Scheme For

Fifth Semester : Computer Engineering

S. No.	Course Code	Course Title	Hours Per Week			Internal Marks	External Marks (3 Hours)	Credits
			L	T	P			
1	23COPC301	Object Oriented Programming in Java	3	1	0	50	100	4
2		Program Elective - 1	3	0	0	50	100	3
	23COPE303	Computer Graphics						
	23COPE305	Free & Open Source Software (FOSS)						
3		Program Elective - 2	3	0	0	50	100	3
	23COPE307	Cyber Security & Cyber Law						
	23COPE309	Multimedia Technologies						
4		Open Elective - I	3	0	0	50	100	3
	23COOE311	Fundamentals of AI						
	23COOE313	Fundamentals of Machine Learning						
5	23COPC351	Object Oriented Programming in Java Lab	0	0	4	50	100	2
6		Lab Based on Program Elective-1	0	0	2	50	100	1
	23COPE353	Computer Graphics (Lab)						
	23COPE355	Free & Open Source Software (FOSS) (Lab)						
7		Open Elective - I (Lab)	0	0	4	50	100	2
	23COOE357	Fundamentals of AI						
	23COOE359	Fundamentals of Machine Learning						

8	23COSI361	*Summer Internship-II	0	3	0	50	50	3
		Total				400	750	21

*Four Week Summer Internship Training after Fourth Semester will be evaluated on the basis of Daily Diary/Training Report/PPT Presentation

As per AICTE Guidelines:

Teachers should use the following strategies to achieve the various outcomes of the course.

- Different methods of teaching and media to be used to attain classroom attention.
- Massive open online courses (MOOCs) may be used to teach various topics/subtopics.
- 15-20% of the topics which are relatively simpler or descriptive in nature should be given to the students for self-learning and assess the development of competency through classroom presentations.
- Case Studies/Micro-projects may be given to group of students for hand-on experiences

However, Students will be required to appear for the exams STRICTLY as per the scheme of exam as proposed by Board of Technical Education, GNCT of Delhi.

FIFTH SEMESTER

Course Code	23COPC301 (Theory) / 23COPC351 (Lab)
Course Title	Object Oriented Programming in Java
Credits	3 (L:3, T:1, P:2) (3 Hours Theory, 1 Hour Tutorial, 4 Hours Lab)
Semester	Fifth Semester
Course Category	Program Core

Course Objective:

- Understand the fundamental concepts of object-oriented programming and how they are implemented in Java.
- Learn to design and develop modular Java applications using classes, objects, inheritance, polymorphism, and abstraction.
- Apply exception handling and file I/O techniques to build robust and reliable Java programs.
- Explore and utilize Java's built-in libraries, packages, and the Collections Framework for effective data management.
- Gain introductory knowledge of GUI development and event handling using AWT and Swing components.

Unit 1	**Introduction to Java and Programming Fundamentals** History and features of Java, Java architecture: JDK, JRE, JVM, Structure of a Java program, compiling and execution process, Java tokens, identifiers, keywords, data types, Variables, constants, type casting, Operators and expressions, Control flow statements: if-else, switch, loops (for, while, do-while)	6
Unit 2	**Classes, Objects, and Constructors** Defining classes and creating objects, Fields, methods, and access modifiers (public, private, protected, default), Constructor types: default, parameterized, copy constructor, Method overloading, this keyword, Static vs. instance members, Initialization blocks (static and instance), Garbage collection and finalize(), Use of final keyword	7
Unit 3	**Arrays, Strings, and Wrapper Classes** One-dimensional and multi-dimensional arrays, Array of objects, String handling using String, StringBuffer, StringBuilder, String immutability and memory management, Wrapper classes and autoboxing/unboxing	7
Unit 4	**Inheritance, Polymorphism, and Abstraction** Inheritance: single, multilevel, hierarchical, Use of super keyword, Method overriding and dynamic method dispatch, Compile-time and run-time polymorphism, Object type casting and instanceof operator, Abstract classes and methods, Interfaces and multiple inheritance via interfaces, Introduction to lambda expressions and functional interfaces	7
Unit 5	**Exception Handling, Packages, and File I/O** Exception types: checked and unchecked, Exception handling using try-catch-finally, throw and throws keywords, Creating custom exceptions, Introduction to packages: built-in and user-defined, Import statements and access control, Basics of file handling using File, FileReader, FileWriter, Serialization and deserialization basics	6
Unit 6	**Collections Framework and GUI Basics** Introduction to Java Collections Framework, Lists, Sets, Maps overview: ArrayList, HashSet, HashMap, Iterators and enhanced for-loop, Generics and type safety, Basics of GUI using AWT and Swing (components: JFrame, JButton, JLabel), Event handling	6
	overview, Integration of OOP concepts in application development, Best practices and coding standards in Java	

Course Outcome:

- Explain the basic structure, syntax, and control flow of Java programs.
- Apply object-oriented programming principles like encapsulation, inheritance, and polymorphism using Java.
- Develop Java programs using abstraction, interfaces, exception handling, and file I/O.
- Analyze and use Java Collections and Generics for data structure manipulation.
- Design simple GUI applications using AWT/Swing and demonstrate event handling in Java.

