John G. Proakis (Pearson)
4. DSP First by James H. McClellan, Ronald W. Schafer, and Mark A. Yoder (Pearson)
5. Digital Signal Processing: A Computer-Based Approach by Sanjit K. Mitra (Tata McGraw Hill)
6. Adaptive Filter Theory by Simon Haykin (Pearson)

List of Experiments

1. Sampling and Reconstruction of Signals
   Requirements - MATLAB/Python, DAC-ADC Converter, Oscilloscope.
   Objective - Understand the effects of under-sampling and aliasing.
2. Design a digital Differentiator using a Hamming Window
3. Design a digital Hilbert Transformer using a Hamming Window
4. Wavelet Transform for Signal Denoising
   Requirements - MATLAB Wavelet Toolbox, noisy biomedical signals.
   Objective - Analyze the denoising capability of wavelet transforms.
5. Implementation of FIR Filters
   Requirements - MATLAB, Python (Scipy), Filter Design Toolbox.
   Objective - Design filters and analyze their performance in suppressing noise.
6. Implementation of IIR Filters
   Requirements - MATLAB, Python (Scipy), Filter Design Toolbox.
   Objective - Design filters and analyze their performance in suppressing noise.
7. Adaptive Noise Cancellation
   Requirements - DSP Kit (e.g., TMS320C6713), MATLAB, microphone, and speakers.
   Objective - Implement LMS algorithm for noise removal in audio.
8. Adaptive Filtering Using RLS Algorithm
   Requirements - DSP Kit (e.g., TMS320C6713), MATLAB, noisy communication signals.
   Objective - Implement RLS for dynamic noise cancellation.
9. Multi-rate Signal Processing
   Requirements - MATLAB, Python, signal generator, decimation/interpolation circuits.
   Objective - Explore decimation and interpolation in multi-rate systems.
10. Filter Implementation using Fixed- and Floating-Point processors.

(Note: Course instructor may add/update new experiments in addition to the above suggested practical exercises.)

Analog Communication Systems (DSC - 15)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Analog Communication Systems	4	3	0	1	Electronic Devices and Circuits, Signals and Systems, Network Analysis and Synthesis

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- Introduce the core principles and analysis techniques for analog communication systems.
- Develop the skills to design, implement, and test analog modulation, transmission, and reception circuits.
- Expose students to emerging analog communication technologies, including SDR (Software-Defined Radio), cognitive radio, quantum communication influences, IoT integration, and cutting-edge modulation techniques.
- Enhance problem-solving and critical-thinking abilities through extensive hands-on lab work, projects, and case studies.

Course Outcomes:

After completing the course, the students should be able to:

1. Understand and analyze fundamental and advanced analog modulation techniques.
2. Design and simulate analog communication circuits using modern EDA tools and SDR platforms.
3. Integrate analog front-ends with IoT and edge computing systems, addressing real-world constraints.
4. Explore and critically assess emerging research fields such as cognitive radio, quantum-influenced communication, and OAM-based modulation.
5. Implement complex analog experiments, interpret results, and propose improvements, preparing for contemporary industry requirements.

Unit - I

Fundamentals of Analog Communication: Basics of Communication Systems: Representation of signals, frequency spectrum, and noise considerations, Amplitude Modulation (AM): DSB, SSB, VSB – generation, demodulation, bandwidth, power considerations, Angle Modulation: Frequency Modulation (FM) and Phase Modulation (PM) – principles, Carson's rule, generation, and detection, Signal-to-Noise Ratio (SNR), Noise Figure, and Noise in Communication Channels.

Unit - II

Advanced Analog Modulation & System Architecture: Pulse Modulation: PAM, PWM, PPM – characteristics, generation, and detection, Superheterodyne Receiver Architecture: Mixers, Local Oscillators, IF stages, Software Defined Radio (SDR) Basics: Architecture, reconfigurable front-ends, FPGA-based implementations, IoT Integration: Analog front-ends for low-power IoT devices, sensor signal conditioning, Introduction to Advanced RF Components: Filters, low-noise amplifiers (LNAs), impedance matching.

Unit - III
Transmission, Reception & System Integration: Transmission Lines and Wave Propagation: Parameters, reflections, impedance matching techniques, Multiplexing Techniques: Frequency Division Multiplexing (FDM), Time Division Multiplexing (TDM) in analog systems, Antennas in Analog Communications: Basic antenna theory, analog link budgeting, Analog Communication Over Fiber: Basic principles, intensity modulation, optical detectors, Introduction to Advanced RF Design: oscillator, mixer, and amplifier design.

Unit - IV
Emerging Trends and Research Directions: Cognitive Radio Systems: Spectrum sensing, dynamic spectrum allocation, energy-efficient analog signal processing, Satellite and CubeSat Analog Systems: Analog link design, small-satellite telemetry, and OAM (Orbital Angular Momentum) modulation for high capacity, Noise-Shaping and Advanced Filtering: Techniques to improve SNR and signal integrity.

Suggested Readings

1. Principles of Communication Systems by Herbert Taub and Donald Schilling
2. Communication Systems by Simon Haykin
3. Modern Digital and Analog Communication Systems by B.P. Lathi and Zhi Ding
4. RF Microelectronics by Behzad Razavi
5. Cognitive Radio Communications and Networks by A. Wyglinski, M. Nekovee

List of Experiments (Hardware on Breadboard / Software using NI Multisim)

1. Amplitude Modulation and Demodulation
   Objective: Implement and analyze standard AM.
2. Frequency Modulation using Varactor Diodes
   Objective: Generate and observe FM signals.
3. Generation and Analysis of Pulse Amplitude Modulation (PAM)
   Objective: To generate and study the characteristics of Pulse Amplitude Modulation (PAM) and understand its applications in communication systems.
   Requirements: Hardware
4. Generation and Analysis of Pulse Width Modulation (PWM)
   Objective: To generate and analyze Pulse Width Modulation (PWM) signals and observe how the pulse width varies with the amplitude of the modulating signal.
   Requirements: Hardware
5. Generation and Analysis of Pulse Position Modulation (PPM)
   Objective: To generate and analyze Pulse Position Modulation (PPM) signals and understand the impact of modulating signal amplitude on pulse position.
   Requirements: Hardware
6. Demodulation of PAM, PWM, and PPM Signals
   Objective: To demodulate the PAM, PWM, and PPM signals and recover the original modulating signal using appropriate techniques.
   Requirement: Hardware
7. Superheterodyne Receiver Simulation
   Objective: Design and simulate superheterodyne receiver stages.
   Requirements: MATLAB/Simulink.
8. SDR-Based AM/FM Modulation
   Objective: Implement modulation schemes using SDR (GNU Radio + USRP).
   Requirements: SDR hardware (e.g., USRP), GNU Radio.

9. Analog Signal over Fiber
   Objective: Transmit and receive analog signals over fiber optics.
   Requirements: Fiber optic trainer kit, Laser diode, Photodiode.
10. RF Amplifier Design and Analysis
    Objective: Design an RF amplifier and analyze gain, noise figure.
    Requirements: Keysight ADS, RF components.
11. QAM under Noise Conditions
    Objective: Generate QAM signals and analyze their performance under noise.
    Requirements: Signal Generator, GNU Radio, Spectrum Analyzer.
12. Noise Cancellation with Analog Filters
    Objective: Implement active filters to enhance SNR.
13. Orbital Angular Momentum (OAM) Modulation Setup
    Objective: Generate and analyze OAM-modulated analog signals.
    Requirements: OAM modulation kit, Signal Generator, MATLAB.
14. Satellite Analog Link Simulation (CubeSat scenario)
    Objective: Simulate an analog downlink for a CubeSat telemetry system.
    Requirements: CubeSat communication module simulator, MATLAB.
15. Noise-Shaping Filter Design
    Objective: Implement filters to shape noise spectrum and improve SNR.
    Requirements: Filter Design Toolkit, Multisim.

(Note: Course instructor may add/update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Discipline Specific Elective (DSE) Courses for B. Tech. (ECE) – Semester V

Spread Spectrum Communication (DSE – 3)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Spread Spectrum Communication	4	3	0	1	Signals and Systems, Mathematics - II

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To understand the fundamental concepts of Spread Spectrum (SS) communication systems.
- To design and analyze Spread Spectrum techniques and their applications in modern communication systems.
- To develop practical skills in implementing SS systems using hardware and software tools.
- To explore advanced topics such as Code Division Multiple Access (CDMA), frequency hopping, and their role in 5G and satellite communication systems.

Course Outcomes:
After completing the course, the students should be able to:
1. Explain the principles of Spread Spectrum communication and its advantages over conventional methods.
2. Design and simulate basic SS communication systems using MATLAB/Simulink or Python.
3. Implement hardware-based SS systems using tools like SDR, FPGA, and microcontrollers.
4. Analyze the role of Spread Spectrum techniques in CDMA, GPS, 5G, and secure communication.
5. Integrate advanced Spread Spectrum concepts into real-world communication challenges.

Unit - I
Fundamentals of Spread Spectrum Communication: Introduction to Spread Spectrum Systems, Definition, history, and applications, Benefits: interference rejection, multipath fading reduction, and security, Types of Spread Spectrum Techniques, Direct Sequence Spread Spectrum (DSSS), Frequency Hopping Spread Spectrum (FHSS), Spreading Codes, Pseudorandom noise (PN) sequences: properties and generation, Gold codes, Walsh codes, and Kasami sequences.

Unit - II
System Design and Performance Analysis: DSSS System Design, Transmitter and receiver architecture, Spreading and despreading processes, Processing gain and Bit Error Rate (BER) analysis, FHSS System Design, Hopping patterns and synchronization, BER analysis and jamming resistance, Multipath and Interference Effects, Rake receivers and diversity combining techniques, Interference rejection capabilities.

Unit - III

Applications of Spread Spectrum: Code Division Multiple Access (CDMA), Principles, Walsh codes, and orthogonality, Forward and reverse link analysis, Global Positioning System (GPS), Spread Spectrum in GPS signal structure, Synchronization and tracking, Secure Communications, Anti-jamming and eavesdropping resilience, Military and commercial applications.

Unit - IV

Advanced Topics and Emerging Trends: Spread Spectrum in 5G and Beyond, Role in massive MIMO and millimeter-wave communication, Integration with OFDM and NOMA, Cognitive Radio and Spectrum Sharing, Spread Spectrum for dynamic spectrum allocation, Hardware Implementation, SDR-based SS communication systems, FPGA and microcontroller-based DSSS/FHSS systems.

Suggested Readings

1. Digital Communication by John G. Proakis and Masoud Salehi.
2. Wireless Communications: Principles and Practice by Theodore S. Rappaport.
3. Spread Spectrum Communications Handbook" by Marvin K. Simon, Jim K. Omura, Robert A. Scholtz.
4. Principles of Communication Systems by Herbert Taub and Donald Schilling.
5. CDMA: Principles of Spread Spectrum Communication by Andrew J. Viterbi

List of Experiments (Software using NI Multisim/Matlab/Simulink)

1. Generation and Analysis of PN Sequences.
2. Simulation of DSSS System.
3. Frequency Hopping Spread Spectrum Simulation.
4. BER Analysis of DSSS under Noise
5. Implementation of DSSS Transmitter and Receiver on SDR (Tools: GNU Radio, USRP SDR)
6. Design of a Simple CDMA System.
7. Implementation of a Rake Receiver.
8. Spread Spectrum in GPS Signal Simulation
9. Study and implementation of FHSS.
10. SDR-based Real-Time Spread Spectrum Communication (Tools: USRP, GNU Radio)

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Network Technologies and Interfacing (DSE – 3)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Network Technologies and Interfacing	4	3	0	1	Fundamentals of Computer Programming

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To understand the principles of computer networks and interfacing systems.
- To learn about modern communication protocols, IoT technologies, and network design principles.
- To gain hands-on experience with hardware-software interfacing and network performance analysis.
- To explore advanced topics such as 5G technologies, network security, and AI-driven network optimization.

Course Outcomes:
After completing the course, the students should be able to:
1. Design and implement basic and advanced computer networks using industry-standard tools.
2. Interface hardware and software components to create efficient communication systems.
3. Analyze and optimize network performance for IoT and 5G applications.
4. Develop solutions for network security and manage modern interfacing challenges using AI/ML tools.

Unit - I
Fundamentals of Network Technologies: Network architectures: OSI and TCP/IP models, Communication protocols: IP, UDP, TCP, HTTP, FTP, Overview of IPv4 and IPv6, Switching techniques: Circuit switching, packet switching, and virtual circuits, Wireless and wired network design, Introduction to Network Simulation Tools (NS2, NS3, Wireshark). Network security: Firewalls, VPNs, and intrusion detection systems.

Unit - II
Interfacing Systems and IoT: Basics of hardware-software interfacing, Embedded systems for networked communication, IoT protocols: MQTT, CoAP, Zigbee, LoRaWAN, Interfacing microcontrollers (e.g., Arduino, STM32) with sensors and actuators, Networked IoT systems using Raspberry Pi, Use of cloud platforms (AWS IoT, Azure IoT) for real-time monitoring.

Unit - III
Advanced Network Technologies: Fundamentals of 5G: Architecture, protocols, and applications, SDN (Software-Defined Networking) and NFV (Network Function Virtualization), Performance evaluation using traffic models and QoS metrics, Network Optimization.

Unit - IV
Emerging Technologies and Applications: Vehicular networks and VANETs, Underwater and satellite communication networks, Blockchain in network security, Industrial IoT (IIoT) applications in smart factories, Network automation tools (e.g., Ansible, Puppet), AI and ML for network optimization.

Suggested Readings

1. Computer Networking: A Top-Down Approach by Kurose and Ross (Pearson).
2. Computer Networks by A. S. Tanenbaum (Pearson).
3. Computer Networks: A Top – Down Approach by Forouzan (McGraw Hill).
4. Data and Computer Communications by William Stallings (Pearson)
5. Mastering Networks: An Internet Lab Manual by Jorg Liebeherr and Magda El Zarki.
6. 5G Mobile and Wireless Communications Technology by Afif Osseiran et al.
7. Hands-On Networking Fundamentals by Michael Palmer.
8. Python for Data Analysis by Wes McKinney (for AI/ML in networks).
9. Introduction to Network Simulator NS2 by Teerawat Issariyakul and Ekram Hossain

List of Experiments (Based on Hardware/ Simulation)

1. Packet Sniffing and Analysis using Wireshark: Analyze network traffic and identify protocols.
2. Network Topology Design: Simulate wired and wireless networks using NS2/NS3.
3. IoT Device Communication: Establish an MQTT-based communication system using Raspberry Pi.
4. IPv6 Configuration: Design and configure an IPv6 network for efficient communication.
5. Firewall and VPN Setup: Implement a basic firewall and VPN for secure communication.
6. QoS Analysis: Analyze network performance using QoS metrics and traffic models.
7. 5G Network Simulation: Simulate basic 5G network architecture using MATLAB or NS3.
8. Sensor Interfacing with Microcontrollers: Interface sensors and collect data using Arduino/STM32.
9. IoT Cloud Integration: Connect Raspberry Pi to AWS IoT and monitor real-time data.
10. AI-based Network Optimization: Use Python to develop basic ML models for network performance analysis.
11. Network Automation: Automate network configurations using Ansible.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Digital Image Processing (DSE – 3)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Digital Image Processing	4	3	0	1	Fundamentals of Computer Programming, Mathematics – I

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Introduce fundamental concepts and techniques of digital image processing.
- Provide students with practical exposure to image enhancement, compression, and segmentation.
- Explore advanced topics like deep learning applications in image processing.
- Develop proficiency in using industry-standard tools for solving real-world image processing
- problems.

