Course Title: Java Programming for Application Development			
Course Code: UDS 305	**No. of credits:** 4	**L-T-P:** 30-15-30	**Learning hours:** 60
			L: Lectures**; T:** Tutorials**; P:** Practical
Pre-requisite Course Code and Title (if any): UDS 106: Database Management System, UDS 104: Fundamentals of Information Technology			
Department: Natural and Applied Sciences			
Course Coordinator: Dr. Priyanka Singh		**Course Instructor:**	
Contact Details: priyanka.singh@terisas.ac.in			
Course Type: Major		**Course Offered in:** Semester-5	

Course Description

This course is designed to provide students with a strong foundation in Java programming and equipping students with the necessary skills to develop robust, scalable, and dynamic applications. It covers core Java concepts, object-oriented programming (OOP), graphical user interfaces (GUI), database connectivity, and server-side development. Additionally, students will explore web technologies such as HTML, CSS, JavaScript, Servlets and JSP. By the end of the course, students will be able to design, develop, and deploy Java-based applications and dynamic web applications.

Course Objectives

- Understand object-oriented programming (OOP) concepts such as classes, objects, inheritance, polymorphism, abstraction, and encapsulation.
- Develop graphical user interfaces (GUIs) Applications using Java
- Understand Database Connectivity with Server-Side Programming
- Work on industry-oriented case studies and best practices

Course Content

Module	Topic	L	T	P
1	**Fundamentals of JAVA**			
	This module introduces the core concepts of Java programming, laying the foundation for object-oriented programming (OOP) and software development. Following topics are as follows: Introduction to Java - Overview of Java programming language, History and evolution of Java, Features and advantages of Java; Basic Syntax - Structure of a Java program; Data types, variables, and constants, comments and documentation; Control structures/Statements - Conditional statements (if, else, if, else); Switch-case statements, Looping statements (for, for-each, while, do-while); Object-Oriented Programming concepts: objects, classes, Inheritance, polymorphism, encapsulation, and abstraction, Constructors and Destructors.	4	2	6
2	**Exception handling, Collections and Framework**			
	This module delves into advanced Java programming concepts, focusing on modularity, exception handling, file operations, data structures, and concurrency. The major topics will include: Packages and Interfaces, Exception Handling: Understanding exceptions and errors, Try-catch blocks, Custom exception handling; File Handling: Reading and writing to files, Working with input and output streams; Collections Framework: Overview of collections (List, Set, Map), Iterators and Enumerations, ArrayList, LinkedList, HashMap, etc.;	8	3	8

	Multithreading: Introduction to threads, Synchronization and coordination, Thread safety and deadlock.			
3	**JDBC & Transaction Management**			
	This module focuses on Java Database Connectivity (JDBC), enabling students to interact with relational databases using Java. It covers the fundamentals of database connectivity, executing SQL queries, and other topics as follows: Java applets - Life cycle of an applet, adding images to an applet, adding sound to an applet, passing parameters to an applet, event handling; Introducing AWT - working with windows graphics and text, AWT controls, layout managers and menus; Introduction to Java API - working with java standard libraries, utilizing predefined classes and methods; Java Database Connectivity (JDBC) - JDBC architecture, JDBC drivers, connection pooling, handling SQL queries; Java Servlets: Servlet life cycle, Servlet containers, Servlet Configurations and Parameters.	8	5	8
4	**JAVA for Web Technologies**			
	This module provides a comprehensive introduction to web page designing using HTML, client-side scripting (JavaScript), and server-side scripting (JSP). Following topics will be covered in course delivery: Web page Designing using HTML - Client side and server-side scripting. Java Script-Object, names, literals, operators and expressions- statements and features- events- windows- documents- frames- data types- built-in functions; JavaServer Pages (JSP) - Introduction to JSP, JSP life cycle, JSP expressions and declarations, Page directives, JSP actions and implicit objects, JSP Tag Libraries; Session Management: HttpSession and Cookies; Handling Form Data- REST APIs (GET and POST methods); Model-View-Controller (MVC) Architecture, MVP, MVVM	10	5	8
	Total	**30**	**15**	**30**

