Pokhara University
Faculty of Science and Technology

Course No.:	Full marks: 100
Course title: **Advanced Programming with Java**	Pass marks: 45
Nature of the course: Theory/Tutorial/Practical	Time per period: 1 hour
Year:	Total periods: 45
Level: Undergraduate	Program: BE Computer/IT/Software

1. Course Description

This comprehensive course provides students with an in-depth understanding of advanced Java programming, covering essential theoretical and practical aspects. Topics include Java architecture, object-oriented principles, graphical user interface development, networking, database integration, web development, and advanced Java topics like ORM, Hibernate, and concurrency. By the end of the course, students will be proficient in Java programming, equipped to build GUI applications, handle networked and database-driven projects, and explore advanced Java concepts, making them well-prepared for diverse Java-related challenges in the professional world.

2. General Objectives

The course is designed with the following objectives:

1. To impart a thorough understanding of Java programming fundamentals, including its architecture, data types, and control structures, enabling students to build robust and efficient Java applications.
2. To delve into advanced object-oriented principles, such as inheritance, polymorphism, and abstraction, equipping students with the skills to design and implement sophisticated Java programs.
3. To develop expertise in creating graphical user interfaces (GUIs) using AWT, Swing, and JavaFX, enabling students to design interactive and visually appealing software applications.
4. To explore networking and distributed programming concepts, including socket programming, URL handling, and email integration, empowering students to develop networked Java applications.
5. To provide comprehensive knowledge of database connectivity with Java through JDBC, covering connection management, SQL operations, and security measures.
6. To introduce web development with servlets and JSP, teaching students how to create dynamic web applications with database integration, session management, and form processing.
7. To expose students to advanced Java topics, including Object-Relational Mapping (ORM), Hibernate, Spring Boot, concurrency, and design patterns, enabling them to tackle complex Java projects and applications effectively.

3. Methods of Instruction

As this course encompasses a wide range of Java programming concepts, it will adopt a diverse instructional approach to cater to various learning dimensions. The delivery will include traditional lectures that provide students with a strong theoretical foundation in advanced Java. In addition to lectures, practical classes will allow students to gain hands-on experience in Java programming, reinforcing their understanding. Tutorials will complement lectures and encourage interactive discussions to address questions and clarify complex topics. To promote knowledge sharing and exploration of emerging trends, students will engage in group discussions and presentations related to advanced Java programming. Periodic short quizzes will be conducted to gauge students' comprehension, and project work will be assigned, challenging students to create Java software applications that demonstrate their mastery of the course material.

4. Contents in detail with specific objectives

Specific Objectives	Contents
The chapter intends to provide a brief introduction of programming in JAVA and familiarize students with concepts of basics of Programming. It intends to enhance the understanding of the programming with respect to JAVA along with concepts of Access Modifiers and Java Collections.	**Unit 1: Basics of Programming in Java (7 hrs)** 1.1 Java Architecture, Class paths, Sample Program 1.2 Classes, Objects, Constructors 1.3 Packages and Data Types 1.4 Conditional Statements 1.5 Access Modifiers 1.6 Exception Handling 1.7 Java Collections
Students will learn about object-oriented principles in JAVA. Students will be able to implement the concepts of inheritance, abstraction, polymorphism. This chapter helps students learn how object-oriented concepts are implemented in JAVA.	**Unit 2: Object Oriented Principles in Java (6hrs)** 2.1 Review of object-oriented principles 2.2 Super class, sub class, inheritance, and member access 2.3 Types of inheritance 2.4 Extends and super keyword 2.5 Overriding/Overloading 2.6 Final classes and methods 2.7 Abstract classes and methods 2.8 Upcasting vs Down casting 2.9 Interfaces and Implementations

