ANNA UNIVERSITY, CHENNAI

NON- AUTONOMOUS COLLEGES

AFFILIATED TO ANNA UNIVERSITY

MASTER OF COMPUTER APPLICATIONS (2 YEARS)

REGULATIONS 2025

PROGRAMME OUTCOMES (POs):

PO	Programme Outcomes
PO1	An ability to independently carry out research /investigation and development work to solve practical problems
PO2	An ability to write and present a substantial technical report/document.
PO3	Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

PROGRAMME SPECIFIC OUTCOMES:
PSO1: Design and develop computer applications by applying programming, database, networking, and system software knowledge to meet real-world requirements.
PSO2: Apply modern computing technologies and frameworks, including AI, cloud, and web platforms, to build efficient, user-friendly, and secure applications.

ANNA UNIVERSITY, CHENNAI

POSTGRADUATE CURRICULUM (NON-AUTONOMOUS AFFILIATED INSTITUTIONS)

Programme: Master of Computer Applications (2 Years) **Regulations:** 2025

Abbreviations:

BS – Basic Science (Mathematics)

ES – Engineering Science (Programme Core (**PC**), Programme Elective (**PE**))

SD – Skill Development

SL – Self Learning

OE – Open Elective

L – Laboratory Course

T – Theory

LIT – Laboratory Integrated Theory

PW – Project Work

TCP – Total Contact Period(s)

Semester – I

S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
1.	MC25101	Mathematical Foundation of Computer Applications	T	3	1	0	4	4	BS
2.	MC25102	Networking and Communication Systems	T	3	0	0	3	3	ES (PC)
3.	MC25103	Software Testing and Automation	T	3	0	0	3	3	ES (PC)
4.	MC25104	Data Structures and Algorithms using Python	LIT	2	0	2	4	3	ES (PC)
5.	MC25105	Data Exploration and Visualization	LIT	2	0	2	4	3	ES (PC)
6.	MC25106	Advanced Java Programming	LIT	2	0	2	4	3	ES (PC)
7.	MC25107	Technical Seminar	-	0	0	2	2	1	SD
Total Credits							**24**	**20**	

Semester – II

S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
1.		UI & UX Design	T	3	0	0	3	3	ES (PC)
2.		Programme Elective I	T	3	0	0	3	3	ES (PE)
3.		Foundations of Data Science	LIT	3	0	2	5	4	ES (PC)
4.		Full Stack Web Development	LIT	3	0	4	7	5	ES (PC)
5.		Mobile Application Development	LIT	3	0	4	7	5	ES (PC)
6.		Quantum Computing	T	2	0	0	2	2	ES (PC)
7.		Industry-Oriented Course I	-	1	0	0	1	1	SD
8.		Industrial Training	-	-	-	-	-	2	SD
9.		Self-Learning Course	-	-	-	-	-	1	-
Total Credits							**28**	**26**	

Semester – III

S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
1.		Programme Elective II	T	3	0	0	3	3	ES (PE)
2.		Programme Elective III	T	3	0	0	3	3	ES (PE)
3.		Programme Elective IV	T	3	0	0	3	3	ES (PE)
4.		Open Elective	T	3	0	0	3	3	-
5.		Machine Learning Techniques	LIT	3	0	2	5	4	ES (PC)
6.		Industry-Oriented Course II	--	1	0	0	1	1	SD
Total Credits							**18**	**17**	

Semester – IV

S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
1.		Project Work	-	0	0	20	20	10	SD
Total Credits							**20**	**10**	

Programme Elective Courses (PEC)

S. NO.	Course Code	Course Title	Periods Per Week			Total Contact Periods	Credits
			L	T	P		
1.		Big Data Analytics	3	0	0	3	3
2.		VIBE coding	3	0	0	3	3
3.		Network Programming and Management	3	0	0	3	3
4.		Software Project Management	3	0	0	3	3
5.		E-Learning	3	0	0	3	3
6.		Accounting and Financial Management	3	0	0	3	3
7.		Digital and Mobile Forensics	3	0	0	3	3
8.		Cryptocurrency and Blockchain Technologies	3	0	0	3	3
9.		Game Programming	3	0	0	3	3
10.		Social Network Analysis	3	0	0	3	3
11.		Entrepreneurship Development	3	0	0	3	3
12.		Generative AI and Prompt Engineering	3	0	0	3	3
13.		Wireless Sensor Networks and Body Area Network	3	0	0	3	3
14.		Fog and Edge Computing	3	0	0	3	3
15.		Internet of Things	3	0	0	3	3
16.		Deep Learning Techniques	3	0	2	5	4
17.		Artificial Intelligence	3	0	2	5	4
18.		DevOps	3	0	2	5	4
19.		Ethical Hacking and Penetration Testing	3	0	2	5	4
20.		E–Commerce Technologies	3	0	2	5	4