Course Outcomes:
After completing the course, the students should be able to:
1. Understand and implement fundamental image processing techniques such as filtering, transformation, and restoration.
2. Analyze and interpret image data for applications in various domains like healthcare, surveillance, and entertainment.
3. Develop image processing solutions using modern frameworks and libraries (e.g., OpenCV, MATLAB, Python).
4. Explore and apply machine learning and deep learning algorithms for advanced image processing applications.

Unit - I
Introduction to Digital Image Processing: Fundamentals of Image Processing, Image Formation, Sampling, and Quantization, Types of Images: Grayscale, RGB, Multispectral, Image Representation: Pixels, Bit Depth, and Resolution, Color Models: RGB, HSV, CMY, YUV, File Formats: BMP, JPEG, PNG, TIFF.

Unit - II
Image Enhancement and Restoration: Spatial Domain Techniques, Histogram Equalization, Spatial Filtering: Smoothing, Sharpening, Edge Detection, Frequency Domain Techniques, Fourier Transform and Its Applications, Frequency Filters: Low-Pass, High-Pass, Band-Pass, Noise Reduction and Image Restoration Techniques.

Unit - III
Image Analysis and Compression: Image Segmentation, Thresholding (Global, Adaptive), Region-Based Segmentation, Morphological Operations: Erosion, Dilation, Opening, Closing, Image Features and Descriptors, SIFT, SURF, HOG, Image Compression, Lossless vs. Lossy Compression, JPEG Compression Steps and Implementation.

Unit - IV

Advanced Topics in Digital Image Processing: Image Classification and Object Detection, Introduction to Convolutional Neural Networks (CNNs), Pretrained Models (VGG, ResNet, YOLO), Image Denoising using Autoencoders, Applications in Healthcare and Surveillance, Introduction to Video Processing and Motion Analysis.

Suggested Readings

1. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods (Pearson).
2. Computer Vision: Algorithms and Applications by Richard Szeliski.
3. Deep Learning for Computer Vision with Python by Adrian Rosebrock.
4. Programming Computer Vision with Python by Jan Erik Solem.

List of Experiments (Based on Hardware/ Simulation)

1. Basics of Image Processing
   Objective: Load, display, and convert images between color spaces.
   Requirements: Python (OpenCV, PIL), MATLAB.
2. Image Enhancement
   Objective: Implement histogram equalization and various spatial filters.
   Requirements: Python/NumPy, MATLAB, sample images.
3. Frequency Domain Filtering
   Objective: Apply Fourier transforms and design frequency filters.
   Requirements: Python (SciPy), MATLAB, signal processing toolbox.
4. Noise Removal
   Objective: Remove noise using different filters and compare results.
   Requirements: Python, MATLAB.
5. Image Segmentation
   Objective: Apply thresholding and morphological operations to segment images.
   Requirements: Python (OpenCV), MATLAB.
6. Feature Detection
   Objective: Detect features using SIFT, SURF, and HOG descriptors.
   Requirements: OpenCV with Python, dataset of sample images.
7. Image Compression
   Objective: Implement JPEG compression steps manually.
   Requirements: Python (NumPy, PIL), MATLAB.
8. CNN-based Image Classification
   Objective: Build a CNN for handwritten digit classification (MNIST dataset).
   Requirements: Python (TensorFlow/Keras), GPU support (optional).
9. Object Detection using YOLO
   Objective: Detect objects in real-time using YOLO.
   Requirements: Python, pretrained YOLO weights, webcam.
10. Motion Tracking in Video
    Objective: Implement motion tracking using optical flow.
    Requirements: Python (OpenCV), video datasets.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Discipline Specific Core (DSC) Courses for B. Tech. (ECE) – Semester VI

Hands – On CMOS VLSI Design (DSC – 16)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Hands – On CMOS VLSI Design	4	3	0	1	Physics, Introduction to Electrical and Electronics Engineering

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Provide a solid grounding in MOSFET device physics, CMOS technology, and fabrication flows, enabling students to understand how semiconductor devices are realized at the transistor level.
- Equip students with the ability to design, analyze, and optimize CMOS circuits, both digital and analog, and to understand their trade-offs in terms of power, speed, area, and reliability.
- Encourage maximum practical exposure by integrating simulation, layout, prototyping, and measurement, ensuring students feel engaged and enjoy learning through experiential labs.
- Introduce a broad range of CMOS circuit applications—oscillators, amplifiers, memory cells, frequency dividers, data recovery circuits—to help students appreciate the diversity and complexity of real-world VLSI systems.

Course Outcomes:
After completing the course, the students should be able to:
1. Demonstrate a thorough understanding of MOSFET operation, CMOS fabrication steps, and the principles underlying CMOS inverters and logic gates.
2. Design, simulate, and layout fundamental CMOS building blocks and assess their performance via SPICE simulations, post-layout verification, and basic prototyping.
3. Apply CMOS principles to implement and analyze advanced circuit blocks such as latches, memory cells, oscillators, and frequency dividers, understanding their practical roles in larger systems.

Unit - I
Introduction to CMOS Devices and Fabrication: Review of MOSFET. CMOS Inverter (response and characteristics), CMOS basic fabrication flow. Layout Fundamentals: Design rules, stick diagrams, layout for matching and symmetry. SPICE simulation basics.

Unit - II
Core CMOS Digital Circuits: Basic Gates: NAND, NOR, XOR, and their transistor-level implementations. Tapered Buffer: Design concepts, driving large capacitive loads efficiently. CMOS Latches and Flip-Flops: T-Gate Static Latch Tri-state Static Latch SR Latch and Dynamic Latch. Memory Elements: Static Memory Cell (e.g., 6T SRAM cell).

Unit - III
Analog and Mixed-Signal CMOS Circuits: LNA (input matching, noise figure, and gain optimization). The Self-Based Inverter & Duty Cycle Correction Circuits. Active Inductor: Realizing inductive behavior using CMOS transistors. Transimpedance Amplifier (TIA): Current-to-voltage conversion, bandwidth, and noise considerations.

Unit - IV
Oscillators: Ring Oscillator, Differential & Quadrature Ring Oscillator, Current-Controlled.
Clock and Data Recovery (CDR): Burst-mode CDR circuits, lock time, jitter performance. Frequency Dividers: Feedforward frequency divider, frequency divider with quadrature output.

Suggested Readings

1. CMOS VLSI Design: A Circuits and Systems Perspective by Neil H. E. Weste and David Harris (Pearson)
2. Digital Integrated Circuits: A Design Perspective by Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolić (Prentice Hall)
3. Design of Analog CMOS Integrated Circuits by Behzad Razavi (McGraw-Hill)
4. CMOS Analog Circuit Design by Phillip E. Allen and Douglas R. Holberg (Oxford University Press)
5. Analysis and Design of Analog Integrated Circuits by Paul R. Gray, Paul J. Hurst, Stephen H. Lewis, and Robert G. Meyer (Wiley)
6. The Art of Analog Layout by Alan Hastings (Prentice Hall)
7. RF Microelectronics by Behzad Razavi (Prentice Hall)
8. VLSI Technology by S. M. Sze (McGraw-Hill)
9. CMOS IC Layout: Concepts, Methodologies, and Tools by Dan Clein (Newnes)
10. VLSI Design Techniques for Analog and Digital Circuits by R. L. Geiger, P. E. Allen, and N. R. Strader (McGraw-Hill)

List of Experiments
(Will be based on open-source/ Proprietary EDA tools such as Silvaco's TCAD or Synopsys Sentaurus)

1. MOSFET Characterization
   Objective: Extract threshold voltage, measure I-V curves, and observe channel length modulation.
2. CMOS Inverter DC and Transient Response
   Objective: Simulate and measure inverter transfer characteristics, noise margins, and propagation delay.
3. Inverter Layout and Verification (Cadence Virtuoso)
   Objective: Draw layout of a CMOS inverter, run DRC/LVS, extract parasitics, and re-simulate.
4. Tapered Buffer Design
   Objective: Optimize buffer chain for driving large loads; compare delay and power.
5. T-Gate Static Latch Implementation
   Objective: Implement a T-Gate latch using FPGA logic slices, verify timing and functionality.
6. Tri-State Static Latch and SR Latch
   Objective: Construct and test latch behavior, measure setup/hold times.
7. Dynamic Latch and Static Memory Cell Simulation
   Objective: Simulate and analyze read/write operations and cell stability.

8. Low-Noise Amplifier Design
   Objective: Design and simulate an LNA at a given frequency, measure gain and noise figure.
9. Self-Based Inverter & Duty Cycle Correction Simulation
   Objective: Adjust duty cycle and observe waveform integrity in simulation.
10. Active Inductor Implementation
    Objective: Build an active inductor using discrete transistors or op-amps and measure frequency response.
11. Transimpedance Amplifier Testing
    Objective: Connect a photodiode to a TIA and measure output voltage for various light intensities.
12. Ring Oscillator Frequency Measurement
    Objective: Simulate a ring oscillator, vary supply voltage and measure frequency changes.
13. Quadrature Ring Oscillator Simulation
    Objective: Achieve quadrature outputs, measure phase offset.
14. Current-Controlled / Digitally Controlled Oscillator
    Objective: Vary control signals and observe frequency tuning range.
15. Burst-Mode CDR Simulation
    Objective: Analyze lock time and jitter in a burst-mode CDR circuit.
16. Feedforward Frequency Divider Implementation
    Objective: Design and simulate a CMOS-based frequency divider, measure output frequency.
17. Frequency Divider with Quadrature Output
    Objective: Generate and verify quadrature phase signals.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Embedded Systems and Applications (DSC - 17)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Embedded Systems and Applications	4	3	0	1	Fundamentals of Computer Programming

Course Hours: L – 03, T – 00, P – 02

Course Objectives:

- To understand the architecture and programming of microprocessors and microcontrollers.
- To gain insight into interfacing peripheral devices with processors/microcontrollers.
- To familiarize students with ARM CPU architecture and programming.
- To develop the ability to design and implement embedded systems for real-world applications.

Course Outcomes:
After completing the course, the students should be able to:

1. Analyze the architecture and instruction set of 8085, 8086, and 8051 processors/microcontrollers.
2. Develop assembly-level programs for solving computational problems.
3. Design and interface peripheral devices using protocols like SPI, I2C, and UART.
4. Implement embedded solutions using ARM architecture and modern software development tools.

Unit - I
Introduction to 8085 basic/applications: 8086 Microprocessor: Internal architecture, Real mode memory addressing, Instruction Format. Addressing modes: Data-Addressing modes, Program Memory, Addressing modes, Stack Memory-Addressing modes.

Unit - II
Instruction Set: Programming 8086 using: Data movement instructions: MOV, PUSH/POP, Load-Effective Address, String data transfers, miscellaneous data transfer instructions, Arithmetic and logic instructions, BCD and ASCII arithmetic, Shift and Rotate, String comparisons, Program control instructions, Introduction to interrupts.

Unit - III
Peripheral Devices: 8255-Programmable Peripheral Interface, 8254- Programmable interval Timer. DMA: Introduction to Direct memory Access. Interrupts: Basic interrupt processing, Interrupt instructions, Operation of real mode interrupt, interrupt flag bits, Hardware interrupts. Introduction to 8051 microcontrollers. RISC vs. CISC Architecture.

Unit – IV
ARM CPU Architecture: Introduction to ARM Architecture Programmer's Model for ARM CPU Operating Modes, Instruction Set, Exception Handling, and Pipelining. ARM Cores (e.g., ARMv4). Peripheral-Processor Interfacing: Concepts of Peripheral Interfacing. Review of Peripheral Interface Protocols: SPI, I2C, UART, and One-Wire. Interface with Sensors, Radios, and ADCs. Embedded System Features: Hardware Timers and Interrupt Handling, Interrupt Service Routines.

Suggested Readings

1. The Intel Microprocessors: Architecture, Programming, and Interfacing by B. Brey (Pearson Education)
2. Microprocessors and Interfacing: Programming and Hardware by D. V. Hall (Tata McGraw Hill)
3. The x86 PC: Assembly Language, Design & Interfacing by Mazidi, Mazidi & Causey (Pearson Education)
4. An Embedded Software Primer by David E. Simon (Pearson Education)
5. Embedded Systems: Architecture, Programming, and Design by Raj Kamal (Tata McGraw Hill)
6. ARM System Developer's Guide by Andrew N. Sloss (Elsevier)
7. Embedded System Design by Frank Vahid and Tony Gwargie (John Wiley & Sons)
8. Embedded System Design by Steve Heath (Elsevier, Second Edition)
9. Embedded System Architecture: A Comprehensive Guide for Engineers and Programmers by Tammy Noergaard (Elsevier)
10. Programming Embedded Systems in C and C++ by Michael Barr (O'Reilly)

List of Experiments (Hardware Based)

1. Write an assembly language program to add two 8-bit and 16-bit hexadecimal numbers.
2. Write an assembly language program to transfer a block of data.
3. Write an assembly language program to multiply two 8-bit hexadecimal numbers.
4. Write an assembly language program to generate a Fibonacci series.
5. Write an assembly language program to sort hexadecimal numbers in ascending/descending order.
6. Study the working of IC 8255/8254 interfaced with the 8086 microprocessor.
7. Design and implement SPI and I2C-based communication for interfacing sensors with a microcontroller.
8. Develop an ARM-based interrupt-driven application using hardware timers.
9. Write an assembly language program to toggle an LED using the 8051 microcontroller.
10. Interface a 7-segment display with the 8051 microcontroller and display numbers.
11. Design a digital clock using the 8051 microcontroller and interfaced LCD.
12. Implement serial communication using the UART protocol on the 8051 microcontroller.
13. Write an ARM assembly program to add two numbers and display the result.
14. Implement an ARM-based program for a simple digital thermometer using ADC and a sensor interface.
15. Develop an ARM-based system to control a DC motor using PWM.
16. Interface an external EEPROM using I2C with an ARM microcontroller and perform read/write operations.
17. Implement interrupt-driven GPIO control on an ARM microcontroller.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Digital Communication Systems (DSC – 18)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Digital Communication Systems	4	3	0	1	Mathematic – II, Signals and Systems, Analog Communication Systems

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To introduce students to the fundamental concepts of digital communication systems.
- To enable the analysis and design of communication systems, focusing on performance under practical constraints like noise and interference.
- To incorporate hands-on experience with both hardware and software tools for implementing communication systems.
- To explore advanced and emerging topics like adaptive modulation, MIMO systems, and spread spectrum techniques.