Text Books

1. Schildt, H. (2018). Java: The complete reference (11th ed.). McGraw-Hill Education.
2. Sierra, K., & Bates, B. (2005). Head First Java (2nd ed.). O'Reilly Media.
3. Horstmann, C. S. (2019). Core Java Volume I – Fundamentals (11th ed.). Prentice Hall.
4. Liang, Y. D. (2021). Introduction to Java programming and data structures (12th ed.). Pearson.

Reference Books:

1. Bloch, J. (2018). Effective Java (3rd ed.). Addison-Wesley.
2. Eckel, B. (2006). Thinking in Java (4th ed.). Prentice Hall.
3. Gosling, J., Joy, B., Steele, G., Bracha, G., & Buckley, A. (2014). The Java™ Language Specification (Java SE 8 ed.). Addison-Wesley.
4. Arnold, K., Gosling, J., & Holmes, D. (2005). The Java programming language (4th ed.). Addison-Wesley.

Online Resources:

1. Oracle. (n.d.). The Java™ Tutorials. Oracle. https://docs.oracle.com/javase/tutorial/
2. GeeksforGeeks. (n.d.). Java Programming Language. https://www.geeksforgeeks.org/java/
3. W3Schools. (n.d.). Java Tutorial. https://www.w3schools.com/java/
4. JavaTpoint. (n.d.). Java Tutorial. https://www.javatpoint.com/java-tutorial
5. Oracle. (n.d.). Java SE Documentation. https://docs.oracle.com/en/java/javase/

List of Experiments (Tentative, Should not be limited to the provided list):

Java Basics and Control Structures

- Write a program to print "Hello World" and basic variable operations.
- Write a program to demonstrate all data types and type casting.
- Write a program to implement decision-making statements (if-else, switch).
- Write a program to demonstrate different loops (for, while, do-while).

Classes, Objects, and Constructors

- Write a program to create a class with methods and instantiate objects.
- Write a program using parameterized and default constructors.
- Write a program to illustrate use of this keyword and static members.
- Write a program to demonstrate method overloading.
- Write a program to show use of access modifiers.

Arrays and Strings

- Write a program to implement single and multi-dimensional arrays.
- Write a program to sort an array and search elements using linear and binary search.
- Write a program to demonstrate string manipulation using String and StringBuilder.
- Write a program to check whether a string is palindrome.

Inheritance, Polymorphism, Abstraction

- Write a program to demonstrate single and multilevel inheritance.
- Write a program to use the super keyword and constructor chaining.
- Write a program to illustrate method overriding and dynamic method dispatch.
- Write a program to create and use abstract classes.
- Write a program to implement interfaces and multiple inheritance using interfaces.

Exception Handling, Packages, File I/O

- Write a program to implement try-catch-finally blocks.
- Write a program to demonstrate use of throw and throws.
- Write a program to create a custom exception.
- Write a program to create and use user-defined packages.
- Write a program to read from and write to a text file using FileReader and FileWriter.

Collections and GUI

- Write a program using ArrayList, HashSet, and HashMap.
- Write a program to demonstrate use of Generics with a class and a method.

- Write a simple GUI application using Swing (JFrame, JButton, JLabel) with event handling.

Course Code	23COPE303 (Theory) / 23COPE353 (Lab)
Course Title	Computer Graphics
Credits	Total: 4 (L:3, T:0, P:1) (3 Hrs Theory, 2 Hours of Lab)
Semester	Fifth Semester
Course Category	Program Elective

Course Objective:

- To understand fundamental concepts and algorithms in computer graphics.
- To explore the mathematics behind 2D and 3D transformations.
- To implement basic graphic algorithms using programming.
- To develop skills in designing simple animations and visual simulations.
- To understand applications of computer graphics in real-world fields.