Semester I

MC25101	Mathematical Foundation for Computer Applications	L	T	P	C
		3	1	0	4

Course Objectives:

- To apply fundamental concepts of discrete mathematics in the context of computer science and software development.
- To develop the ability to model and solve computational problems using mathematical tools such as combinatorics, matrices, graph theory, and number theory.
- To enhance analytical and logical thinking skills necessary for designing algorithms, analyzing complexity, and supporting various applications in computer science.

Relations and Functions: Properties of Binary Relations, equivalence, transitive closure, compatibility and partial ordering relations – Inverse Function Composition of functions, recursive functions Lattices – Hasse diagram, Lattice and its Properties.

Propositional and predicate Logic: Propositional logic – Logical connectives – Truth tables – Normal forms (principal conjunctive and principal disjunctive normal forms) – Predicate logic – Universal and existential quantifiers – Proof techniques – Direct and indirect proofs – Proof by contradiction – Mathematical Induction.

Recurrence Relation and Generating Function: Generating Functions – Function of Sequences Calculating Coefficient of generating function, Recurrence relations – Solving recurrence relation by substitution and Generating funds. Characteristics roots solution of In homogeneous Recurrence Relation.

Graph Theory: Basic terminology: Different types of graphs – Directed and undirected, Simple, Pseudo, Complete, Regular, Bipartite. Incidence and degree, Pendant and Isolated vertex and Null graph. Isomorphism, Sub graphs, Walk, Path and Circuit, Connected and disconnected graphs and components, operations on graphs. Euler Graphs, Fleury's Algorithm – Hamiltonian circuits and paths. Traveling salesman problem. Matrix representation of graphs – Incidence and Adjacency matrices.

Grammars and Automata: Alphabets, Strings and Languages; Automata and Grammars, Deterministic finite Automata (DFA) – Formal Definition, Simplified notation: State transition graph, Transition table – Language of DFA, Nondeterministic finite Automata (NFA), NFA with epsilon transition – Language of NFA, Equivalence of NFA and DFA, Minimization of Finite Automata.

Weightage: Continuous Assessment: 40%, End Semester Examinations: 60%.

Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).

References:

1. Trembly, J. P., & Manohar, R. (2017). Discrete mathematical structures with applications to computer science (International ed.). Tata McGraw-Hill.
2. Rosen, K. H. (2017). Discrete mathematics and its applications (7th ed.). McGraw-Hill.
3. Deo, N. (2016). Graph theory with applications to engineering and computer science. Prentice Hall.
4. Hopcroft, J. E., & Ullman, J. D. (2006). Introduction to automata theory, languages, and computation (3rd ed.). Pearson Education.
5. Sharma, J. K. (2011). Discrete mathematics (3rd ed.). Macmillan Publishers India.

E-resources:

1. https://ocw.mit.edu/courses/6-042j-mathematics-for-computer-science-fall-2010/?
2. https://teachyourselfcs.com/?

MC25102	Networking and Communication Systems	L	T	P	C
		3	0	0	3

Course Objectives:

- To provide a comprehensive understanding of networking principles and communication models of modern communication systems.
- To enable students to analyze, design, and evaluate computer networks and communication systems.
- To develop skills in applying networking and communication concepts to real-world applications

Networking Fundamentals and Protocol Models: Basics of storage, Need for modern storage, Overview of data growth and digital transformation, Features of DAS, NAS, SAN, CAS, Introduction to RAID: types, levels, and configurations, Disk subsystems, Performance and reliability considerations, Traditional vs. software-defined storage.

Activity: Create visual OSI vs TCP/IP Model Mapping Chart with real-world protocol examples at each layer.

Data Link and Network Layer Protocols: Data link layer services, design issues, error detection and correction, Medium Access Control (MAC): Ethernet, CSMA/CD, CSMA/CA, Network layer functions and IP routing, store and forward packet switching, connection less and connection, oriented networks, Routing algorithms: Distance vector, Link state, OSPF, BGP, ARP, ICMP, and DHCP operations

Activity: Simulate IP Routing in a Virtual LAN using Packet Tracer Tool.