Course Outcomes:
After completing the course, the students should be able to:
1. Design and analyze digital communication systems using theoretical concepts and practical tools.
2. Implement digital modulation, waveform coding, and baseband shaping using MATLAB, Simulink, and hardware kits.
3. Evaluate the performance of communication systems under various noise and channel conditions.
4. Explore and simulate advanced techniques like adaptive modulation, spread spectrum, and MIMO communication.

Unit - I
Transition from Analog to Digital Communication (Advantages/Disadvantages), Waveform Coding: Sampling Theorem for baseband and bandpass signals, Quantization and coding techniques: PCM, DPCM, Delta Modulation, and Adaptive Delta Modulation, Design and performance analysis of waveform coding systems.

Unit - II
Baseband Shaping: Discrete PAM signals and their power spectra. Pulse shaping and intersymbol interference (ISI). Nyquist criterion for distortionless transmission. Eye diagram analysis and equalization techniques. Concepts of Gram-Schmidt orthogonalization, Geometric Representation of Signals.

Unit - III
Detection and Estimation: Review of Gaussian Random Process, Detection of Known Signals in Noise, Optimum Threshold Detection, Optimum Receiver for AWGN Channel, Matched Filter and Correlation Receivers, Decision Procedure: Maximum A- Posteriori Probability Detector- Maximum Likelihood Detector, Probability of Error, Bit Error Rate.

Unit - IV
Digital modulation schemes: Shift Keying Methods (including ASK, FSK, Q/DQ/OQ/(pi/4)/8-/16-PSK, 16/64-QAM). Coherent M-ary Schemes, Non-Coherent Schemes.
Uncertainty, Information, Entropy, Source Coding, Huffman Coding, Shannon Fano Coding. Introduction to error control coding, Linear Block Code, Convolution Code.

Suggested Readings

1. Digital Communications by John G. Proakis (McGraw-Hill)
2. Principles of Communication Systems by Taub and Schilling (McGraw-Hill)
3. Wireless Communications: Principles and Practice by Theodore S. Rappaport (Pearson)
4. Information Theory, Inference, and Learning Algorithms by David J.C. MacKay (Cambridge)
5. Digital Communication Systems by Simon Haykin (John Wiley)
6. Modern Digital and Analog Communication by B.P. Lathi (Oxford)
7. Digital Communication by Sklar (Pearson)

List of Experiments (Hardware and Software using NI Multisim)

1. Design and Implementation of PAM Modulation and Demodulation Circuit
   Objective: Design a circuit for generating and demodulating a Pulse Amplitude Modulated (PAM) signal.
2. Design and Implementation of PWM and PPM Circuits
   Objective: Generate PWM and PPM signals using a monostable multivibrator and observe their waveforms.
3. Sampling and Reconstruction of Signals
   Objective: Design a sampling circuit using a sample-and-hold IC and reconstruct the original signal using a low-pass filter.
4. PCM Encoding and Decoding
   Objective: Implement PCM encoding and decoding using discrete components and compare the results with simulated output.
5. Delta Modulation and Demodulation
   Objective: Design and implement a delta modulation system using discrete components.
6. Eye Diagram Observation Using Oscilloscope
   Objective: Generate digital signals with ISI and observe their eye diagram on an oscilloscope.
7. Analyze the performance of waveform coding scheme.
8. Design and implement pulse shaping filters in MATLAB and verify Nyquist criterion.
9. Perform equalization experiments using DSP kits.
10. Implement digital modulation schemes: ASK
    Objective: Design and implement a circuit to perform ASK.
11. Implement digital modulation schemes: FSK
    Objective: Design and implement a circuit to perform FSK
12. Implement digital modulation schemes: PSK
    Objective: Design and implement a circuit to perform PSK.
13. Perform BER analysis for different modulation schemes under noise using MATLAB.
14. Matched Filter Receiver Design
    Objective: Design and analyze a matched filter receiver for detecting signals in noise.
15. Signal Detection in Noise
    Objective: Implement a signal detection circuit using thresholding and analyze its performance with varying noise levels.

16. QPSK Modulation and Demodulation
    Objective: Design a QPSK modulator and demodulator circuit using hardware.
17. Perform adaptive modulation experiments using MATLAB.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Discipline Specific Elective (DSE) Courses for B. Tech. (ECE) – Semester VI

Wireless Sensor Networks (DSE – 4)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Wireless Sensor Networks	4	3	0	1	Fundamentals of Computer Programming, Signals and Systems

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To understand principles of sensor networks and its difference with mobile ad hoc networks.
- To evaluate computations related to energy saving using different routing schemes.
- To analyse different MAC protocols used for different communication standards in WSN.
- To design small sensor networks for different applications.

Course Outcomes:
After completing the course, the students should be able to:
1. Understand principles of sensor networks and its difference with mobile ad hoc networks.
2. Understand the concepts of Middleware and Transmission Technologies.
3. Evaluate computations related to energy saving using different routing schemes.
4. Analyse different MAC protocols used for different communication standards in WSN.
5. Design small sensor networks for different applications.

Unit - I
Cellular and Ad hoc wireless Networks: Mobile Ad-Hoc Networks, Sensor Networks. Applications, Categories, Issues and challenges in designing, Operating environment (Propagation). Architecture: Sensor node technology, Hardware and Software, Performance Metrics.

Unit - II
Middleware Functions, Architecture, Data management functions, Operating Systems, Design issues, Available wireless Technologies: WSN, Bluetooth, WLAN, Zigbee, WiMax, 3G and beyond. Performance modelling of WSN.

Unit - III
Fundamentals of MAC Protocols: Requirements and Design Constraints. Overview of MAC Protocols for WSN. Schedule-Based Protocols: SMAC, LEACH, and TRAMA. Contention-Based Protocols: CSMA and Slotted CSMA-CA Protocol. Non-beaconed Mode in MAC. Sensor MAC Protocol: Overview, Periodic Listen and Sleep Operations. Schedule Selection and Coordination, Synchronization, and Adaptive Listening. Access Control, Data Exchange, and Message Passing.

Unit - IV

Routing Protocols: Challenges, Issues, and Data Dissemination/Gathering. Location Discovery and Routing Strategies: Flooding, Gossiping, SPIN, PEGASIS, GEAR. Attribute-Based Routing: Direct Diffusion, Rumor Routing, Geographic Hash Tables. Transport Protocols: Design Issues and Feasibility of TCP/UDP for WSN. Advanced Protocols: CODA and GARUDA. Network Management: Techniques for Managing WSN Performance.

Suggested Readings

1. Fundamentals of Wireless Sensor Networks: Theory and Practice by Waltenegus Dargie and Christian Poellabauer (John Wiley & Sons Publications)
2. SENSORS Handbook by Sabrie Soloman (McGraw Hill Publication)
3. Wireless Sensor Networks by Feng Zhao and Leonidas Guibas (Elsevier Publications)
4. Wireless Sensor Networks: Technology, Protocols, and Applications by Kazem Sohraby and Daniel Minoli (Wiley-Interscience)

List of Experiments

1. Sensor Node Architecture Exploration
   Objective: Understand sensor node hardware and software.
   Requirements: Arduino/NodeMCU, temperature sensor, humidity sensor, multimeter.
2. Signal Propagation in Wireless Networks
   Objective: Study propagation impairments in WSNs.
   Requirements: RF transceiver modules, spectrum analyzer, wireless signal simulator.
3. Middleware Functions for WSNs
   Objective: Explore middleware functions and data management.
   Requirements: Raspberry Pi, Python, SQLite database.
4. Performance Modeling of WSN
   Objective: Analyze performance metrics of WSNs.
   Requirements: MATLAB/NS3 simulator.
5. Bluetooth and Zigbee Communication
   Objective: Implement WSN communication using Bluetooth and Zigbee.
   Requirements: Bluetooth modules, Zigbee modules, Arduino/NodeMCU.
6. Implementation of SMAC Protocol
   Objective: Implement and evaluate SMAC for energy-efficient communication.
   Requirements: NS3 simulator
7. Study PHY and MAC layer operations of IEEE 802.15.4.
   Requirements: Zigbee modules, NS3 simulator.
8. Explore LEACH and TRAMA protocols.
   Requirements: NS3 simulator, PCs.
9. Routing Protocol Implementation
   Objective: Implement and compare different routing strategies.
   Requirements: NS3 simulator, PCs.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Artificial Intelligence in Electronics (DSE – 4)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Artificial Intelligence in Electronics	4	3	0	1	Fundamentals of Computer Programming, Mathematics - I

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Introduce fundamental AI and machine learning concepts relevant to electronics and signal processing.
- Explore AI algorithms on electronic hardware platforms (microcontrollers, FPGAs, ASICs, SoCs) for real-time applications.
- Apply ML techniques to sensor signals (vision, audio, RF, biomedical) to develop intelligent systems.
- Expose students to advanced concepts like neuromorphic computing, low-power edge AI, and AI accelerators.
- Provide hands-on experience from simulation to hardware prototyping, ensuring job-ready skills.

Course Outcomes:
After completing the course, the students should be able to:
1. Implement AI/ML models for classification, regression, and pattern recognition in electronics.
2. Integrate and optimize ML algorithms for embedded platforms considering power, performance, and area.
3. Design AI-based intelligent embedded solutions like smart sensors and predictive maintenance modules.
4. Use AI/ML libraries (TensorFlow, PyTorch, scikit-learn) and hardware tools.

Unit - I
Foundations of AI and ML for ECE: AI, ML, and Deep Learning: Key concepts and relevance to ECE. Supervised & Unsupervised Learning: Linear regression, logistic regression, k-means clustering. Classification & Regression Techniques: SVMs, Decision Trees, Random Forests. Basics of Neural Networks: Perceptron, Multi-layer Perceptrons, activation and loss functions. Gradient Optimization, Backpropagation, Regularization Techniques. Python-based ML frameworks: scikit-learn, TensorFlow.

Unit - II
Embedded and Hardware-Aware AI: Hardware Basic: FPGAs, and AI accelerators. Real-time Constraints: Latency, throughput, memory optimization. Model Compression & Quantization for Edge Devices. AI on Microcontrollers & Low-power SoCs (ARM Cortex-M, RISC-V). FPGA-based AI design flows and high-level synthesis (HLS) for ML.

Unit - III
Intelligent Signal Processing & Sensor Fusion: Signal Processing with AI: Denoising, filtering, feature extraction. Computer Vision & Image Processing: CNNs, object detection, classification. Speech & Audio Processing: RNNs, LSTMs, Transformer-based models. Sensor Fusion & IoT: Multi-sensor integration and ML algorithms. Applications: Smart homes, wearables, biomedical, industrial IoT. Model Distillation.

Unit - IV
Advanced Topics & Emerging Trends: Neuromorphic Computing: Spiking Neural Networks, event-driven sensors. Edge AI & TinyML: Low-power inference, federated learning. Neural Architecture Search (NAS) & AutoML: Automated model optimization. Domain-Specific AI Accelerators: ASIC design for ML inference. Case Studies: Robotics, drones, biomedical diagnostics. Model Distillation.

Suggested Readings

1. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (MIT Press, 2016)
2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurelien Geron (O'Reilly, 2nd Edition)
3. Neural Networks and Learning Machines by Simon Haykin (Pearson, 3rd Edition)
4. Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig (Pearson, 4th Edition)
5. TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers by Pete Warden and Daniel Situnayake (O'Reilly)
6. Vivado Design Suite User Guides (Xilinx) and Intel Quartus Documentation for HLS
7. RISC-V-Based SoC Design by S. Wong et al. (Springer)

List of Experiments (Based on Simulation)

1. Basic ML Implementation
   Objective: Train a simple linear regression and classification model on a given dataset.
2. Neural Network for Signal Classification
   Objective: Implement a small MLP to classify sine vs. square waves or detect a known waveform.
3. Edge AI on a Microcontroller
   Objective: Deploy a compressed NN model (tiny model trained offline) onto an Arduino or STM32 board to classify simple sensor data (e.g., temperature thresholds or vibration patterns).
   Requirements: Trained tiny model (tflite), Arduino/STM32 board.
4. Computer Vision on Embedded Platform (Raspberry Pi + Camera)
   Objective: Run a CNN-based object detection or face recognition model on a Raspberry Pi with a camera feed.
   Requirements: Raspberry Pi, camera module, pre-trained CNN model (MobileNet or YOLO-tiny), OpenCV.
5. Audio Keyword Spotting
   Objective: Train a simple keyword spotting model (e.g., detecting the word "Hello") and deploy it on an NVIDIA Jetson Nano.
   Requirements: Microphone, recorded audio samples, TensorFlow Lite, Jetson Nano.
6. FPGA-based Inference Accelerator (to lean hardware acceleration)
   Objective: Implement a basic neural network inference engine on an FPGA using HLS.
   Requirements: FPGA board (PYNQ-Z2), Vivado, a small pre-trained NN model.
7. Sensor Fusion Project
   Objective: Combine data from an IMU (accelerometer/gyroscope) and a camera feed, run a simple fusion model to determine object orientation.
   Requirements: Raspberry Pi/STM32 with IMU and camera, Python, fusion algorithm.
8. Spiking Neural Network
   Objective: Implement a Spiking Neural Network simulation and observe event-driven processing.
   Requirements: Python-based SNN simulator (e.g., Brian2), synthetic spiking data.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Coding and Data Compression Techniques (DSE – 4)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Coding and Data Compression Techniques	4	3	0	1	Fundamentals of Computer Programming, Mathematics – II, Signals and Systems

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- Understand the fundamental concepts of information theory, entropy, and mutual information.
- Learn about source coding, data compression techniques, and error control coding.
- Explore the theoretical and practical aspects of channel capacity and coding theorems.
- Gain proficiency in advanced coding techniques like convolutional codes, Turbo coding, and universal source coding.
- Develop analytical skills to compute rate-distortion functions and implement algorithms such as Lempel-Ziv and Blahut-Arimoto.