Unit 1	Introduction to Computer Graphics Definition and applications Overview of graphics systems (raster scan, random scan) Coordinate systems Pixels and resolution	6
Unit 2	Graphics Primitives Drawing basic shapes (line, circle, ellipse) Line drawing algorithms (DDA, Bresenham's) Circle drawing algorithm (Midpoint) Polygon drawing basics	7
Unit 3	2D Transformations Translation, Rotation, Scaling Matrix representation of transformations Homogeneous coordinates Composite transformations	7
Unit 4	3D Graphics Basics 3D coordinate systems Projections: Perspective and Orthographic (basic ideas only) Basic 3D transformations (conceptual level)	7
Unit 5	Clipping and Filling Point, Line, and Polygon Clipping (Cohen-Sutherland Algorithm) Scanline polygon filling Boundary fill and flood fill algorithms	6

Unit 6	Introduction to Graphics Programming Introduction to OpenGL or C graphics (like using Turbo C++/graphics.h) Drawing basic shapes using code	6

Course Outcome:

- Define and explain basic concepts of computer graphics.
- Apply algorithms to draw basic shapes and perform transformations.
- Implement 2D and 3D transformations using matrix methods.
- Develop simple animation and interactive graphic programs.
- Explore applications of computer graphics in various domains.

Text Books

Donald Hearn and M. Pauline Baker, “Computer Graphics”, Pearson Education. A.P. Godse & D.A. Godse, “Computer Graphics”, Technical Publications.

Reference Books:

F.S. Hill Jr. and Stephen M. Kelley, “Computer Graphics Using OpenGL”, Pearson.
Pradeep K. Bhatia, “Computer Graphics”, Khanna Publishing.
Donald Hearn and M. Pauline Baker, “Computer Graphics C Version”, Pearson.

Online Resources:

https://www.geeksforgeeks.org/computer-graphics/
https://nptel.ac.in/courses/106102065
https://www.tutorialspoint.com/computer_graphics/index.htm
https://www.coursera.org/learn/computer-graphics
https://cg2021.com/resources

List of Experiments (Tentative, Should not be limited to the provided list):

1. Drawing Basic Shapes
    - Write a program to draw: o A line using DDA algorithm o A line using Bresenham’s algorithm o A circle using Midpoint Circle Drawing algorithm
2. Drawing Polygons and Ellipses
    - Program to draw:
        - A polygon with user-defined coordinates
        - An ellipse using built-in or user-defined functions
3. 2D Transformations
    - Implement programs to:
        - Perform translation of a polygon
        - Rotate a polygon about the origin

- Scale a polygon
- Apply composite transformation (Translate + Rotate)

4. Line and Polygon Clipping
    - o Implement Cohen-Sutherland line clipping algorithm o Program to clip a polygon using a simple window-based logic
5. Fill Algorithms
    - o Program to fill a polygon using:
    - o Boundary Fill algorithm o Flood Fill algorithm o Scan-line fill (basic polygon)
6. Simple Animation
    - o Create a moving car or ball using basic animation logic o Use delay() and cleardevice() for frame transitions
7. Introduction to OpenGL (Optional / If Available) o Create a window using OpenGL o Draw basic shapes (line, triangle) using OpenGL commands o Apply transformation using OpenGL functions (glTranslatef, glRotatef)
8. Mini Project (Optional)
    - o Create a basic 2D drawing tool o Create a simple animated scene (sunrise, bouncing ball, etc.)

Tools/Software Required

- Turbo C++ or Code::Blocks (with graphics.h)
- OpenGL with GLUT (optional)
- Any simple IDE that supports C/C++

Course Code	23COPE305 (Theory) / 23COPE355 (Practical)
Course Title	Free & Open Source Software (FOSS)
Credits	Total 4 Theory:3 (L:3 T:0 P:1) (3 Hours Theory; 2 Hours Lab)
Semester	Fifth Semester
Course Category	Elective

Course Objective:

- To introduce students to the philosophy, history, and importance of Free and Open Source Software (FOSS).
- To provide a foundation in the use and administration of Linux-based operating systems.
- To expose students to popular open-source tools and platforms for productivity, programming, and collaboration.
- To promote awareness of licensing, copyright, and community-driven development models.