Transport and Application Layer Protocols: Transport service, elements of transport protocol, Simple Transport Protocol, Internet transport layer protocols: UDP and TCP, Domain name system, electronic mail, World Wide Web: architectural overview, dynamic web document and http. Simple Network Management Protocol, File Transfer Protocol, Simple Mail Transfer Protocol.

Activity: Demonstration of Transport and Application layer protocols.

Wireless and Mobile Communication Systems: Cellular networks and mobile IP,Wireless LANs and IEEE 802.11, Bluetooth, ZigBee, and NFC basics, Mobile communication challenges: handoff, roaming, latency, MANETs and VANETs

Activity: Realization of Blue Tooth and ZigBee Network

Modern Network Trends, IoT, Cloud & Secure Networking: Cloud networking and virtualization, IoT architecture and protocols (MQTT, CoAP), Edge and fog computing, Network security basics: firewalls, encryption, VPNs, Network challenges in modern computing (5G, SDN, NFV)

Activity: Demonstration of IoT Architecture for Home Automation
Weightage: Continuous Assessment: 40%, End Semester Examinations: 60%.
Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).
References: 1. Tanenbaum, A. S., & Wetherell, D. J. (2014). Computer Networks. Pearson Education, Elsevier Inc. 2. Gupta, P. C. (2014). Data communications and computer network. PHI Learning Pvt Ltd. 3. Forouzan, B. A. (2013). Introduction to data communication & networking. McGraw Hill Education Pvt Ltd. 4. Larry, P., Peterson, L., & Davis, B. S. (2014). Computer networks*:* A system approach. Elsevier Inc. 5. Forouzan, B. A. (2022). Data communications and networking. McGraw-Hill.
E-RESOURCES: 1. Computer Networks by Prof. S. K. Sinha, IIT Kharagpur https://nptel.ac.in/courses/106105183 2. Introduction to Wireless and Cellular Communications by IIT Madras https://nptel.ac.in/courses/117106114

	Description of CO	PO	PSO
CO1	Elaborate the fundamental concepts of data communication, OSI, and TCP/IP models.	--	--
CO2	Apply transmission media, switching, and error control techniques to optimize communication efficiency	PO1(3)	PSO1(3)
CO3	Design and simulate routing, congestion control, and network protocols for efficient communication systems.	PO3(2)	PSO2(2)
CO4	Evaluate and justify emerging networking technologies and security mechanisms for real-world applications.	PO2(1)	PSO1(3)

MC25103	Software Testing and Automation	L	T	P	C
		3	0	0	3

Course Objectives:

- To impart knowledge of software testing principles and life cycle models for ensuring software quality and reliability.
- To enable students to design effective test cases and utilize automation tools for functional and non-functional testing.
- To develop skills in implementing automated testing frameworks for real-world software projects.

Foundations of Software Testing: Foundations of Software Testing Introduction to test Software, Black-Box Testing and White-Box Testing-Software Testing Life Cycle, V- model of Software Testing-Program Correctness and Verification, Reliability versus Safety, Failures, Errors and Faults (Defects), Software Testing Principles, Program Inspections, Stages of Testing, Unit Testing-Integration Testing, System Testing.

Activities:

- Creation of Black-Box and White, Box test cases based on functional and Non-functional requirements.
- Simulation of Software Testing Life Cycle to understand the stages of testing

Test Planning: The Goal of Test Planning, High Level Expectations, Intergroup Responsibilities, Test Phases, Test Strategy, Resource Requirements, Tester Assignments, Test Schedule, Test Cases, Bug Reporting, Metrics and Statistics.

Activities:

- Design a comprehensive test plan for a sample software project,
- Bug Reporting and Metrics Simulation.

Test Design and Execution: Test Objective, Identification, Test Design, Factors, Requirement, identification, Testable, Requirements, Modeling a Test Design Process, Modeling Test Results, Boundary Value Testing, Equivalence Class Testing, Path Testing-Data Flow Testing, Test Design Preparedness Metrics, Test Case Design Effectiveness, Model, Driven Test Design, Test Procedures, Test Case Organization and Tracking-Bug Reporting, Bug Life Cycle.

Activities:

- Conduct load, stress, volume, fail-over, recovery, and configuration tests on a sample application
- Test a web or mobile application for usability, compatibility, security.

Advanced Testing Concepts: Performance Testing: Load Testing, Stress Testing, Volume Testing, Fail-Over Testing, Recovery Testing-Configuration Testing, Compatibility Testing, Usability Testing, Testing the Documentation, Security testing, Testing in the Agile Environment-Testing Web and Mobile Applications.