Course Outcomes:
After completing the course, the students should be able to:

1. Analyze and calculate information measures such as entropy and mutual information.
2. Design efficient codes for data compression and error control.
3. Evaluate the capacity of different communication channels and understand their theoretical limits.
4. Implement universal coding algorithms for data compression.
5. Apply advanced error control and source coding techniques to practical communication problems.

Unit - I
Entropy, Relative Entropy, and Mutual Information: Entropy, Joint Entropy and Conditional Entropy, Relative Entropy and Mutual Information, Chain Rules, Data-Processing Inequality, Fano's Inequality. Source Coding and Data Compression: Kraft Inequality, Huffman Codes, Optimality of Huffman Codes, Shannon Fano Codes.

Unit - II
Channel Capacity: Symmetric Channels, Properties of Channel Capacity, Jointly Typical Sequences, Channel Coding Theorem, Fano's Inequality and the Converse to the Coding Theorem. Variable to Block Length Coding: The Asymptotic Equipartition Property Block to Block Coding of Discrete Memoryless Sources (DMS). Rate Distortion Theory, Blahut-Arimoto Algorithm for Channel Capacity and Rate-Distortion Function.

Unit - III
Standard Error control coding- Block codes: Definitions and Principles: Hamming weight, Hamming distance, Minimum distance decoding - Single parity codes, Hamming codes, Repetition codes -Linear block codes. Cyclic codes - Syndrome calculation, Encoder and decoder – CRC

Unit - IV
Error control coding- convolution codes: code tree, trellis, state diagram, Encoding – Decoding: Sequential search and Viterbi algorithm – Principle of Turbo coding. Universal Source Coding: Lempel-Ziv Algorithm, Introduction to Reed Solomon (RS) Codes.

Suggested Readings

1. Elements of Information Theory by T. M. Cover, and J. A. Thomas (Wiley).
2. Channel Codes: Classical and Modern by William Ryan, and Shu Lin (Cambridge).
3. Information Theory and Reliable Communication by Robert Gallager (Wiley).
4. Error Control Coding by Shu Lin and Daniel Costello Jr (Pearson).
5. Theory and Practice of Error Control Codes by Richard E. Blahut (Addison-Wesley).

List of Experiments (Hardware/ Software based)

1. Implement entropy calculation and mutual information for given data sets.
2. Design Huffman and Shannon-Fano codes for a given source.
3. Simulate the channel coding theorem.
4. Evaluate Gaussian channel capacity and compute rate-distortion functions using the Blahut-Arimoto algorithm.
5. Analyze the performance of single parity codes.
6. Design and implement Hamming codes.
7. Implement cyclic redundancy check (CRC) encoding and decoding.
8. Implement Lempel-Ziv algorithms for text compression.
9. Simulate linear block codes.
10. Simulate convolutional codes.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Electromagnetic Compatibility: Principles and Applications (DSE – 4)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Electromagnetic Compatibility: Principles and Applications	4	3	0	1	Physics, Network Analysis and Synthesis, Signals and Systems, Electromagnetic Theory

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To provide a comprehensive understanding of electromagnetic compatibility (EMC) principles and challenges in electronic systems.
- To enable students to analyze and design electronic systems that minimize electromagnetic interference (EMI) and ensure compliance with EMC standards.
- To familiarize students with EMC testing, measurement techniques, and the use of computational modeling tools.
- To equip students with hands-on skills in designing noise-suppressed circuits, effective PCB layouts, and implementing shielding and ESD protection strategies.

Course Outcomes:
After completing the course, the students should be able to:
1. Develop the ability to identify and classify EMC problems in electronic systems and propose appropriate mitigation strategies.
2. Gain proficiency in utilizing simulation tools and experimental setups to evaluate and optimize EMC performance.
3. Design and implement EMC-compliant PCBs, shielding solutions, and noise suppression techniques in practical applications.
4. Demonstrate knowledge of international EMC standards, testing protocols, and regulatory requirements for electronic systems.

Unit - I
Fundamentals of EMC and Standards: Introduction to EMC: Problem classifications, physical and electrical dimensions of components. Common EMC units and their significance. Basics of transmission line theory for EMC applications. Overview of EMC signal sources and their characterization. EMC Standards: Conducted emissions standards and testing. Radiated emissions standards and testing. Regulatory frameworks: FCC, CISPR. Antenna factors and their importance in EMC measurements.

Unit - II
Component Behavior and Noise Suppression: Low- and high-frequency approximations of circuit components. Internal impedance of round wires and PCB traces. External inductance, capacitance, and conductance in coaxial and parallel wires. Nonideal behavior of resistors, capacitors, and inductors. Common mode vs. differential mode currents. Use of ferrites and common mode chokes for noise suppression. Digital circuit devices and their role in EMC.

Unit - III

Signal Spectra and Radiated Emissions: Classification of signals: Periodic and aperiodic. Fourier series and transforms for signal spectrum analysis. Spectra of digital clock waveforms and their implications. Emission models for wires, PCB lands, and their corresponding signal spectra. Measured spectra and antenna factor effects on emissions.

Unit - IV

Crosstalk, Shielding, and PCB Design for EMC: Crosstalk in multi-conductor systems: Per-unit-length parameters and time-domain analysis. Shielding: Far-field and near-field effectiveness. EMC/EMI computational modeling: FDTD and Method of Moments techniques. PCB Design for EMC: Board stack-up issues. Component placement and signal isolation strategies. Electrostatic Discharge (ESD): Mechanisms, models (Human Body Model), and protection techniques.

Suggested Readings

1. Introduction to Electromagnetic Compatibility by Clayton R. Paul (John Wiley & Sons, 2nd Ed., 2006).
2. Noise Reduction Techniques in Electronic Systems by Henry W. Ott (John Wiley & Sons, 2nd Ed., 1988).
3. PCB Design for Real-World EMI Control by Bruce R. Archambeault (Springer Science + Business Media, LLC, 2002).
4. EMC for Product Designers by Tim Williams (Elsevier, 5th Ed., 2016).
5. EMC of Analog Integrated Circuits by Christoph Schubert & Bruno Schumacher (Springer, 2015).
6. Electromagnetic Compatibility: Principles and Applications by David Weston (Marcel Dekker, Inc., 2nd Ed., 2001).

List of Experiments (Hardware/ Software based)

1. Signal Characterization:
   Generate and analyze periodic and aperiodic signals using oscilloscopes and spectrum analyzers. Software implementation of Fourier transforms for signal spectra (MATLAB/Python).
2. PCB Design for EMC:
   Design and test PCB layouts for minimal crosstalk and enhanced signal isolation.
3. Noise Suppression Techniques:
   Construct circuits with ferrites, capacitors, and inductors to suppress common mode noise.
4. ESD Testing:
   Measure and protect against electrostatic discharges using protection circuits.
5. EMC Shielding:
   Evaluate shielding effectiveness for near-field and far-field scenarios.
6. Simulation of Transmission Line Behavior:
   a. Use simulation tools (e.g., LTSpice, ADS, or CST Studio) to model and analyze the behavior of transmission lines under different signal frequencies.
   b. Evaluate the effects of impedance mismatches on signal integrity.
7. Modeling EMC Signal Sources:
   a. Simulate different EMC signal sources, such as sinusoidal, square, and pulse waveforms, using MATLAB/Simulink or Python.
   b. Analyze their spectra and relate them to real-world EMC challenges.
8. Analysis of Non-Ideal Component Behavior:
   a. Simulate inductors, capacitors, and resistors at high frequencies using tools like LTSpice or COMSOL Multiphysics.
   b. Compare the simulated performance with ideal models and observe deviations.

9. Fourier Transform Analysis of Signal Spectra:
   Objective: Implement Fourier transforms in MATLAB or Python to analyze the spectra of periodic and aperiodic signals.
10. Crosstalk Simulation:
    a. Model multi-conductor transmission lines in CST Studio or Ansys HFSS.
    b. Simulate and visualize crosstalk between adjacent traces or cables and identify mitigation strategies.
11. Shielding Effectiveness Analysis:
    a. Simulate shielding effectiveness in near-field and far-field scenarios using CST Studio or COMSOL Multiphysics.
    b. Compare different materials and configurations for their shielding performance.
12. Modeling Electrostatic Discharge (ESD) Events:
    a. Use SPICE-based tools or MATLAB to simulate ESD events and observe their impact on circuits.
    b. Implement and test ESD protection circuits virtually.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

List of Discipline Specific Elective (DSE)/ Generic Elective (GE) courses offered for Minors/ Specializations by the Department in Second Year

1. **Minor in ECE (Offered only to CSE and EE)**
   a) DSE - 3/ GE - 5: Communication Architecture
   b) DSE - 4/ GE - 6: Antenna and Wave Propagation
      Or
      Elements of Wireless Communication

2. **Minor/ Specialization in Telecommunication Networks (Offered to ECE, CSE, and EE)**
   a) DSE - 3/ GE - 5: Wireless Communication and Mobile Networks
   b) DSE - 4/ GE - 6: Satellite Communication
      Or
      Cognitive Radio & Networks

3. **Minor/ Specialization in VLSI Technology and System Design (Offered to ECE, CSE, and EE)**
   a) DSE - 3/ GE - 5: Advanced Digital VLSI Circuits and Physical Design
   b) DSE - 4/ GE - 6: CMOS Analog IC Design
      Or
      Current Mode Analog VLSI Circuits

4. **Minor/ Specialization in IoT System Design (Offered to ECE, CSE, and EE)**
   a) DSE - 3/ GE - 5: Introduction to Security of Cyber-Physical Systems
   b) DSE - 4/ GE - 6: Ubiquitous Sensing, Computing and Communication
      Or
      Introduction to Embedded Systems for IoT

5. **Minor/ Specialization in Computer Vision (Offered to ECE, CSE, and EE)**
   a) DSE - 3/ GE - 5: Intelligent Imaging
   b) DSE - 4/ GE - 6: Deep Learning for Image Analysis
      Or
      Deep & Reinforcement Learning

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Generic Elective (GE) courses offered for Minors/ Specializations by the Department in Semester V

Communication Architecture (DSE – 3/ GE – 5)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Communication Architecture	4	3	0	1	NIL

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To understand the fundamental principles of analog and digital communication systems.
- To familiarize students with communication architectures and their practical implementations.
- To develop hands-on skills in designing and analyzing communication systems using modern tools.
- To prepare students for advanced topics in wireless communication and antenna theory.

Course Outcomes:
After completing the course, the students should be able to:
1. Explain the structure and functionality of analog and digital communication systems.
2. Design and simulate basic communication architectures using appropriate software and hardware tools.
3. Analyze and troubleshoot communication systems for practical applications.
4. Transition smoothly to advanced topics like antennas and wireless communication.

Unit - I
Fundamentals of Communication Systems: Introduction to communication: Analog vs. Digital communication. Basic elements of a communication system: Transmitter, channel, receiver. Signal modulation: Amplitude Modulation (AM), Frequency Modulation (FM), and Phase Modulation (PM). Noise in communication systems and signal-to-noise ratio (SNR). Practical communication channels and their characteristics.

Unit - II
Digital Communication Systems: Sampling theorem and quantization. Pulse Code Modulation (PCM), Differential PCM, and Delta Modulation. Digital Modulation Techniques: ASK, PSK, FSK, QAM. Introduction to error detection and correction. Basics of Multiplexing: Time Division Multiplexing (TDM) and Frequency Division Multiplexing (FDM).

Unit - III
Communication Architectures and Protocols: Introduction to layered communication architecture (OSI and TCP/IP models). Physical and Data Link Layer protocols. Basics of wired communication standards (Ethernet) and wireless standards (Wi-Fi, Bluetooth). Signal propagation and channel impairments.

Unit - IV
Advanced Communication Concepts: Introduction to Spread Spectrum Techniques: DSSS and FHSS. Basics of MIMO (Multiple Input Multiple Output) systems. Introduction to IoT communication protocols: MQTT and Zigbee. Overview of Software Defined Radio (SDR) and Cognitive Radio.

Suggested Readings

1. Modern Digital and Analog Communication Systems by B.P. Lathi (Oxford).
2. Communication Systems by Simon Haykin (Wiley).
3. Digital Communications by John G. Proakis (McGraw-Hill).
4. Wireless Digital Communications by K. Feher (Prentice Hall).
5. Wireless Communications by Andrea Goldsmith (University Press).
6. Wireless Communications: Principles and Practice by Theodore S. Rappaport (Pearson).

List of Experiments (Hardware on Breadboard / Software using NI Multisim)

1. Design and implement an AM transmitter and receiver.
2. Design and implement Pulse Code Modulation (PCM).
3. Design and Simulate ASK (MATLAB or Simulink and Hardware).
4. Design and Simulate PSK (MATLAB or Simulink and Hardware).
5. Design and Simulate QAM (MATLAB or Simulink and Hardware).
6. Simulate a basic wireless communication link using SDR tools.
7. Measure signal impairments in a communication link using spectrum analyzers.
8. Build a simple IoT communication system using MQTT on NodeMCU.
9. Simulate a basic MIMO system using MATLAB or Simulink.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Wireless Communication and Mobile Networks (DSE – 3/ GE – 5)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Wireless Communication and Mobile Networks	4	3	0	1	NIL

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- To introduce the basic cellular concepts.
- To understand the various signal propagation effects.
- To study various multiple access schemes.
- To familiarize with various mobile standards.
- To study the implementation of MIMO systems for high-speed communication

Course Outcomes:
After completing the course, the students should be able to:

1. Understand cellular concepts and signal propagation in mobile communication.
2. Understand the fundamental principles of signal propagation in wireless communication.
3. Compare and contrast various multiple access techniques and their suitability for different applications. Evaluate the advantages and limitations of different modulation schemes
4. Understand the architecture of mobile communication networks, including GSM, CDMA, 3G, 4G LTE, and 5G.
5. Design MIMO configurations to enhance wireless communication systems.

Unit - I
Wireless Channel Characteristics & Cellular Concepts: Cellular concepts: Cell structure, frequency reuse, cell splitting, channel assignment, handoff mechanisms, interference management, capacity planning, power control. Antennas for mobile communication: Mobile terminal antennas (monopole, PIFA), base station antennas, antenna arrays. Propagation mechanisms: Reflection, refraction, diffraction, scattering Large-scale signal propagation models and lognormal shadowing.

Unit - II
Fading Channels & Their Characterization: Fading channels: Multipath and small-scale fading, Doppler shift, statistical channel models. Narrowband and wideband fading models, power delay profile, RMS delay spread. Coherence bandwidth, coherence time, flat vs. frequency-selective fading, slow vs. fast fading, average fade duration, level crossing rate. Channel capacity for flat and frequency-selective fading channels.