Unit 1	Introduction to Free and Open Source Software Definition: Free Software vs. Open Source Software; History and Evolution of FOSS; Philosophy of Free Software (Richard Stallman, FSF) and Open Source (OSI); Advantages and challenges of FOSS; Popular open-source projects and communities (Linux, Mozilla, LibreOffice, GitHub, etc.) Licensing and Legal Aspects Copyright and Copyleft; Free Software Licenses: GPL, LGPL, BSD, MIT, Apache; Open Standards and Software Patents; Licensing compliance and ethics in FOSS	6
Unit 2	Introduction to Linux Operating System Linux Architecture: Kernel, Shell, File System; Popular Linux Distributions: Ubuntu, Fedora, Debian; Linux Installation; The Linux Desktop Environment; Basic Linux Commands and Navigation; Users, Groups, and File Permissions; Package Management (APT, YUM); Process Management; Input/Output Redirection and Pipes; Text Editors	9
Unit 3	Open Source Tools for Productivity Office Suite: LibreOffice (Writer, Calc, Impress); Browsers and Email Clients: Mozilla Firefox, Thunderbird; Multimedia Tools: VLC Media Player, Audacity; Graphics Tools: GIMP, Inkscape	5
Unit 4	Open Source Development Tools Code Editors: VS Code, Atom; Version Control: Introduction to Git and GitHub; Scripting with Bash and Python (basic concepts); Educational and STEM tools: GCompris, GeoGebra	6
Unit 5	Basic Shell Scripting Introduction to Shell Scripting; Variables and Comments; Basic Input/Output; Conditional Statements; Loops, Case in Shell Script; Creating and Executing Simple Scripts Examples: script to create directories, backup files, display system info.	4
Unit 6	FOSS in Education, Government & Industry FOSS Adoption in Indian Government and Education (e.g., NPTEL, DIKSHA, BOSS OS); Case Studies of Successful FOSS Implementations; Careers and contribution in the FOSS ecosystem	4

Course Outcome:

After completing this course, students will be able to:

- CO1: Differentiate between proprietary and open-source software licenses and articulate the advantages and disadvantages of FOSS.
- CO2: Install and navigate a popular Linux distribution effectively.
- CO3: Execute essential Linux commands for file and directory management, process control, and system information.
- CO4: Manage users, groups, and file permissions in a Linux environment.
- CO5: Utilize common open-source applications for office productivity, web Browse, and multimedia.
- CO6: Write simple shell scripts to automate repetitive tasks.
- CO7: Identify and utilize online resources for FOSS learning and support.

Text Books

1. "Linux System Administration" by Sandeep Kamble, Sayali Parab, Chayan Bhattacharjee
2. A Practical Guide to Linux Commands, Editors, and Shell Programming by Mark G. Sobell
3. Unix: Concepts and Applications by Sumitabh Das
4. Arnold Robbins, “Linux Programming by Examples: The Fundamentals”, Pearson Education
5. "Introduction to Free Software" – FTA (Free Technology Academy)
6. "Linux Pocket Guide: Essential Commands" by Daniel J. Barrett (O'Reilly Media)
7. "Linux for Beginners: An Introduction to the Linux Operating System and Command Line" by Jason Cannon
8. "Beginning Linux Programming" by Richard Stones and Neil Matthew (Wrox)

Reference Books:

1. "Linux Command Line and Shell Scripting Bible" by Richard Blum and Christine Bresnahan
2. "UNIX and Linux System Administration Handbook" by Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley
3. "Producing Open Source Software" by Karl Fogel

Online Resources:

- https://onlinecourses.swayam2.ac.in/aic20_sp24/preview
- https://nptel.ac.in/courses/106105214
- https://linuxjourney.com/
- https://training.linuxfoundation.org/resources/
- Open Source Initiative (OSI): https://opensource.org/
- GNU Project: https://www.gnu.org/

List of Experiments (Tentative, Should not be limited to the provided list):

1. Installing a Linux OS (Ubuntu/CentOS) on VirtualBox or hardware
2. Basic shell commands and file operations
3. User, group and file permission management
4. Package installation and software updates
5. Monitoring system processes and managing services
6. Basic networking setup and troubleshooting
7. Writing and executing basic shell scripts
8. Automating tasks with cron

Course Code	23COPE307
Course Title	Cyber Security & Cyber Law
Credits	Total 3 (L:3 T:0 P:0)
Semester	Fifth Semester
Course Category	Elective

Course Objective:

- To introduce students to fundamental concepts in cyber security.
- To create awareness about cyber threats, attacks, and security measures.
- To familiarize students with Indian cyber laws and their importance.
- To build an understanding of safe online practices and digital ethics.
- To develop responsible digital citizens capable of identifying and reporting cyber crimes.