Activities:

- Perform configuration tests on a web or mobile application using performance testing tools
- Performing iterative testing in an Agile environment on web and mobile applications.

Test Automation and Tools: Automated Software Testing, Automate Testing of Web Applications-Selenium: Introducing Web Driver and Web Elements, Locating Web Elements, Actions on Web Elements, Different Web Drivers, Understanding Web Driver Events, Testing: Understanding Testing.xml, Adding Classes, Packages, Methods to Test-Test Reports.

Activities:

- Use Selenium WebDriver to generate test reports.
- Design automated test scripts on a sample web application

Weightage: Continuous Assessment: 40%, End Semester Examinations: 60%.

Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).

References:

1. Desikan, S., & Ramesh, G. “Software testing: Principles and practices”. Pearson Education.
2. Westerveld, D. (2021). “API testing and development with Postman”. Packt Publishing.
3. Gundecha, U. “Selenium Web Driver 3 practical guide”. Packt Publishing.
4. Rodrigues, A. G. “Mastering JMeter: From load testing to DevOps”. Packt Publishing.
5. Menon, V. “TestNG beginner’s guide”. Packt Publishing.

E- Resources:

1. NPTEL: Software Testing by Prof. Yogesh Singh
2. Postman Tutorials: Postman.com
3. JMeter Documentation: https://jmeter.apache.org/
4. TestNG Documentation: https://testng.org/doc/

	Description of CO	PO	PSO
CO1	Explain software testing principles and defect management to ensure software reliability.	--	--
CO2	Apply test planning strategies using black-box and white-box techniques.	PO1(3)	PSO1(3)
CO3	Develop automated test scripts to validate functionality, performance, and security.	PO3(2)	PSO2(2)
CO4	Evaluate advanced testing methodologies to address real-world challenges.	PO2(1)	PSO1(3)

MC25104	Data Structures and Algorithms using Python	L	T	P	C
		2	0	2	3

Course Objectives:

- To provide a strong understanding of fundamental data structures and their implementation in Python.
- To enable students to design, analyze, and implement efficient algorithms for searching, sorting, and problem-solving using Python.
- To develop skills in applying algorithmic thinking and data structure concepts to real-world computational problems

Python Programming and Algorithmic Analysis: Python environment, variables, operations, control flow, conditionals, loops, and functions, Python types and expressions, Strings, lists, and tuples, names, mutable and immutable values, List operations and slicing, Dictionaries, Functions, Object oriented programming, Abstract data types, Classes and objects, Encapsulation and data abstraction, Inheritance, Namespace and Object Orientation.

Practical:

- Student Record Management System (Procedural Approach)
- Bank Account Simulation Using OOP

Abstract Data Types and Linear Structures: Abstract Data Types (ADTs), ADTs and classes, introduction to OOP, classes in Python, inheritance namespaces, shallow and deep copying Introduction to analysis of algorithms, asymptotic notations divide & conquer, recursion, analyzing recursive algorithms- Stack ADT, Queue ADT

Practical:

- Implement Custom Stack and Queue ADTs with Recursion and Algorithm Analysis
- Recursive Merge Sort with Operation Count and Custom Object Sorting

Sorting and Searching: Sorting Algorithms: Bubble sort, selection sort, insertion sort, merge sort, quick sort, analysis of sorting algorithms, Searching Techniques: linear search, binary search, hashing, hash functions, collision handling, load factors, rehashing, and efficiency

Practical:

- Implement sorting algorithms in Python Bubble Sort**,** Selection Sort
- Implement Linear Search, Binary Search, Hash Table with Collision Handling

Tree Structures: Tree ADT, Binary Tree ADT, tree traversals,binary search trees, AVL trees, heaps, multiway search trees

Practical:

- lement Binary Tree ADT
- Implement an AVLTree class with Rotations (LL, RR, LR, RL)

Graph Structures: Graph ADT, representations of graph, graph traversals, DAG, topological ordering, greedy algorithms, dynamic programming, shortest paths, minimum spanning trees, Introduction to complexity classes and intractability

Practical:

- Implement Breadth-First Search (BFS), Depth-First Search (DFS)
- Implement Dijkstra's Algorithm for shortest path (Greedy)

Weightage: Continuous Assessment: 50%, End Semester Examinations: 50%

Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20)

References:

1. Agarwal, B., & Baka, B. (2018). Hands-on data structures and algorithms with Python: Write complex and powerful code using the latest features of Python 3.7. Packt Publishing.
2. Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2021). Data structures & algorithms in Python (Indian adaptation). John Wiley & Sons.
3. Lee, K. D., & Hubbard, S. (2015). Data structures and algorithms with Python. Springer.
4. Necaise, R. D. (2011). Data structures and algorithms using Python. John Wiley & Sons.
5. Weiss, M. A. (2014). Data structures and algorithm analysis in C++. Pearson Education.