Unit - III
Multiple Access Techniques & Receiver Architectures: Multiple access techniques: FDMA, TDMA, CDMA, SDMA, OFDMA. Receiver design: Diversity receivers (selection, MRC), RAKE receiver, Equalization (linear-ZFE, adaptive, DFE). Transmit diversity: Alamouti scheme. Introduction to spread spectrum techniques. Brief introduction to WLAN (Wi-Fi) and PAN (Bluetooth, Zigbee)

Unit - IV
Wireless Standards, Advanced Technologies & Network Architectures: Cellular standards evolution: Overview of 2G (GSM), 3G (WCDMA/UMTS), 4G (LTE), 5G NR, and emerging 6G. System examples: GSM, EDGE, GPRS, IS-95, CDMA2000, WCDMA. MIMO and space-time signal processing: Spatial multiplexing, diversity/multiplexing trade-off. Performance measures: Outage probability, average SNR, BER, QoS considerations. Network architectures: Mobile IP, mobility management, LTE/EPC architecture (MME, SGW, PGW), planning & optimization

Suggested Readings

1. 4G, LTE-Advanced Pro and The Road to 5G (Third Edition, 2016) by Erik Dahlman (Academic Press)
2. 5G NR: Architecture, Technology, Implementation, and Operation of 3GPP New Radio Standards (2019) by Sassan Ahmadi (Academic Press)
3. Wireless Communication and Networking by Vijay K. Garg (Elsevier, Morgan Kaufmann, 2012)
4. Wireless Communications: Principles and Practice by T. S. Rappaport (PHI, 2006)
5. Mobile Cellular Telecommunications: Analog and Digital Systems by William Lee (McGraw Hill Education, 2017)
6. Fundamentals of Wireless Communication by David Tse and Pramod Viswanath (Cambridge University Press, 2005)
7. Mobile Communications by Jochen Schiller (Addison-Wesley, 2003)
8. LTE for UMTS: Evolution to LTE-Advanced by Harri Holma and Antti Toskala (Wiley, 2007)
9. Wireless Communication Networks and Systems by Cory Beard and William Stallings (Pearson, 2016)

List of Experiments (Hardware on Breadboard / Software using NI Multisim)

1. Measure path loss, received signal strength, and antenna radiation patterns using a signal generator and spectrum analyzer.
   Evaluate monopole and PIFA antennas for mobile terminals.
2. Channel emulators to observe multipath fading effects, measure delay spreads, and analyze fading statistics.
3. Implement simple modulation/demodulation schemes (QPSK, OFDM) and observe BER under varying channel conditions.
4. Simulate cellular networks, analyze handover strategies, and evaluate resource scheduling algorithms.
5. Antenna Orientation Experiment:
   a. Attach a simple dipole or whip antenna to a transmitter and measure received power at a fixed receiver position.
   b. Rotate the antenna at various angles and observe how polarization and orientation affect the signal.
6. Basic Network Simulator Introduction (NS-3):
   a. Run a basic NS-3 simulation for a simple point-to-point wireless scenario.
   b. Compare throughput and delay under different traffic loads and link distances.
7. Indoor/Outdoor Path Loss Comparison:
   a. Use a simple RF transmitter-receiver pair (e.g., low-cost FM module or a simple 2.4 GHz transceiver) to measure signal strength indoors and outdoors.
   b. Record how the signal attenuates with distance in each environment.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Advanced Digital VLSI Circuits and Physical Design (DSE – 3/ GE – 5)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Advanced Digital VLSI Circuits and Physical Design	4	3	0	1	Physics, Microelectronics Design

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Equip students with advanced knowledge of CMOS logic families, dynamic logic, and memory elements beyond basic combinational and sequential logic.
- Expose students to the back-end design flow, including floorplanning, placement, routing, design rule checks, and layout verification.
- Enable students to use industry-standard EDA tools for implementing, verifying, and optimizing digital circuits at both the schematic and layout levels.
- Familiarize students with timing closure, advanced node scaling challenges, and emerging trends like 3D IC integration and advanced packaging.
- Provide a learning experience that balances theory and hands-on labs, encouraging creative problem-solving and teamwork through mini-projects and gamified activities.

Course Outcomes:
After completing the course, the students should be able to:
1. Create and optimize advanced digital logic circuits (dynamic logic gates, memory cells, arithmetic blocks) with a clear understanding of performance, area, and power.
2. Perform floorplanning, placement, routing, and post-layout verification using professional EDA tools, ensuring DRC/LVS compliance and timing closure.
3. Understand and apply methods for static timing analysis, power optimization (clock gating, multi-V_t design), and signal integrity checks.
4. Gain awareness of advanced nodes, FinFET/3D integration, advanced packaging, and their implications on design methodologies.

Unit - I
Advanced Digital CMOS Circuits: Review of CMOS Logic & Static CMOS: Brief refresher of CMOS inverter, NAND, NOR, complex gates, Dynamic Logic & Domino Circuits: Principles, advantages/ disadvantages, charge sharing, noise margins, Pass-Transistor & Transmission Gate Logic: Design techniques, area and speed considerations, Memory Elements (SRAM, ROM, CAM Basics): 6T SRAM cell layout, read/write operations, timing margins, Case Studies: Arithmetic circuits (adders, multipliers), pipeline registers.

Unit - II
Physical Design Flow & Methodologies: Introduction to ASIC & Full-Custom Flows: Standard-cell methodology, IP reuse, PDKs, Floorplanning & Partitioning: Hierarchical design, IO and macro placement, minimizing wirelength and congestion, Placement & Routing: Timing-driven placement, global vs. detailed routing strategies, crosstalk avoidance, Design Rule Checking (DRC) & Layout vs. Schematic (LVS): Physical verification fundamentals, running DRC/LVS tools, fixing violations.

Unit - III
Timing Closure & Power Optimization: Static Timing Analysis (STA): Setup/hold timing, clock skew, Clock Tree Synthesis & Optimization: Balancing skew, buffering, clock gating for power reduction, Power Optimization Techniques: Multi-threshold CMOS, power gating, dynamic voltage/frequency scaling, Signal Integrity & Parasitic Extraction: Crosstalk, IR drop, RC extraction, post-layout optimization.

Unit - IV
Advanced Transistor Technologies: FinFETs, Gate-All-Around FETs, impact on design rules and PPA (Power-Performance-Area), 3D ICs & Advanced Packaging: Through-Silicon Vias (TSVs), chiplets, heterogeneous integration, Reliability & Variability: Aging, NBTI, EM, process variation, methods to mitigate reliability issues.

Suggested Readings

1. CMOS VLSI Design: A Circuits and Systems Perspective by N.H.E. Weste and D. Harris (Pearson).
2. CMOS Digital Integrated Circuits: Analysis and Design by Sung-Mo (Steve) Kang, Yusuf Leblebici (McGraw-Hill).
3. VLSI Physical Design Automation by Sadiq M. Sait and Habib Youssef (McGraw-Hill).
4. Synthesis and Optimization of Digital Circuits Giovanni De Micheli (McGraw-Hill).
5. Digital Integrated Circuits: A Design Perspective by Jan M. Rabaey, Anantha Chandrakasan, Borivoje Nikolic (Pearson).

List of Experiments (Hardware / Software based)

1. Complex Gate Design & Simulation:
   Task: Design and simulate a complex AOI/OAI gate at transistor-level using SPICE. Analyze delay, power, and voltage transfer characteristics.
2. Dynamic Logic Implementation:
   Task: Implement a domino logic gate and verify its operation for a given digital function. Investigate charge leakage and noise margins.
3. SRAM Cell Layout & Verification:
   Task: Layout a 6T SRAM bit-cell and perform DRC and LVS checks. Extract parasitics and run post-layout simulation to verify read/write stability.
4. RTL-to-Gates Synthesis (Simple ALU Block):
   Task: Take a behavioral RTL code of an 8-bit ALU, synthesize it to gates.
5. RTL-to-Gate Synthesis
   Task: Synthesize an RTL description of a small arithmetic module (e.g., a 4-bit multiplier) into a gate-level netlist using a standard-cell library.
6. Floorplanning & Placement (Software: ICC2 or Cadence Innovus)
   Task: Perform floorplanning and placement of the synthesized netlist.
7. Routing & DRC/LVS Checks (Software: ICC2 + Calibre/ICV)
   Task: Route the design and verify it against foundry DRC/LVS rules.
8. Advanced Node Exploration (FinFET PDK):
   Task: Using a simplified FinFET PDK, implement a small inverter and compare area, delay, and power to planar CMOS.

9. FinFET vs. Planar CMOS Device Simulation (Software: Silvaco)
   Task: Simulate a FinFET structure and compare its I-V characteristics, threshold voltage, and short-channel effects to a traditional planar MOSFET.

10. Reliability & Aging Simulation
    Task: Model NBTI aging in a PMOS device and observe how threshold voltage shifts over time.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Introduction to Security of Cyber-Physical Systems (DSE – 3/ GE – 5)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Introduction to Security of Cyber-Physical Systems	4	3	0	1	Introduction to IoT, Introduction to IoT System Design

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To introduce fundamental concepts of security in IoT-based Cyber-Physical Systems (CPS).
- To understand security challenges, vulnerabilities, and risks in CPS.
- To explore practical solutions for securing CPS using hands-on implementations.

Course Outcomes:
After completing the course, the students should be able to:
1. Identify security challenges and threats in CPS.
2. Implement basic security mechanisms for IoT systems.
3. Analyze case studies and scenarios for real-world CPS security issues.

Unit - I
Fundamentals of CPS Security: Overview of Cyber-Physical Systems (CPS) and IoT. Security needs in CPS: Confidentiality, Integrity, Availability (CIA). Common threats and attack vectors in IoT and CPS (DDoS, Malware, Eavesdropping, Physical Tampering). Overview of Security Standards: NIST Cybersecurity Framework, ISO/IEC 27001.

Unit - II
Authentication and Access Control in CPS: Cryptographic Basics: Hashing, Encryption (AES, RSA), Digital Signatures. Authentication mechanisms for IoT devices. Access Control Policies: Role-Based Access Control (RBAC), Attribute-Based Access Control (ABAC). Secure communication using TLS/SSL and HTTPS.

Unit - III
Intrusion Detection and Mitigation Techniques: Intrusion Detection Systems (IDS) for IoT. Malware analysis and prevention. Secure device boot and firmware updates. Lightweight security protocols for resource-constrained devices.

Unit - IV
Software-Defined Networks: Introduction, Security. CSP – Platform, Components, Implementation Issues, Intelligent CPS, Secure Deployment of CPS. Designing a secure CPS system: Practical insights. Introduction to Blockchain for IoT security.

Suggested Readings

1. Cybersecurity for IoT by Brian Russell and Drew Van Duren.
2. Securing the Internet of Things by Li Da Xu, and Shancang Li (Syngress).
3. IoT Security Issues by Alasdair Gilchrist (De Gruyter).
4. The Internet of Risky Things by Sean Smith (O'Reilly).

List of Experiments (Hardware on Breadboard / Software using NI Multisim)

1. Encryption and Decryption using Python (AES/RSA algorithms).
   Software: Python libraries (PyCrypto, Cryptography).
   Hardware: Raspberry Pi.
2. Configuring HTTPS for IoT Devices using NodeMCU.
   Software: Arduino IDE, OpenSSL.
3. Intrusion Detection Simulation using Wireshark and Snort.
   Software: Wireshark, Snort.
4. Secure Device Communication using TLS with MQTT.
   Hardware: NodeMCU.
5. Firmware Update Simulation for IoT device.
   Hardware: ESP32.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Intelligent Imaging (DSE – 3/ GE – 5)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Intelligent Imaging	4	3	0	1	Mathematics – I, Fundamentals of Computer Programming, Mathematics – II, Fundamentals of Image Processing, Image Filtering and Restoration

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Introduce advanced imaging techniques (e.g., hyperspectral, 3D imaging, light-field imaging) and their roles in intelligent systems.
- Apply state-of-the-art machine learning and deep learning models for image interpretation, object recognition, and scene understanding.
- Reinforce hands-on proficiency with industry-standard libraries (OpenCV, TensorFlow/PyTorch) and hardware (embedded vision systems, specialized sensors).
- Encourage students to experiment, prototype, and innovate intelligent imaging solutions for real-world problems (e.g., medical imaging, autonomous navigation, surveillance, robotics).
- Introduce concepts of fairness, privacy, and explainability in AI-driven imaging solutions.

Course Outcomes:
After completing the course, the students should be able to:
1. Explain advanced imaging modalities and select appropriate imaging techniques for a given application.
2. Implement and fine-tune state-of-the-art computer vision algorithms and deep learning models for intelligent imaging tasks.
3. Design and execute end-to-end intelligent imaging pipelines, from sensor data acquisition to interpretation and decision-making.
4. Analyze and evaluate system performance using quantitative metrics, improve models through iterative experimentation, and ensure robustness and reliability.
5. Apply ethical considerations and understand regulatory frameworks governing intelligent imaging solutions.

Unit - I
Advanced Imaging Techniques and Modalities: Recap of standard 2D imaging and introduction to advanced modalities. Hyperspectral & Multispectral Imaging: Concepts, sensors, data cubes, and applications (agriculture, remote sensing). Light-field Imaging: Principles, plenoptic cameras, and depth estimation. 3D Imaging: Structured light, Time-of-Flight sensors, LIDAR, and stereo vision. Computational Photography: High Dynamic Range (HDR), panoramic stitching, synthetic aperture, and refocusing.

Unit - II

Intelligent Image Representation and Feature: Feature Representation: SIFT, SURF, ORB, and advanced local descriptors. Introduction to Deep Feature Extractors: Convolutional Neural Networks (CNNs), transfer learning, and feature maps. Dimensionality Reduction & Manifold Learning: PCA, t-SNE, UMAP for image embedding. Sparse and Dense Representations: Autoencoders, Variational Autoencoders (VAE) for image compression and generation.

Unit - III

Intelligent Image Interpretation via Machine Learning: Deep Learning Architectures for Vision: CNNs, ResNets, EfficientNets, Vision Transformers. Object Detection & Recognition: YOLO, Faster R-CNN, Mask R-CNN, DETR. Semantic and Instance Segmentation: U-Net, DeepLab, Segment-Anything Model. Explainability and Fairness in Vision: Grad-CAM, LIME, bias in datasets. Domain Adaptation and Transfer Learning: Fine-tuning pre-trained models for new tasks.

Unit - IV

Intelligent Imaging Systems and Applications: Embedded and Real-Time Vision Systems: Edge computing, NVIDIA Jetson. Industrial and Medical Applications: robotics navigation, autonomous vehicles, medical diagnostics. Performance Evaluation & System Optimization: Accuracy metrics, speed benchmarks, memory optimization, and model quantization.