Unit 1	Introduction to Cyber Security Definition and Importance of Cyber Security; Key Concepts: Threats, Vulnerabilities, Risks; Types of Cyber Attacks: Phishing, Malware, Ransomware, DoS/DDoS, Social Engineering; Cyber Security Practices: Passwords, Firewalls, Anti-virus, Software Updates	6
Unit 2	Network & Data Security Introduction to Computer Networks; Safe Browsing and Email Security; Data Encryption and Decryption (Basic Concepts); Backups and Data Protection; Security in Mobile and Wireless Devices	6
Unit 3	Cyber Ethics and Safe Computing Practices Digital Footprints and Privacy; Responsible Use of Social Media and Messaging Apps; Cyber Bullying, Online Harassment, and Prevention; Good Practices: Digital Hygiene, Secure Communication, Multi-Factor Authentication	6
Unit 4	Introduction to Cyber Law in India IT Act 2000 and Amendments; Legal Definitions: Cyber Crime, Identity Theft, Hacking, Data Breach; Offences and Penalties under the IT Act; Intellectual Property Rights in Cyberspace	6

Unit 5	Cyber Crime Investigation and Reporting How to Report a Cyber Crime; Role of CERT-In, Cyber Crime Cells; Real-life Case Studies of Cyber Crimes in India; Preventive and Legal Measures for Protection	6

Course Outcome:

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

- CO1: Identify different types of cyber threats and attacks.
- CO2: Explain basic cyber security concepts and tools.
- CO3: Describe the legal framework governing cyber security in India.
- CO4: Practice safe and ethical use of technology and the internet.
- CO5: Understand the process for reporting cyber crimes and legal remedies.

Text Books

1. Cyber Security Essentials by James Graham, Richard Howard, and Ryan Olson – CRC Press
2. Introduction to Cyber Security by Chwan-Hwa (John) Wu and J. David Irwin – CRC Press
3. Cyber Laws by Vakul Sharma – Universal Law Publishing
4. Cyber Security & Cyber Laws by Alfred Basta, Nadine Basta – Cengage Learning

Reference Books:

1. Information Security: Principles and Practice by Mark Stamp – Wiley
2. Cyber Law: Indian and International Perspectives on Key Topics by Talat Fatima – LexisNexis
3. Cyber Crime and Security by Sunit Belapure and Nina Godbole – Wiley India

Online Resources:

- https://nptel.ac.in/courses/106105217
- https://www.csk.gov.in/
- https://cybercrime.gov.in/

Course Code	23COPE309
Course Title	Multimedia Technologies
Credits	Total 3 Theory: 3 (L: 3 T: 0 P: 0)
Semester	Fifth Semester
Course Category	Elective

Course Objective:

- To introduce the fundamental components and formats of multimedia. ● To explore image, audio, video, and animation tools and technologies.
- To understand multimedia compression techniques and file standards.
- To study the use of multimedia in web applications, education, and entertainment.

Unit 1	Definition and elements of multimedia – Multimedia system architecture – Text, image, audio, video, animation – Applications of multimedia in education, business, entertainment, and web – Multimedia authoring tools and software overview.	6
Unit 2	Types of images – Raster vs vector graphics – Image file formats: BMP, JPEG, PNG, GIF – Color models: RGB, CMY, HSV – Basic image editing and enhancement techniques – Overview of graphic design tools (Photoshop, Canva, GIMP).	8
Unit 3	Audio Technology: Sound basics: frequency, amplitude, sampling – Analog to digital conversion – Audio file formats: MP3, WAV, AAC – Audio compression basics – Sound editing and mixing – Overview of audio editing tools (Audacity, GarageBand).	9
Unit 4	Video and Animation Basics of video signal – Frame rate, resolution, interlacing – Video file formats: AVI, MP4, MOV – Video compression techniques (MPEG, H.264) – Introduction to 2D/3D animation – Tools: Adobe Premiere, Blender (overview only).	8
Unit 5	Need for compression – Lossy vs lossless compression – Image compression (JPEG), Audio compression (MP3), Video compression (MPEG) – Multimedia standards: MPEG, JPEG, H.264, MP4, MP3 – Introduction to streaming concepts.	8
Unit 6	Multimedia on the web – HTML5 multimedia tags – Introduction to multimedia authoring tools – Multimedia in e-learning and gaming – Basic UI/UX principles for multimedia applications – Case studies: YouTube, Canva, Google Arts & Culture.	8

Course Outcome:

After completing this course, students will be able to:

- CO1: Identify and explain the key components of multimedia systems.
- CO2: Work with image, audio, and video file formats and tools.
- CO3: Understand basic compression techniques and multimedia standards.
- CO4: Analyze multimedia integration in websites, education, and media platforms