Practical Sessions	Object-Oriented Programming concepts of objects and classes, Inheritance, polymorphism, encapsulation, and abstraction	-	-	6
	List, set, map, threads, deadlocks, synchronization, multithreading			8
	Java applets, AWT, Java API, JDBC – architecture, drivers, connection pooling, SQL handling, Servlets	-	-	8
	client and server-side scripting, JavaScript, JSP – basics, lifecycle, expressions, HttpSession and cookies, GET/POST, MVC architecture	-	-	8
	Total Practical Sessions	**-**	**-**	**30**

Evaluation Criteria

- Minor Test: Written test [at the end of teaching of modules 1 and 2] -- 20%
- Major Test: Written test [end of semester, full syllabus, modules 1, 2, 3, and 4] -- 30%
- Practical Test: Practical test [including modules 1, 2 and 3] -- 20%
- Project-based learning: Project presentation [at end of teaching of module 4] – 30%

Learning Outcomes

By the end of the course, students will:

- Understand Java programming competencies [Module 1 and 2; Minor Test 1]
- Able to design and develop a full-stack web application using Java, JSP/Servlets, and databases (MySQL, PostgreSQL)
  [Module 3 and 4; Minor Test 2; Practical Test]
- Gain OOPs knowledge and web technologies skills
  [Module 1, 2, 3, and 4; Minor Project; Major Test]

Pedagogical Approach

- The course will provide blended learning approach - lecture-based instruction, hands-on labs & coding assignments, flipped classroom and Project-Based Learning (PBL)
- The course is dedicated to incremental learning starting from basic to advanced progression, i.e. starting from Java basics (variables, loops, OOP) and gradually introduce complex topics like multithreading, JDBC, and web frameworks
- This course will have formative assessments (mini-projects, coding exercises), summative assessments (Mid-term and final project evaluations), and continuous evaluation (online coding platforms for auto-grading assignments).

Reading Resources (* = compulsory readings)

- * Herbert Schildt. (2021). *Java The Complete Reference*, *Twelfth Edition*. Tata McGrawHill Edition.
- C Xavier. (2011). *Java Programming - A Practical Approach*. Tata McGraw-Hill Edition
- * John Pollock. (2016). *Javascript A Beginners Guide, 3rd Edition.* Tata McGraw-Hill Edition.
- K. Arnold and J. Gosling. (2000). *The JAVA programming language, Third edition.* Pearson Education.
- R. Nageswara Rao. (2016). *Core JAVA: An Integrated Approach, Includes All Versions upto Java 8.* Dreamtech Press.

Coding Practice & Interactive Platforms

- LeetCode (Java Coding Challenges) – leetcode.com
- HackerRank (Java & Web Technologies Challenges) – hackerrank.com/domains/tutorials/10-days-of-java
- CodeGym (Java Learning Platform) – codegym.cc
- Project Euler (Java Programming Problems) – projecteuler.net

Student Responsibilities:

The students are required:

- To be regular and attentive in theory classes and lab sessions
- To come prepared with readings that would be given in the theory class
- To come prepared in lab with readings from respective theory concepts taught in class
- To sincerely participate in the day-to-day lab activities
- To incorporate real-word datasets as case-studies in project-based learning
- To submit assignments and project works on or before the deadlines

Course Designed by:

- Dr Priyanka Singh, Assistant Professor, Department of Natural and Applied Sciences, TERI School of Advanced Studies, New Delhi

Course Reviewed by:

- Dr. Satish Chand, Professor, School of Computer & Systems Sciences, Jawaharlal Nehru University, Delhi
- Dr. Abhijit Kumar, Senior Associate Professor, School of Computer Science, UPES, Uttarakhand