E-resources:

1. https://runestone.academy/ns/books/published/pythonds/index.html
2. https://www.geeksforgeeks.org/data-structures-and-algorithms-in-python/

	Description of CO	PO	PSO
CO1	Elaborate programming concepts and object-oriented paradigms to solve computational problems using python.	--	--
CO2	Apply linear and nonlinear data structures to solve algorithmic problems efficiently.	PO1(3)	PSO1(3)
CO3	Evaluate and implement classical algorithms for sorting, searching, traversing, and optimization with complexity analysis.	PO3(2)	PSO2(2)
CO4	Design and implement Python-based applications using ADTs to solve real-world computational challenges	PO2(1)	PSO1(3)

MC25105	Data Exploration and Visualization	L	T	P	C
		2	0	2	3

Course Objectives:

- To provide students with the ability to explore datasets to uncover patterns.
- To enable learners to create effective visualizations to communicate insights clearly.
- To develop skills in presenting data-driven insights for informed decision-making.

Exploratory Data Analysis: Identifying Data Quality, Missing Values, Irregular Cardinality, Outliers, Handling Data Quality, Describing Data, Preparing Data Tables, Understanding Relationships, Identifying and Understanding Groups, Building Models from Data, Significance, Classical and Bayesian Analysis

Practical:

- Identification of missing values and detection of outliers
- Creation of summary table and visualization of data distribution

Univariate and Bivariate Analysis: Distributions and Types of Variables, Numerical Summaries (Level and Spread), Scaling and Standardizing, Measures of Inequality, Smoothing Time Series, Relationships Between Two Variables, Percentage Tables, Analyzing Contingency Tables.

Practical:

- Generation of Plots and application of scaling.
- Creation of a scatterplot and interpret the relationship.

Multivariate and Time Series Analysis: Multivariate Analysis, Causal Explanations, Three, Variable Contingency Tables and Beyond, Longitudinal Data, Fundamentals of Time Series Analysis (TSA), Characteristics of Time Series Data, Data Cleaning, Time, Based Indexing, Grouping, Resampling.

Practical:

- Creation of a three–variable contingency table
- Time series data analysis for clean missing or inconsistent timestamps

Data Visualization and Exploration: Visualization Objectives and Key Factors, Data Representation and Presentation, Stages of Data Visualization, Visualization Tools and Widgets, Mapping and Geographic Data, Time Series Visualization, Correlations and Scatterplots, Trees, Hierarchies, Networks

Practical:

- Visualization of dataset using multiple subplot.
- Generation of correlation heatmap and a map-based plot u.

Interactive Data Visualization: Text and Document Visualization, Levels of Text Representations, Single Document Visualizations, Document Collection Visualizations, Interaction Concepts and Techniques, Designing Effective Visualizations, Comparing and Evaluating Visualization Techniques.

Practical:

- Visualization of word frequencies from a small text dataset.
- Creation of chart using interactive visualization libraries.

Weightage: Continuous Assessment: 50%, End Semester Examinations: 50%

Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20)

References:

1. Mukhiya, S. K., & Ahmed, U. (2020). Hands-on exploratory data analysis with Python. Packt Publishing.
2. Sharma, T. (2023). Mastering exploratory data analysis with Python: Gain a solid understanding of exploratory data analysis and data visualization techniques using Python. Packt Publishing.
3. Schwabish, J. (2021). Better data visualizations: A guide for scholars, researchers, and wonks. Columbia University Press.

E-Resources:

1. https://app.datacamp.com/learn/courses/statistical-thinking-in-python-part-1
2. https://realpython.com/python-data-visualization-bokeh

	Description of CO	PO	PSO
CO1	Explain the data analysis using statistical and exploratory techniques	--	--
CO2	Apply univariate, bivariate, and multivariate analysis methods to extract insights from data.	PO1(3)	PSO1(3)
CO3	Design effective visualizations to communicate complex data patterns.	PO3(2)	PSO2(2)
CO4	Evaluate interactive visualizations for real-world applications	PO2(1)	PSO1(3)

MC25106	Advanced Java Programming	L	T	P	C
		2	0	2	3

Course Objectives:

- To provide in-depth knowledge of advanced Java concepts.
- To enable students to develop Java applications using object-oriented and event-driven programming techniques.
- To develop proficiency in using Java frameworks, APIs, and libraries for building real-world applications.