Text Books

1. Tay Vaughan – Multimedia: Making It Work – McGraw-Hill
2. Ranjan Parekh – Principles of Multimedia – McGraw-Hill Education India
3. Ralf Steinmetz and Klara Nahrstedt – Multimedia Systems – Pearson

Reference Books:

1. Ze-Nian Li and Mark S. Drew – Fundamentals of Multimedia – Pearson
2. Nigel Chapman – Digital Multimedia – Wiley
3. Gokhale & Wadhwani – Multimedia Technology – Everest Publishing

Online Resources:

- https://developer.mozilla.org/en-US/docs/Web/HTML/Element/video
- https://www.gimp.org – Open-source image editor
- https://www.audacityteam.org – Audio editor
- https://www.blender.org – 2D/3D animation tool
- https://nptel.ac.in – NPTEL multimedia video lectures

List of Experiments (Tentative, Should not be limited to the provided list):

1. Create a multimedia presentation using text, images, and animations (e.g., Canva or PowerPoint).
2. Edit and export an image using GIMP or Photoshop – apply filters and transformations.
3. Convert and compress an image using online or offline tools (JPEG, PNG).
4. Record and edit audio using Audacity – apply noise reduction and export in MP3.
5. Create a short video with subtitles and effects using OpenShot or Adobe Premiere.
6. Compress a video using HandBrake or FFmpeg and compare file sizes.
7. Embed audio and video in an HTML5 web page using <audio> and <video> tags.
8. Create a basic UI design for a multimedia-rich mobile/web interface using Figma or pen-paper.
9. Demonstrate streaming a video using YouTube Live or similar platform (demo only).
10. Create an animation using Blender or any 2D animation software (basic motion

Course Code	23COOE311 / 23COOE357 (Practical)

Course Title	Fundamentals of AI
Credits	Total: 3 (Theory - 3), L:3 T:0 P:2 (3 Hrs Theory, 4 Hours of Lab)
Semester	Fifth Semester
Course Category	Open Elective

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

- Understand the basic concepts and definitions of Artificial Intelligence.
- Identify the different domains and applications of AI in daily life and various industries.
- Appreciate the historical evolution and current trends in AI.
- Recognize the ethical considerations and societal impact of AI.
- Develop an initial understanding of how machines "learn" through simple examples.
- Be familiar with basic problem-solving approaches in AI.

Unit	Description	Hours
Unit 1	Introduction to Artificial Intelligence: What is AI? Definition of Artificial Intelligence (AI); Intelligence vs. Artificial Intelligence; Goals of AI: Thinking humanly, acting humanly, thinking rationally, acting rationally; Brief history of AI (key milestones and breakthroughs) Domains of AI; Introduction to key AI domains: Machine Learning (ML); Natural Language Processing (NLP); Computer Vision (CV); Robotics; Expert Systems; (Brief mention of other areas like Reinforcement Learning, Generative AI) Applications of AI; Real-world examples of AI in various sectors: Healthcare (e.g., diagnostics, drug discovery); Finance (e.g., fraud detection, algorithmic trading); Education (e.g., personalized learning); Transportation (e.g., self-driving cars, route optimization) Entertainment (e.g., recommendation systems, game AI); Smart assistants (e.g., Siri, Alexa); Impact of AI on daily life. AI vs. Related Fields; Distinction between AI, Machine Learning, and Deep Learning (conceptual overview); How AI differs from traditional programming.	8
Unit 2	Problem Solving in AI: Intelligent Agents; Concept of agents and environments; Rational agents; Examples of simple agents; Problem Formulation: Defining a problem for AI to solve (initial state, goal state, actions); Examples of well-defined problems; Search Techniques (Conceptual); Introduction to problem-solving by search; Uninformed Search (brief idea of Breadth-First Search, Depth-First Search – no algorithms, just concept); Heuristic Search; Rule-Based Systems: Introduction to "If-Then" rules; Simple expert systems (concept and examples)	8