Suggested Readings

1. Computer Vision: Algorithms and Applications by Richard Szeliski (Springer).
2. Digital Image Processing by Rafael C. Gonzalez and Richard E. Woods (Pearson).
3. Digital Image Processing by Bernd Jähne (Springer).

List of Experiments (Hardware on Breadboard / Software using NI Multisim)

1. Familiarization with OpenCV scripts
   Tasks: Run OpenCV scripts, explore TensorFlow
2. Advanced Imaging Modalities:
   Objective: Work with light-field and hyperspectral image samples.
   Tasks: Load hyperspectral datasets, perform spectral band selection, and reconstruct views from light-field data.
3. Computational Photography:
   Objective: Implement HDR merging and panoramic stitching.
   Tasks: Capture multiple exposure images using a camera module, merge into HDR; stitch overlapping images into a panorama.
4. Feature Extraction and Matching:
   Objective: Extract and match features for image registration or stitching.
   Tasks: Implement SIFT/SURF features, match keypoints between two images, visualize correspondences.
5. Deep Feature Representation:
   Objective: Compare handcrafted features with CNN-based features.
   Tasks: Extract features from a pre-trained CNN, use them for classification of a small dataset; compare accuracy with SIFT/SURF.
6. Object Detection and Recognition:
   Objective: Implement YOLO or Faster R-CNN on a given dataset.
   Tasks: Fine-tune a pre-trained object detector on a custom dataset (e.g., lab images of objects).

7. Segmentation Task:
   Objective: Implement semantic segmentation with U-Net or DeepLab.
   Tasks: Train a segmentation model on a subset of annotated images (e.g., medical or aerial images).
8. Explainable AI for Vision:
   Objective: Visualize model decisions.
   Tasks: Use Grad-CAM to highlight regions in an image that influence a CNN's classification.
9. Embedded Vision Implementation:
   Objective: Deploy a trained CNN model on an NVIDIA Jetson or Raspberry Pi.
   Tasks: Real-time object detection/inference with a camera feed, optimizing model size.
10. Capstone Project:
    Objective: Integrate multiple techniques into a cohesive intelligent imaging application.
    Tasks: Students work in teams to propose, design, and implement an end-to-end solution (e.g., a smart surveillance camera that detects intruders and segments foreground, or a spectral-based crop health analyzer).

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Department of Electronics and Communication Engineering
Faculty of Technology
University of Delhi

Detailed Syllabus of Generic Elective (GE) courses offered for Minors/ Specializations by the Department in Semester VI

Antenna and Wave Propagation (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Antenna and Wave Propagation	4	3	0	1	Physics, Mathematics – I, Electromagnetic Theory

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- Learn the basic parameters of an antenna and its radiation mechanism
- Understand the design and analyze of various wire antenna, and antenna arrays
- Understand the design and analyze of various broadband and planar antennas
- Learn the concept of aperture as well as high gain reflector antennas
- Express the basic concepts of ground, space, sky wave propagation mechanism

Course Outcomes:
After completing the course, the students should be able to:

1. Understand antenna characteristics for different applications.
2. Understand the concepts of Aperture and Slot Antennas
3. Analyze and design different types of antennas.
4. Design antenna arrays and understand operation of smart antennas.
5. Investigate different modes of propagation and their suitability for wireless communication.

Unit - I
Electromagnetic Radiation and Antenna Fundamentals- Review of Maxwell's equations: Retarded vector potential, Solution of wave equation in retarded case, Concept of radiation, Antenna equivalent circuits, Antenna characteristics: Radiation pattern, Beam solid angle, Radiation intensity, Directivity, Gain, Input impedance, Polarization, Bandwidth, Effective aperture, Antenna effective height, Antenna temperature.

Unit - II
Wire Antenna and Antenna Arrays-Wire antennas: Hertzian dipole, short dipole, Radiation resistance and Directivity, Half wave Dipole, Monopole, Small loop antennas. Antenna Arrays: Linear Array and Pattern Multiplication, Two-element Array, Uniform Array, Array with non-uniform Excitation, Binomial Array.

Unit - III

Special and Broad band Antennas-Special Antennas: Long wire, V and Rhombic Antenna, Yagi-Uda Antenna, Turnstile Antenna, Helical Antenna- Axial and Normal mode helix, Bi- conical Antenna, Frequency Independent Antenna, Log periodic Dipole Array, Spiral Antenna, Microstrip.
Aperture Antennas- Aperture Antennas: Slot antenna, Horn Antenna, Pyramidal Horn Antenna, Reflector Antenna- Flat reflector, Corner Reflector, Common curved reflector shapes, parabolic reflector, Lens Antenna, Patch Antennas.

Unit - IV

Radio Wave Propagation- Ground Wave Propagation, Free-space Propagation, Ground Reflection, Surface waves, Diffraction, Wave propagation in complex Environments, Tropospheric Propagation, Space waves, Ionosphere propagation: Structure of ionosphere, Skywaves, Skip distance, Virtual height, Critical frequency, MUF, Electrical properties of ionosphere, Effects of earth's magnetic fields, Faraday rotation.
Modern Antennas- Phase Array Antennas, Smart Antennas for Mobile Communication, MIMO Antennas for 5G Communication System, Reconfigurable Antenna

Suggested Readings

1. Antennas by J.D. Kraus (McGraw Hill).
2. Antenna Theory - Analysis and Design by C.A. Balanis (John Wiley).
3. Antennas and Radio Wave Propagation by R.E. Collin (McGraw Hill).
4. Antenna Engineering Handbook by R.C. Johnson and H. Jasik (McGraw Hill).
5. Electromagnetic Waves by R.K. Shevgaonkar (McGraw Hill).
6. Antenna Theory and Design by Stutzman, W.L. and Thiele, H.A (John Wiley & Sons).
7. Electromagnetic waves and Radiating Systems by E.C. Jordan and Balmain (Pearson Education).

<u>List of Experiments</u>

1. Electric Field Mapping: To map the electric field lines around different charge distributions and study the concept of electric flux density and Gauss's law.
2. Transmission Line Parameters: To measure the characteristic impedance and propagation constant of transmission lines and study the wave propagation in lossless and conducting media.
3. Electromagnetic Wave Propagation: To study the reflection and refraction of plane waves at the interface of different dielectric materials and measure the depth of penetration of electromagnetic waves.
4. Polarization of Electromagnetic Waves: To study the polarization states of electromagnetic waves and demonstrate linear, circular, and elliptical polarization.
5. Impedance measurement and Frequency measurement using rectangular waveguide.
6. Using simulation software
    a) Plot radiation pattern of dipole antenna
    b) Plot radiation pattern of monopole antenna
    c) Wire Antenna and Antenna Arrays-Wire antennas
    d) Reflector Antenna
    e) MIMO Antennas for 5G Communication System
    f) Reconfigurable Antenna

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Elements of Wireless Communication (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Elements of Wireless Communication	4	3	0	1	Mathematics – II, Signals and Systems

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- To introduce the fundamental concepts and techniques of wireless communication systems.
- To provide an understanding of wireless channel characteristics and modulation techniques.
- To familiarize students with wireless network protocols and standards.
- To equip students with hands-on experience in designing and analyzing wireless communication systems.

Course Outcomes:
After completing the course, the students should be able to:

1. Explain the basic principles of wireless communication systems and technologies.
2. Analyze wireless channel characteristics and their impact on communication performance.
3. Apply modern modulation and multiple access techniques for wireless systems.
4. Design and simulate wireless communication links and networks using software tools.
5. Understand the evolution of wireless standards and the basics of current and emerging technologies.

Unit - I
Introduction to Wireless Communication: Overview of wireless communication systems. Evolution of wireless technologies (1G to 6G, IoT). Basic structure of wireless communication systems: Transmitter, channel, receiver. Key performance metrics: Bandwidth, latency, throughput, and reliability. Introduction to frequency spectrum and channel allocation.

Unit - II
Wireless Channel Characteristics: Propagation models: Free space, two-ray, and multipath. Large-scale fading: Path loss and shadowing. Small-scale fading: Types, Doppler effect, and coherence time. Rayleigh and Rician Fading. Channel capacity and diversity techniques.

Unit - III
Modulation and Multiple Access Techniques: Review of Digital modulation techniques: shift keying and QAM. Spread spectrum techniques: DSSS and FHSS. OFDM. FEC Techniques: Convolutional and Turbo Codes. Multiple access techniques: FDMA, TDMA, CDMA, and OFDMA.

Unit - IV
Wireless Standards and Networks: Overview of wireless communication standards: GSM, CDMA2000, LTE, 5G. Basics of WLAN (Wi-Fi), Bluetooth, ZigBee, and LoRaWAN. Introduction to MIMO systems (Role in Wireless and Beamforming). Future trends in wireless communication: mmWave, IoT, and 6G.

Suggested Readings

1. Modern Digital and Analog Communication Systems by B.P. Lathi (Oxford).
2. Communication Systems by Simon Haykin (Wiley).
3. Digital Communications by John G. Proakis (McGraw-Hill).
4. Wireless Digital Communications by K. Feher (Prentice Hall).
5. Wireless Communications by Andrea Goldsmith (University Press).
6. Wireless Communications: Principles and Practice by Theodore S. Rappaport (Pearson).
7. Fundamentals of Wireless Communication by David Tse and Pramod Viswanath (Cambridge).
8. Modern Wireless Communications by Simon Haykin and Michael Moher (Pearson).
9. Wireless Communications and Networks by William Stallings (Pearson).

List of Experiments

1. Set up a basic wireless link using SDR and measure signal strength.
2. Analyze wireless channel allocation in a real environment using Wireshark.
3. Experiment with indoor and outdoor path loss measurements.
4. Simulate Rayleigh and Rician fading channels using MATLAB.
5. Simulate and compare OFDM with single-carrier modulation using MATLAB or Python.
6. Implement a basic FEC system using coding algorithms in MATLAB or Python.
7. Simulate a MIMO system using MATLAB and measure capacity gains.
8. Design a basic IoT communication system using LoRa or Zigbee modules.

Hardware Requirements:

1. Software-Defined Radio (SDR) kits (e.g., RTL-SDR, HackRF, or USRP).
2. Zigbee and LoRa modules for IoT experiments.
3. NodeMCU or Raspberry Pi for IoT applications.
4. Antennas for diversity and beamforming experiments.
5. Spectrum analyzers for propagation and channel measurements.

Software Requirements:

1. MATLAB/Simulink or GNU Octave for simulation.
2. GNURadio for SDR programming.
3. Python with libraries such as NumPy, SciPy, and Matplotlib.
4. Wireshark for network protocol analysis.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Satellite Communication (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Satellite Communication	4	3	0	1	Mathematics – I, Electromagnetic Theory

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- To provide in-depth knowledge of satellite communication principles, system design, and applications.
- To enhance problem-solving skills and foster innovation through hands-on learning in satellite communication systems.
- To familiarize students with advanced technologies and emerging trends in satellite networks.
- To prepare students for careers in telecommunications, aerospace, and related industries.

Course Outcomes:

After completing the course, the students should be able to:

1. Apply the principles of orbital mechanics and satellite subsystems in real-world satellite communication systems.
2. Design and analyze satellite links, including uplink, downlink, and link budgets, with a focus on performance optimization.
3. Evaluate multiple access techniques and advanced technologies in satellite networks.
4. Develop skills in designing, testing, and troubleshooting satellite communication systems using simulation tools and hardware.

Unit - I

Fundamentals of Satellite Communication and Orbital Mechanics Satellite types and applications. Orbital mechanics: Orbital elements, equations of motion, orbital perturbations. Satellite constellations and inter-satellite links. Look angle determination, limits of visibility, eclipse, and sun transit outage. Spacecraft technology: Structural design, power systems, attitude and orbit control, thermal management, propulsion, telemetry, tracking, command, and communication subsystems. Launch vehicles and procedures.

Unit - II

Satellite Link Design and Earth Station Technology Earth station architecture: Transmitters, receivers, and antenna systems. Basic transmission theory: Uplink and downlink design. Link budget analysis: Eb/No calculations, system noise, and intermodulation effects. Performance impairments: Propagation characteristics and reliability. Case studies: Design for IMMARSAT, INTELSAT, and other satellites.

Unit - III

Multiple Access Techniques and Optical Communication in Satellites FDMA, TDMA, CDMA: Concepts, system design, and comparison. Satellite switch TDMA, SPADE systems, and backoff considerations. Optical communication in satellite networks: Inter-satellite links, laser communication, beam acquisition, tracking, and pointing. Advanced topics: Quantum communication via satellites.

Unit - IV

Packet satellite networks and services, fixed satellite services, broadcast satellite services, mobile satellite services- VSAT, GPS, maritime satellite services, gateways, ATM over satellite, role of satellite in future network. Integration of satellites in next-gen networks: 5G/6G and IoT.

Suggested Readings

1. Satellite Communications by Timothy Pratt and Jeremy Allnutt (Wiley).
2. Satellite Communication by Dennis Roddy (McGraw Hill).
3. Satellite Communications by Varsha Agrawal, Anil K. Maini (Wiley).
4. Digital Satellite Communications by Tri T. Ha (Tata McGraw Hill).

List of Experiments

1. Understanding Orbital Mechanics
   Objective: Calculate orbital elements and simulate satellite orbits using MATLAB/Simulink.
2. Satellite Link Budget Analysis
   Objective: Design uplink and downlink with link budget calculations.
3. Antenna Design and Analysis for Earth Stations
   Objective: Design and test parabolic and horn antennas for satellite communication.
4. Modulation Techniques for Satellites
   Objective: Implement and test QPSK, BPSK, and QAM modulation schemes.
5. Simulation of FDMA, TDMA, and CDMA Systems
   Objective: Simulate and compare multiple access techniques.
   Requirements: MATLAB, NS3, or similar simulators.
6. Laser Communication System Simulation
   Objective: Design and analyze optical communication links for satellites.
   Requirements: Optical simulation tools (OptiSystem), photodetectors, laser diodes.
7. Satellite IoT Network Implementation
   Objective: Design a small satellite network for IoT applications using LoRaWAN.
   Requirements: LoRa modules,
8. AI in Satellite Communication
   Objective: Implement an AI-based fault detection system for satellite telemetry data.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Cognitive Radio & Networks (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Cognitive Radio & Networks	4	3	0	1	Signals and Systems, Wireless Communication and Mobile Networks

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Understand the principles of cognitive radio, including spectrum scarcity, dynamic spectrum access, and the cognitive cycle.
- Analyze spectrum sensing and management techniques and their application in dynamic spectrum sharing.
- Explore advanced concepts such as cross-layer design, OFDM, and MIMO systems in cognitive radio networks.
- Gain practical skills through experiments and understand the regulatory frameworks and emerging technologies in cognitive radio.