Core Java Fundamentals: Java Virtual Machine, data types, variables, keywords, operators, expressions, control statements, classes, objects, constructors, access control, method overloading, static members, Arrays, Strings, Inheritance: types, constructors in inheritance, method overriding, use of super, abstract classes – interfaces, dynamic method dispatch. Packages and exception handling

Practical:

- Write a Java program that demonstrates the use of classes, objects, constructors, method overloading, inheritance, super, and abstract classes.
- Create a Java application that includes built–in and user–defined exception handling using try, catch, finally, throw, and throws.

Multithreading, JDBC, and Web Client UI Development: Java Thread Model, Concurrent Programming, Thread Life Cycle, Thread Priorities, Creating and Managing Threads, Thread Methods and Exceptions, Inter-thread Communication, Synchronization. JDBC Architecture and Drivers, CRUD Operations – Connecting to Relational Databases. Web Client UI: HTML and CSS, Responsive Web Design using Bootstrap, JavaScript, React JS, Creating Reusable UI Components with React.

Practical:

- Implement multiple threads in Java using the Thread class and Runnable interface. Demonstrate thread lifecycle, priorities, and synchronization.
- Connect to a relational database (MySQL/PostgreSQL) and perform Create, Read, Update, and Delete operations using JDBC

Jave Servlets: Java EE Architecture - Application Servers and Containers, Servlet API and Lifecycle, Handling Client Requests and Responses - Working with Databases through Servlets-Servlet Advanced Concepts: Request Dispatcher -Session Management, Cookies and Http Session, Working with Files, Non- blocking I/O in Servlets.

Practical:

- Implement a servlet for uploading files to the server and downloading them using Multipart Config and File Output Stream
- Develop a servlet to handle HTTP GET and POST methods. Accept form data (e.g., user registration) and display it.

Java Server Pages and JSTL: JSP Architecture and Lifecycle, Action Elements, Implicit Objects, Scripting Elements, Scope and EL (Expression Language), JSP Standard Tag Libraries (JSTL), Developing Dynamic Web Applications with JSP.

Practical:

- Design a responsive UI form using Bootstrap and React to capture user input and display the result
- Create a JSP application using scripting elements, EL for data display, and JSTL for conditional and loop–based rendering.

Java Frame works: MVC Architecture, Spring Framework, Spring Modules, Spring MVC and Spring Boot, Hibernate Framework, Other Popular Frameworks. Maven Installation, Maven Core Concepts - Basic Structure of the POM File, Maven Build Life Cycles, Phases and Goals - Maven Build Profiles

Practical:

- Build a Java application using Hibernate and JPA annotations to perform CRUD operations on a Student entity.
- Mini-project integrating Spring and Hibernate framework.

Weightage: Continuous Assessment: 50%, End Semester Examinations: 50%

Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20)

References:

1. Schildt, H. (2019). Java: The complete reference (9th ed.). Tata McGraw Hill Publishing Company Limited.
2. Balagurusamy, E. (2017). Programming with Java: A primer (5th ed.). Tata McGraw Hill Publishing Company Limited.
3. Horstmann, C. S., & Cornell, G. (2013). Core Java volume I – Fundamentals (9th ed.). Pearson Education.
4. Schildt, H. (2014). Java: A beginner's guide (6th ed.). Tata McGraw Hill Publishing Company Limited.
5. Moraes, E. (2018). Java EE 8 cookbook: Build reliable applications with the most robust and mature technology for enterprise development. Packt Publishing.
6. Shah, S., & Shah, V. (2017). Java EE 7 for beginners. SPD.

E-Resources:

1. https://docs.oracle.com/javase/tutorial/
2. https://www.baeldung.com/java-thread
3. https://docs.oracle.com/javaee/7/tutorial/index.html

	Description of CO	PO	PSO
CO1	Elaborate advanced Java programming concepts to develop scalable software solutions.	--	--
CO2	Design database-driven applications using JDBC, servlets, and JSP.	PO1(3)	PSO1(3)
CO3	Develop dynamic web applications using Java frameworks and MVC architecture.	PO3(2)	PSO2(2)
CO4	Integrate modern Java tools and frameworks to build secure enterprise-level applications.	PO2(1)	PSO1(3)