Unit 3	Introduction to Machine Learning: What is Machine Learning? Definition and goal of Machine Learning; Learning from data: the need for data; Supervised, Unsupervised, and Reinforcement Learning (conceptual understanding with simple examples); Data Basics for ML: What is data? Types of data (numbers, text, images - very basic); Importance of data collection and quality Simple idea of data preprocessing (e.g., handling missing values, cleaning data - without technical depth); Supervised Learning (Overview); Classification: Grouping data into categories (e.g., spam detection, image recognition of cat/dog, etc). Simple example: Decision Trees (conceptual: how decisions are made based on features). Regression: Predicting continuous values (e.g., predicting house prices). Simple example: Linear Regression (conceptual: finding a line to fit data points). Unsupervised Learning (Overview):	10
	Clustering: Grouping similar data points (e.g., customer segmentation). Simple example: K-Means Clustering (conceptual: forming clusters based on proximity). Model Evaluation (Basic): Why evaluate a model? Simple ideas of accuracy (e.g., how many predictions were correct).	
Unit 4	Domains in Action: Natural Language Processing (NLP): Understanding human language by machines. Applications: Chatbots, sentiment analysis, machine translation, speech recognition (brief mention of speech-to-text). Computer Vision (CV): Enabling machines to "see" and interpret images/videos. Applications: Facial recognition, object detection, image classification, autonomous vehicles. Robotics: Introduction to intelligent robots. How AI is used in robotics (e.g., perception, decision-making).	6
Unit 5	Ethical AI and Future Trends: Ethical Considerations in AI: Bias in AI (e.g., biased data leading to unfair outcomes). Privacy concerns (data collection and usage). Accountability and transparency of AI systems. Misuse of AI. Responsible AI: Importance of developing AI responsibly. Fairness, accountability, and transparency (FAT) principles. Future of AI: Emerging trends (e.g., Generative AI, Explainable AI). Potential impact of AI on society and jobs. The role of humans in an AI-driven future.	4

Course Outcome:

- Understand the foundational concepts and definitions of Artificial Intelligence.
- Identify and articulate diverse real-world applications of AI across various sectors.
- Describe basic problem-solving approaches used in Artificial Intelligence.
- Explain the fundamental principles of Machine Learning and its basic types.
- Recognize and discuss the ethical considerations and societal implications of Artificial Intelligence.

Text Books

1. "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig
2. "Artificial Intelligence For Dummies" by John Paul Mueller and Luca Massaron
3. "AI for Everyone: A Common Man's Guide to Artificial Intelligence" by Sridhar Seshadri
4. "AI for Everyone: A Beginner's Handbook for Artificial Intelligence (AI)" by Saptarsi Goswami
5. "Life 3.0: Being Human in the Age of Artificial Intelligence" by Max Tegmark

Reference Books:
1. "Artificial Intelligence: A Guide for Thinking Humans" by Melanie Mitchell 2. Machine Learning For Absolute Beginners: A Plain English Introduction" by Oliver Theobald 3. "Human Compatible: Artificial Intelligence and the Problem of Control" by Stuart Russell

Online Resources:
Experiments with Google: https://experiments.withgoogle.com/collection/ai https://onlinecourses.nptel.ac.in/noc25_cs149/preview https://archive.nptel.ac.in/noc/courses/noc21/SEM1/noc21-cs42/ https://onlinecourses.nptel.ac.in/noc25_cs51/preview

List of Experiments (Tentative, Should not be limited to the provided list):
1. Explore AI Applications Around You; Identify and document real-life AI applications from different sectors (e.g., healthcare, finance, education). Create a presentation/report using examples like chatbots, Netflix recommendations, etc. 2. AI vs. Traditional Programming; Compare rule-based (traditional) and data-driven (AI-based) approaches. Show a rule-based calculator vs. an AI-based number classifier. 3. Build a Simple Reflex Agent; Implement a basic reflex agent (e.g., vacuum cleaner agent or temperature control). Use conditional "if-then" rules to simulate an agent in a grid. 4. Problem Formulation; Define AI problems using initial state, goal state, and actions. Formulate a puzzle (e.g., 8-puzzle or pathfinding on a map). 5. Visualizing Search Techniques; Demonstrate BFS vs DFS using a maze or tree. Use a visual online simulator or Python animation. 6. Explore Data Types in ML; Identify and classify different types of data (text, image, number). Collect sample data and tag types. 7. Classification using Decision Trees; Conceptual understanding of decision-making using features. Use scikit-learn to classify fruits based on features like size, color. 8. Regression using Linear Model; Predict numerical output from input data. Predict student marks based on study hours. 9. Unsupervised Learning - K-Means Clustering; Group unlabeled data based on similarity. Cluster 2D data points visually. 10. Model Accuracy (Basic Evaluation); Calculate accuracy of simple classifiers. Compare predicted vs actual labels in Excel. 11. Sentiment Analysis using Online Tools; Understand how NLP can interpret emotions. Use an online sentiment analysis tool for tweets or text reviews. 12. Image Classification using Teachable Machine; Explore Computer Vision without coding. Train a model to recognize objects or hand gestures. 13. Introduction to Robotics Simulation; Visualize how robots perceive and act using sensors. Simulate a line-following robot. https://studio.polygon.com/ or Tinkercad Circuits (for beginners) 14. Case Study – Bias in AI; Explore how biased data affects model outcomes. Analyze a dataset with imbalance (e.g., gender-based hiring) and simulate results. Practical 15: Research Poster – Future Trends in AI 15. Explore Generative AI, Explainable AI, etc. Create a poster/presentation on one future trend with examples.