Course Outcomes:
After completing the course, the students should be able to:
1. Demonstrate an understanding of cognitive radio fundamentals, including its architecture and functionalities.
2. Apply spectrum sensing and sharing techniques to design efficient cognitive radio systems.
3. Analyze regulatory standards and integrate advanced technologies like OFDM and UWB into cognitive radio networks.
4. Develop practical skills for implementing and evaluating cognitive radio systems using hardware and software tools.

Unit - I
Introduction to Cognitive Radio and Networks: spectrum scarcity and Spectrum white space, Fixed spectrum allocation, Software Defined Radio (SDR): Concept, Limitations, Evolution to Cognitive Radio. Dynamic Spectrum Access and Cognitive Cycle, Functions of Cognitive Radio: Spectrum Sensing, Spectrum Management, Spectrum Mobility, Cognitive Radio Architecture.

Unit - II
Spectrum Sensing and Management: Hypothesis Model for Spectrum Sensing. Types of Spectrum Sensing: Non-cooperative Sensing, Cooperative Sensing, Interference-based Sensing. Detection Techniques: Matched Filter Detection, Energy Detection, Cyclostationary Feature Detection. Dynamic Spectrum Access (DSA): Models and Architectures, Opportunistic Spectrum Access (OSA).

Unit - III
Cognitive Radio Networks (CRN): Cognitive Radio Cross-Layer Design: Adaptation and Optimization Security Challenges in Cognitive Radio Networks, MIMO Systems, Smart Antennas, and Beamforming in Cognitive Radio, OFDM-based Cognitive Radio: suitability, challenges, and Multiband OFDM. Standards and Technologies for Cognitive-OFDM.

Unit - IV

Standardization, Regulation, and Emerging Technologies: Regulatory Issues and New Spectrum Management Regimes. Spectrum Planning and Authorization. Standards. Ultra-Wideband (UWB) Cognitive Radio: Fundamentals of Impulse Radio UWB. Cognitive Radio Requirements vs. IR-UWB. Merging Impulse Radio with Cognitive Radio.

Suggested Readings

1. Dynamic Spectrum Access and Spectrum Management in Cognitive Radio Networks by Ekram Hossain, Dusit Niyato, and Zhu Han, 1st Edition (Cambridge University Press, 2009)
2. Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems by Huseyin Arslan, 1st Edition (Springer, 2007)
3. Cognitive Radio Technology by Bruce A. Fette, 2nd Edition (Academic Press, 2009)
4. Suggested Reference Books:
5. Cognitive Radio Networks by Yang Xiao and Fei Hu, 1st Edition (CRC Press, 2008)
6. Essentials of Cognitive Radio by Linda E. Doyle, 1st Edition (Cambridge University Press, 2009)
7. Cognitive Radio Communications and Networks by Alexander M. Wyglinski, Maziar Nekovee, and Thomas Hou (Academic Press)
8. Wireless Communications: Principles and Practice by Theodore S. Rappaport (Pearson)
9. Principles of Modern Wireless Communication Systems by Aditya K. Jagannatham (McGraw Hill)
10. Other Useful Resources:
11. GIAN IIT Kanpur course contents

List of Experiments

1. Basic SDR Implementation Using GNU Radio and USRP
   Objective: Demonstrate dynamic spectrum access and identify available frequency bands using SDR hardware.
   Requirements: GNU Radio software, USRP hardware, antennas, signal source.
2. Spectrum Sensing Using GNU Radio
   Objective: Implement and compare energy detection and cyclostationary feature detection.
   Requirements: GNU Radio, USRP, spectrum analyzer.
3. OFDM-Based Cognitive Radio Simulation
   Objective: Model an OFDM-based cognitive radio system and study the impact of spectrum fragmentation and interference on system performance.
4. Cooperative Spectrum Sensing
   Objective: Evaluate performance improvements with distributed sensing.
   Requirements: Multiple USRP devices, GNU Radio.
5. Dynamic Spectrum Access
   Objective: Simulate dynamic spectrum access and interference management.
   Requirements: NS-3, MATLAB.
6. MAC Protocol Simulation
   Objective: Design and analyze a spectrum-aware MAC protocol.
   Requirements: NS-3, Python.
7. QoS Optimization
   Objective: Implement algorithms to enhance QoS in CRNs.
   Requirements: Python, MATLAB.
8. 5G Cognitive Radio Simulation
   Objective: Simulate the integration of cognitive radio in 5G networks.
   Requirements: MATLAB, NS-3.

9. Real-Time Communication System
   Objective: Prototype a small-scale cognitive radio communication system.
   Requirements: USRP, GNU Radio.
10. Energy Efficiency Analysis
    Objective: Measure and optimize energy consumption in cognitive radios.
    Requirements: MATLAB, power meters.
11. Smart Antenna and Beamforming Experiment
    Objective: Design and test a basic MIMO system with smart antennas for efficient spectrum sharing.
    Requirements: SDR platform, antenna arrays, MATLAB/Python for algorithm testing.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

CMOS Analog IC Design (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
CMOS Analog IC Design	4	3	0	1	Physics, Electronic Devices and Circuits, Digital Electronics – I, Linear Integrated Circuits

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- To understand the principles and building blocks of CMOS analog ICs.
- To develop skills in analyzing and designing CMOS amplifiers and op-amps.
- To explore advanced CMOS architectures and nonlinearity cancellation techniques.
- To design and implement MOSFET-C circuits for filters, oscillators, and multipliers.
- To apply switched-capacitor design principles in practical applications.
- To gain hands-on experience with simulation tools and circuit optimization.
- To foster innovative problem-solving in analog IC design.

Course Outcomes:
After completing the course, the students should be able to:

1. Demonstrate a thorough understanding of CMOS device characteristics and their relevance in analog integrated circuits.
2. Analyze and design CMOS analog building blocks such as current mirrors, amplifiers, and op-amps, including their frequency response and large/small signal behavior.
3. Design and analyze CMOS operational amplifiers (op-amps) and operational transconductance amplifiers (OTAs) with specific emphasis on performance metrics like gain bandwidth, slew rate, and impedance.
4. Apply MOSFET-C circuit design techniques to create filters, oscillators, and multipliers with linear and nonlinear functionalities.
5. Design and simulate switched-capacitor circuits including universal filters, VCOs, and integrators, with a focus on practical implementation challenges.
6. Utilize modern simulation tools to evaluate and optimize the performance of CMOS analog circuit designs.

Unit - I
Basic building blocks of CMOS analog ICs: Review of PMOS, NMOS and CMOS devices, characteristics and circuit models; Importance of CMOS analog circuits; diode-connected MOSFET; Current sources and current mirrors (basic, Wilson, Cascode and others); NMOS and PMOS differential amplifiers with active loads and their large signal analysis; small signal analysis; determination of Ad, Ac and CMRR; CS amplifier as a gain stage; Cascode amplifier and its frequency response analysis; Differential to single-ended converter; Source follower as output stage and level shifter; the IC op-amp architecture.

Unit - II
CMOS op-amps and OTAs: Various CMOS op-amp architectures; Two-stage CMOS op-amp design; large signal and small signal analysis; frequency response analysis-determination of gain bandwidth product, slew rate, Zin and Zout; Basic CMOS OTAs; Non-linearity cancellation techniques; Design of CMOS linear transconductors.

Unit - III
Fully-integrable MOSFET-C Circuits: The basic topologies of nonlinearity cancellation in MOSFET circuits and their analysis; fully differential continuous-time MOSFET-C filters based on CMOS op-amps; MOSFET-C filters design using CMOS Current Conveyors and Current feedback op-amps; MOSFET-C oscillators; CMOS analog multipliers and other linear/nonlinear functional circuits.

Unit - IV
MOS Switched-capacitor networks: The switched-capacitor resistor equivalence; Switched- capacitor integrator; generation of two-phase non-overlapping clock; basic building blocks of switched-capacitor networks; Switched-Capacitor universal biquad filter design; Switched-capacitor filter design using generalised impedance converter (GIC) and bilinear transform; design of the switched-capacitor linear VCOs.

Suggested Readings

1. Design of Analog CMOS Integrated Circuits by Razavi (McGraw-Hill Education).
2. CMOS Analog Circuit Design by Allen and Holberg (Oxford University Press).
3. Analysis and Design of Analog Integrated Circuits by Gray and Meyer (John Wiley & Sons).
4. Analog Filter Design by Schoumann and Van Valkenburg (Oxford University Press).

<u>List of Experiments</u>

1. Evaluation of the performance of various MOS current mirrors
2. SPICE simulation and studies of (i) differential to singe ended converter and (ii) level shifter
3. Frequency response analysis of CS amplifier and cascade amplifier
4. Evaluation of the performance of a two-stage CMOS op-amp
5. Simulation of a CMOS linear transconductor
6. Design of a second order switched-capacitor filter
7. Design of a Switched-capacitor VCO
8. MOSFET-C biquad filter design and simulation
9. Design and simulation of a MOSFET-C VCO
10. Simulation of the CMOS Gilbert Multiplier

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Current Mode Analog VLSI Circuits (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Current Mode Analog VLSI Circuits	4	3	0	1	Physics, Electronic Devices and Circuits, Linear Integrated Circuits

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- Develop a strong theoretical foundation in the principles and advantages of current-mode circuit design compared to traditional voltage-mode circuits.
- Enable students to design, analyze, and implement various current-mode circuits, such as multipliers, dividers, squarers, and other translinear circuits.
- Study and apply various current-mode building blocks like Current Conveyors (CCs) and Current Feedback Op-Amps (CFOAs).
- Equip students with the ability to utilize CMOS technologies for implementing current-mode circuits, addressing challenges like nonlinearity and achieving high performance.
- Explore the concepts and practical implementations of fully integrable current-mode filters and dynamic translinear circuits.

Course Outcomes:
After completing the course, the students should be able to:
1. Evaluate the advantages and limitations of current-mode circuits over traditional voltage-mode circuits.
2. Apply the translinear principle to design and synthesize circuits like analog multipliers, dividers, and square-root circuits.
3. Implement and analyze current conveyors and their applications in practical analog designs.
4. Analyze and design CMOS-based current-mode circuits with nonlinearity cancellation and explore applications in analog signal processing.
5. Design and synthesize fully integrable current-mode filters, including log-domain and square-root domain filters.
6. Use current-mode design principles to tackle complex challenges in analog VLSI design, contributing to advanced IC development.

Unit - I
Voltage-mode versus Current-mode circuits; advantages of current mode circuit design; Current-mode circuits from voltage-mode op-amps using supply-current sensing techniques; the operational mirrored amplifiers (OMA) and their applications; The Translinear principle and circuits; static translinear circuit design- analog multipliers, analog dividers, squarers, square-rooters; geometric mean and harmonic mean circuits; the compete translinear squarer design; Synthesis of a 4-quadrant current-mode analog multiplier.

Unit - II
Current-mode analog building blocks: The current Conveyors and their variants; Current-feedback op-amp; the mixed translinear cell; the translinear implementation of CCs and CFOAs; the gain bandwidth conflict resolution and attainment of high slew rates; the basic analog circuits realisable from CCs and CFOAs; Examples of IC CCs and CFOAs; Miscellaneous current-mode building blocks.

Unit - III

CMOS current-mode circuits: Review of MOS Current mirrors; the translinear principle for MOS circuits; CMOS mixed translinear cell; CMOS implementation of CCs and CFOAs; Nonlinearity cancellation in MOS analog circuits and their applications; CMOS Voltage-controlled linear resistors; CMOS nonlinear functional circuits.

Unit - IV

Fully- integratable Current-mode filters: The dynamic translinear circuits; Log domain and translinear filters; MOS geometric mean circuits, squarers and squarer/divider circuits; Introduction to fully-integratable square-root domain CMOS analog filters.

Suggested Readings

1. Analog IC Design: The Current-Mode Approach by Toumazou, Lidgey, and Haigh (Institution of Engineering and Technology, 1990).
2. Integrated Circuits for Analog Signal Processing by Esteban Tlelo-Cuautle (Springer, 2009).
1. Analog Integrated Circuit Design by Tony Chan Carusone, David Johns, and Kenneth Martin (Wiley, 2011).
2. Design of Analog CMOS Integrated Circuits by Razavi (McGraw-Hill Education, 2017).
3. Analog VLSI: Circuits and Principles by Shih-Chii Liu, Jörg Kramer, Giacomo Indiveri, Tobias Delbrück, and Rodney Douglas (MIT Press, 2002).

List of Experiments

1. Realisation of Current followers using operational-mirrored amplifier
2. Realisation of constant-bandwidth, variable gain amplifiers using AD844 CFOA
3. Design of a Translinear Squarer and verification of its operation
4. Realisation of a CMOS linear VCRs
5. CMOS Gilbert multiplier and determination of its performance
6. Current-mode analog biquad filter design using AD844 CFOA
7. Design and verification of an instrumentation amplifier using CC/CFOA
8. Design of a single resistance controlled oscillator using CFOAs
9. Design of a log domain low pass filter
10. Design of a CMOS square-root domain all pass filter

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Ubiquitous Sensing, Computing and Communication (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Ubiquitous Sensing, Computing and Communication	4	3	0	1	Fundamentals of Computer Programming

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To provide an in-depth understanding of ubiquitous sensing, computing, and communication in IoT systems.
- To explore advanced topics such as context-aware computing, location-based services, and mobile applications.
- To understand the integration of IoT with data analytics and real-time processing for actionable insights.

Course Outcomes:
After completing the course, the students should be able to:
1. Demonstrate knowledge of ubiquitous sensing and computing frameworks.
2. Design and develop context-aware and location-aware IoT systems.
3. Analyze IoT data using modern analytics tools for decision-making and problem-solving.

Unit - I
Fundamentals of Ubiquitous IoT Systems: Introduction to IoT and Ubiquitous Computing. Overview of IoT Networking Basics: NFC, Wireless LAN, LPWAN. Challenges in Ubiquitous IoT Systems. Fundamentals of Location-aware Computing: Concepts, Architecture, and Applications. Location-based Services (LBS): Social Networks, Recommendations, and Real-World Applications.

Unit - II
Context-Aware and Mobile Ubiquitous Computing: Context and Context-Aware Computing: Definitions and Challenges. Developing Context-Aware Applications: System Architecture and Implementation. Wearable Computing: Devices and Applications (Smart Glasses, Augmented Reality, Digital Pen). Mobile Social Networking and Crowd Sensing: Case Studies. Event-Based Social Networks and Applications.

Unit - III
Privacy, Security, and Energy in Ubiquitous IoT: Privacy and Security Challenges in Ubiquitous IoT Systems. Energy Constraints in IoT and Strategies for Efficiency. Human Activity and Emotion Sensing for Mobile Applications. Smart Homes and Intelligent Buildings: Design and Applications. Mobile Peer-to-Peer Computing and Cloud-Centric IoT.

Unit - IV
IoT Data Management and Analytics: IoT Data Management: Cleaning, Processing, and Storage Models. Advanced Search Techniques: Semantic Sensor Web, Deep Web, and Semantic Web Data Management. Real-Time and Big Data Analytics for IoT: High-Dimensional and Heterogeneous Data Processing. Parallel and Distributed Data Processing: Techniques and Tools. QoS and QoE in IoT: Concepts and Protocols.

Suggested Readings

1. Ubiquitous Computing Fundamentals by John Krumm (CRC Press).
2. Internet of Things: Principles and Paradigms by Rajkumar Buyya and Amir Vahid Dastjerdi (Morgan Kaufmann).
3. Real-Time Analytics by Byron Ellis (Wiley)
4. Ubiquitous Computing and Computing Security of IoT by N. Jeyanthi, Ajith Abraham, and Hamid Mcheick (Springer).
5. Enterprise IoT by Dirk Slama (O'Reilly Publisher).

List of Experiments

1. Interfacing DHT11 and soil moisture sensor with Arduino/ESP32.
   Hardware: Arduino/ESP32, Sensors.
2. Uploading sensor data to ThingSpeak and creating real-time dashboards.
   Software: ThingSpeak, Python.
3. Deploying a simple ML model on Raspberry Pi for anomaly detection.
4. Setting up LoRa modules for data transmission.
5. AIoT Application: Using Google Colab for predictive analytics with IoT data.
6. NFC Communication: Implementing a basic NFC-based interaction between devices.
7. Location-Aware Application: Developing a GPS-based location tracking system for personal assistants.
   Hardware: Raspberry Pi, GPS module.
8. Smart Home System: Developing a prototype for smart home automation using NodeMCU and MQTT.
   Hardware: NodeMCU, sensors (temperature, motion), actuators (relay).

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Introduction to Embedded Systems for IoT (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	**Tutorial**	**Practical**	
Introduction to Embedded Systems for IoT	4	3	0	1	Fundamentals of Computer Programming

Course Hours: L – 03, T – 00, P - 02

Course Objectives:
- To understand the purpose, architecture, and design principles of embedded systems in IoT.
- To develop proficiency in interfacing sensors, actuators, and communication modules with embedded platforms.
- To explore IoT-enabling technologies and integrate them with cloud platforms for real-world applications.

Course Outcomes:
After completing the course, the students should be able to:
1. Design embedded systems tailored for IoT applications.
2. Interface IoT devices with sensors, actuators, and other hardware components.
3. Implement communication protocols and cloud integration for IoT solutions.

Unit - I
Fundamentals of Embedded IoT Systems: Introduction to Embedded IoT: Purpose and Requirement Specification. IoT Level Specifications: Functional View, Operational View. Device and Component Integration: Overview and Challenges. Pillars of Embedded IoT: The Internet of Devices and IoT Levels.

Unit - II
Design of Embedded Systems: Core Components: Common Sensors and Actuators. Embedded Processors and Microcontroller Architectures. Memory Architectures: Flash, EEPROM, and RAM. Software Architectures: RTOS vs Bare-Metal Programming.

Unit - III
Inputs and Outputs in Embedded Systems: Digital Inputs/Outputs: GPIO, BusIn, BusOut, BusInOut. Analog Inputs/Outputs: ADC and DAC Interfacing. Pulse Width Modulation (PWM) for Motor and Light Control. Advanced Peripherals: Accelerometer, Magnetometer, SD Card Integration, and File Systems. Sensor and Actuator Integration: Temperature, Humidity, Motion Sensors, and Relays.

Unit - IV
IoT-Enabling Technologies: Communication Technologies: RFID, NFC, Bluetooth Low Energy (BLE), LiFi, ZigBee, Z-Wave, LoRa, and 6LoWPAN. IoT Protocols: HTTP, WebSocket, MQTT, CoAP, XMPP. IoT Platforms: Node-RED, IBM Watson IoT, AWS IoT, Microsoft Azure IoT, Google Cloud IoT, ThingWorx.

Suggested Readings

1. Embedded Systems: Architecture, Programming, and Design by Raj Kamal (McGraw-Hill Education).
2. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things by David Hanes et al. (Cisco Press).
3. Exploring Arduino: Tools and Techniques for Engineering Wizardry by Jeremy Blum (Wiley).
4. Mastering Internet of Things by Peter Waher (Packt Publishing).
5. Internet of Things: Principles and Paradigms by Rajkumar Buyya and Amir Vahid Dastjerdi (Elsevier).

List of Experiments

1. Digital I/O: Blinking LEDs and reading button input using ESP32.
2. Analog Interfacing: Reading sensor data (e.g., potentiometer) and controlling LEDs with PWM.
3. Sensor Integration: Interfacing temperature and motion sensors to display readings on an LCD.
4. IoT Protocol Implementation: Sending data to a cloud platform (ThingSpeak) using MQTT.
5. Cloud Integration: Building a weather monitoring system with live data visualization on AWS IoT.
6. Communication Technologies: Setting up BLE communication between ESP32 and a smartphone.
7. File Systems: Implementing data logging on an SD card and reading the logged data.
8. Smart Home Prototype: Developing a web interface to control home appliances using ESP32.
9. Node-RED: Creating a flow-based application for monitoring IoT device data.
10. Energy Efficiency Analysis: Measuring power consumption of IoT devices under different workloads.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Deep Learning for Image Analysis (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	Credits	Credit Distribution of the Course			Prerequisite of the course (if any)
		Lecture	Tutorial	Practical	
Deep Learning for Image Analysis	4	3	0	1	Fundamentals of Computer Programming, Mathematics – II, Intelligent Imaging

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- To provide in-depth knowledge of advanced deep learning architectures and their application to complex image analysis tasks.
- To enable students to understand and implement state-of-the-art object detection, segmentation, and generative models.
- To introduce vision transformers, self-supervised learning, and other frontier topics in deep learning for images.
- To equip students with practical skills for model training, optimization, deployment, and interpretation in real-world scenarios.

Course Outcomes:

After completing the course, the students should be able to:

1. Understand and critically evaluate advanced deep neural network architectures for image-related tasks.
2. Implement and fine-tune cutting-edge models for object detection, segmentation, and generative image synthesis.
3. Explore and apply vision transformers and self-supervised techniques to improve model performance and reduce labeling costs.
4. Employ interpretability methods, perform model compression, and deploy models on resource-constrained devices.
5. Design and conduct experiments to address real-world image analysis challenges, iterating through data preparation, model training, validation, and optimization.

Unit - I

Advanced CNN Architectures: Revisiting ResNet, DenseNet, MobileNet, EfficientNet. Training Optimization Techniques. Regularization & Generalization: DropBlock, label smoothing, and large-batch training strategies. Hyperparameter Tuning & Transfer Learning. Distributed & Parallel Training: Introduction to multi-GPU training, mixed precision, and cloud-based training pipelines.

Unit - II

Object Detection Models: Faster R-CNN, YOLO families, RetinaNet - architecture insights and training approaches. Image Segmentation: U-Net, Mask R-CNN, and newer variants for medical and satellite imaging. Vision Transformers (ViT): Fundamentals of transformer architecture applied to images, training and fine-tuning ViTs, comparison with CNNs. Advanced Topics in Detection & Segmentation: Handling imbalanced datasets, small object detection, and domain adaptation.

Unit - III

Generative Adversarial Networks (GANs): StyleGAN, CycleGAN, and conditional GAN variants for image generation and domain translation. Variational Autoencoders (VAEs): Concept and applications in image compression and feature disentanglement. Self-Supervised & Semi-Supervised Learning: SimCLR, MoCo, BYOL, DINO - methods to leverage unlabeled data, pretext tasks, and contrastive learning.

Unit - IV

Model Interpretability & Explainability: Grad-CAM, Guided Backprop, SHAP, LIME for understanding decision-making processes. Model Compression & Acceleration. Edge & Embedded Vision: Implementing lightweight models on GPUs, TPUs, and microcontrollers (e.g., NVIDIA Jetson, Google Coral).

Suggested Readings

1. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (MIT Press).
2. Deep Learning with Python by François Chollet (Manning).
3. CS231n Notes by Justin Johnson and Fei-Fei Li (Online).
4. Deep Residual Learning for Image Recognition by He et al. (ResNet).
5. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale by Dosovitskiy et al. (ViT).
6. U-Net: Convolutional Networks for Biomedical Image Segmentation by Ronneberger et al.
7. Generative Adversarial Networks by Goodfellow et al. (Original GAN Paper).
8. A Simple Framework for Contrastive Learning of Visual Representations by Chen et al. (SimCLR).

<u>List of Experiments</u>

1. CNN Fine-Tuning
   Objective: Fine-tune a pre-trained EfficientNet on a custom dataset.
      a. Implement data augmentation strategies (mixup, cutmix)
      b. Optimize hyperparameters using grid search or Bayesian optimization
2. Object Detection with YOLOv5
   Objective: Implement and train YOLO on a small custom dataset (e.g., campus surveillance images).
      a. Annotate a small dataset using labeling tools
      b. Train YOLOv5 and evaluate precision-recall, mAP
      c. Attempt domain adaptation (e.g., day vs. night images)
3. Image Segmentation using U-Net
   Objective: Segment objects in microscopy or satellite imagery.
      a. Train U-Net on a medical imaging dataset (e.g., cell nuclei segmentation)
      b. Evaluate with IoU, Dice scores
      c. Experiment with augmentations to handle class imbalance
4. Vision Transformers
   Objective: Fine-tune a ViT model on a small dataset.
      a. Use Hugging Face Transformers library
      b. Compare ViT performance vs. a standard CNN on the same dataset
5. GAN-based Image Generation
   Objective: Train a DCGAN or StyleGAN to generate synthetic images (e.g., anime faces, fashion items).
      a. Implement GAN training loop
      b. Explore latent space manipulations

6. Self-Supervised Learning
   Objective: Implement SimCLR to learn image embeddings without labels.
      a. Pre-train model using contrastive learning on unlabeled dataset
      b. Fine-tune on a small labeled subset
7. Model Interpretability
   Objective: Use Grad-CAM to visualize which regions of an image influence model predictions.
      a. Implement Grad-CAM on a trained model
      b. Analyze differences between classes and confirm correctness of reasoning
8. Model Compression and Deployment
   Objective: Compress and run a trained model on a Jetson Nano or Raspberry Pi.
      a. Apply pruning or quantization
      b. Deploy model and compare inference speed before and after compression

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)

Deep & Reinforcement Learning (DSE – 4/ GE – 6)
(Credit Distribution and Prerequisites of the Course)

Course Title	**Credits**	**Credit Distribution of the Course**			**Prerequisite of the course (if any)**
		Lecture	**Tutorial**	**Practical**	
Deep & Reinforcement Learning	4	3	0	1	Fundamentals of Computer Programming, Mathematics – II, Intelligent Imaging

Course Hours: L – 03, T – 00, P - 02

Course Objectives:

- Introduce the foundational concepts of Reinforcement Learning (RL), including Markov Decision Processes and basic RL methods.
- Demonstrate how deep learning can enhance RL, enabling agents to handle high-dimensional state spaces.
- Cover essential deep RL algorithms like DQN, policy gradients, and PPO, making students comfortable in both theory and coding.
- Expose students to practical considerations such as stability, exploration strategies, and environment setup.
- Allow learners to experiment with state-of-the-art tools and environments in a step-by-step manner, culminating in a final project.

Course Outcomes:
After completing the course, the students should be able to:

1. Understand core RL principles (policy, value function, reward) and how they map to real-world decision-making tasks.
2. Implement fundamental deep RL algorithms, from DQN to PPO, using modern frameworks.
3. Appreciate the challenges in training stable and efficient RL agents and employ techniques to address them.
4. Experiment with continuous action spaces and policy optimization methods.
5. Develop and present a small RL-based project, demonstrating the ability to apply learned techniques to a new problem.

Unit - I
Foundations of Reinforcement Learning: Basic Concepts: Agents, environments, states, actions, rewards, and returns. Markov Decision Processes (MDPs): Defining the RL problem, Bellman equations. Value-based Methods (Tabular): State-value and action-value functions, Q-learning basics. Limitations of Tabular Methods: Why we need function approximation.

Unit - II
Introducing Deep Reinforcement Learning: Neural Network Function Approximators: Representing value functions with neural nets. Deep Q-Network (DQN): How DQN works, replay buffers, target networks. Improving DQN: Double DQN, Dueling Networks for stability and better performance.

Unit - III

Policy-Based and Actor-Critic Methods: Policy Gradients (REINFORCE): Learning policies directly, basics of gradient-based policy optimization. Actor-Critic Methods: Advantage Actor-Critic (A2C), motivation and benefits. Proximal Policy Optimization (PPO): Conceptual understanding of PPO, clipping to maintain stable updates. Continuous Action Spaces: Brief introduction to handling continuous controls.

Unit - IV

Exploration Strategies: Epsilon-greedy, entropy bonuses, curiosity-driven exploration. Stability and Debugging: Identifying training instability, tuning hyperparameters. Basic Model-Based and Multi-Agent Concepts (Introductory): Light overview of model-based RL and the idea of multi-agent settings (no deep dive). Ethical and Safety Considerations: Brief discussion on when and how RL can impact real-world scenarios.

Suggested Readings

1. Reinforcement Learning: An Introduction by Sutton and Barto (MIT Press).
2. Spinning Up in Deep RL by OpenAI (Documentation and Tutorials).
3. Playing Atari with Deep Reinforcement Learning by Mnih et al. (DQN).
4. Proximal Policy Optimization Algorithms by Schulman et al. (PPO).

List of Experiments

1. Tabular Q-Learning
   Objective: Implement Q-learning to navigate a simple gridworld environment.
2. Deep Q-Networks (DQN) for CartPole
   Objective: Train a neural network-based agent to balance a pole on a cart using Deep Q-Networks (DQN).
3. Enhancing DQN
   Objective: Apply Double DQN or Dueling DQN to improve learning stability.
4. Hyperparameter Tuning
   Objective: Explore the effect of changing learning rates, batch sizes, and gamma values on agent performance.
5. Cliff-Walking Problem
   Objective: Demonstrate the concept of state-value functions using a cliff-walking example.
6. Design a simple neural network to predict action probabilities.

(Note: Course instructor may add/delete/update new experiments in addition to the above suggested practical exercises.)