Course Code	23COOE313 (Theory) / 23COOE359 (Practical)

Course Title	Fundamentals of Machine Learning
Credits	Total 5 (L:3 T:0 P:2) (3 Hrs Theory, 4 Hours of Lab)
Semester	Fifth Semester
Course Category	Open Elective

Course Objective:

- To introduce basic concepts and applications of Machine Learning.
- To expose students to supervised and unsupervised learning techniques.
- To familiarize students with Python-based ML libraries like Scikit-learn.
- To provide hands-on experience in building and evaluating ML models.

Unit 1	Introduction to Machine Learning (4 hours) What is Machine Learning? History & Evolution of ML; Types of Learning: Supervised, Unsupervised, Reinforcement (Intro only); Applications of ML Data Handling for Machine Learning Datasets: Features and Labels; Data Cleaning & Preprocessing (missing data, normalization, encoding); Data Splitting (Train/Test); Feature Scaling	8
Unit 2	Getting Started with Python for Machine Learning Introduction to Python and its basic syntax Introduction to essential libraries: NumPy for numerical operations, Pandas for data manipulation, and Matplotlib/Seaborn for data visualization. Reading and writing data from/to different file formats (CSV, Excel).	6
Unit 3	Supervised Learning Regression (Linear Regression); Introduction to Regression.; Simple Linear Regression: Concept and implementation. ; Multiple Linear Regression: Concept and implementation. Model evaluation metrics for regression: Mean Absolute Error (MAE), Mean Squared Error (MSE), R-squared. Classification (k-Nearest Neighbors, Decision Trees); Introduction to Classification. Logistic Regression.; K-Nearest Neighbors (KNN).; Decision Trees; Model evaluation metrics for classification: Accuracy, Confusion Matrix, Precision, Recall, F1-score.	10
Unit 4	Unsupervised Learning Clustering: K-means Algorithm; Dimensionality Reduction: PCA (Basic Concepts)	5
Unit 5	Model Evaluation and Selection Cross Validation; Overfitting and Underfitting; Confusion Matrix	3
Unit 6	Introduction to Advanced Topics Basics of Neural Networks; A brief introduction to Artificial Neural Networks (ANNs). Ethical issues in Machine Learning: Bias, Fairness, and Privacy.	2

Course Outcome:

- Understand fundamental concepts of machine learning and its types.
- Implement simple machine learning algorithms in Python.
- Analyze and evaluate the performance of ML models.
- Work with datasets using preprocessing and visualization techniques.
- Develop ML models for real-world problems.

Text Books

1. Introduction to Machine Learning by Alpaydin, MIT Press
2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron 3. Python Machine Learning by Sebastian Raschka

Reference Books:

1. Machine Learning for Absolute Beginners by Oliver Theobald
2. Pattern Recognition and Machine Learning by Christopher Bishop
3. An Introduction to Statistical Learning by Gareth James et al.
4. "Python for Data Analysis" by Wes McKinney
5. "Introduction to Machine Learning with Python: A Guide for Data Scientists" by Andreas C. Müller & Sarah Guido

Online Resources:

- https://developers.google.com/machine-learning/crash-course
- https://scikit-learn.org/stable/
- https://www.kaggle.com/
- https://www.w3schools.com/python/python_ml_getting_started.asp ● https://ai.google/education/

List of Experiments (Tentative, Should not be limited to the provided list):

- Lab Setup
  - Python with Jupyter Notebook
  - Libraries: NumPy, Pandas, Matplotlib, Scikit-learn
- Lab Experiments
  - Installing ML Environment
  - Anaconda / Jupyter setup, installing packages
- Working with Datasets
  - Load CSV using Pandas, explore data using .info(), .describe()
- Data Preprocessing
  - Handle missing values, normalize data, encode categorical variables
- Linear Regression
  - Predicting scores/price from simple datasets
- Classification using k-NN
  - Classify Iris dataset
- Decision Tree Classifier
  - Visualize and test tree model on simple dataset
- K-Means Clustering
  - Cluster data points, visualize clusters
- Model Evaluation
  - Compute confusion matrix, accuracy, precision, recall
- Mini Project
  - Real-world dataset analysis (e.g., predicting student grades, customer segmentation)