This chapter intends to provide students' knowledge on how UI components like Swing, AWT and JavaFX is used to create desktop applications along with elements, layouts and controls.	**Unit 3: Building Components using Swing and JavaFX (6 hrs.)** 3.1 Introduction to AWT and Swing: Concept, Applets, Swing Class Hierarchy, Components/Containers 3.2 Layout Management 3.3 GUI Controls 3.4 Menu Elements and Tooltips 3.5 Dialogs and Frames 3.6. Event handling and Listener Interfaces 3.7. Handling Action Events 3.8 JavaFX vs Swing 3.9 JavaFX Layouts 3.10 JavaFX UI Controls
This chapter intends to provide students basic of networking and how JAVA programming can be used to perform basic to advanced network programming. This also intends to cover the basics of Email Handling along with advanced concepts like RMI, IDL and CORBA.	**Unit 4: Distributed Network Programming (8 Hrs.)** 4.1 TCP, UDP, IP Address and Ports 4.2 Socket Programming using TCP and UDP 4.3 Working with URLs and URL Connection Class 4.4 Email Handling using Java Mail API 4.5 Architecture of RMI 4.6. Creating and Executing RMI Applications 4.7. Architecture of CORBA 4.8. RMI vs CORBA 4.9. IDL and Simple CORBA Program
This chapter intends to provide information on how Database connectivity is handled in JAVA along with the architecture, drivers, and basics of database.	**Unit 5: Database Connectivity with JAVA (5 hrs)** 5.1 JDBC Architecture 5.2 JDBC Driver Types and Configuration 5.3 Managing Connections and Statements 5.4 Result Sets and Exception Handling 5.5 DDL and DML Operations 5.6. SQL Injection and Prepared Statements 5.7. Row Sets and Transactions 5.8. SQL Escapes
This chapter intends to provide students' knowledge on how web programming is achieved in JAVA with the help of Servlets and JSP.	**Unit 6: Servlets and JSP (6 hrs.)** 6.1 Overview of Web Application 6.2 HTTP Methods and Responses 6.3 Life Cycle of Web Servlets 6.4 Writing Servlet programs with Servlet APIs 6.5 Reading and Processing Forms 6.6. Handling GET/POST Requests 6.7. Database connectivity through servlets

	6.8. Cookies and Sessions
In this chapter, students will get to know advanced topics like ORM, Frameworks, concurrency, multithreading and how it can be used in JAVA. Along with these, students will also learn basics of design patterns.	**Unit 7: Advanced Topics in JAVA (7 Hrs)** 7.1 Overview of ORM 7.2 Hibernate 7.3 Web Framework Introduction 7.4. Basics of Spring Boot 7.5. Concurrency and Multithreading in JAVA 7.6. Design Patterns: Singleton, Factory and Abstract Factory

5. List of Tutorials

The following tutorial activities of 15 hours per group of maximum 24 students should be conducted to cover all the required contents of this course.

S.N.	Tutorials
1	Setting Up Your Java Development Environment: A tutorial on installing and configuring the Java Development Kit (JDK) and Integrated Development Environment (IDE) for Java programming.
2	Creating Your First Java Program: A step-by-step guide to writing, compiling, and running a simple Java program to understand the basic structure.
3	Working with Classes and Objects: Tutorials covering the creation of classes, instantiation of objects, and implementing constructors and methods.
4	Packages and Data Types: Exploring Java packages and different data types, with hands-on exercises to practice variable declarations and data manipulation.
5	Conditional Statements: Tutorial sessions on using if-else statements, switch-case constructs, and logical operators for conditional programming.
6	Access Modifiers: Understanding access modifiers like public, private,

 7	protected, and default, and their implications on class members. Exception Handling in Java: Comprehensive tutorials on try-catch blocks, checked vs. unchecked exceptions, and best practices for handling exceptions.
8 9	Working with Files in Java: Step-by-step guidance on reading from and writing to files using Java I/O classes. Java Collections Framework: In-depth tutorials on ArrayLists, LinkedLists, HashMaps, and other collection classes, including common operations and use cases.
10	Object-Oriented Concepts: Exploring inheritance, polymorphism, encapsulation, and abstraction with practical examples.
11	Advanced Inheritance and Interfaces: Tutorials on creating subclasses, implementing interfaces, and understanding multiple inheritance in Java.
12	Graphical User Interfaces (GUIs) with Swing: A series of tutorials on building interactive user interfaces using Swing, covering components, event handling, and layout management.
13 14	Introduction to JavaFX: Learning the basics of JavaFX, including scene graphs, UI controls, and event handling for creating modern Java applications. Network Programming with Sockets: Hands-on exercises demonstrating socket programming for both TCP and UDP protocols.
15	Database Connectivity with JDBC: Step-by-step tutorials on JDBC configuration, database connection management, executing SQL queries, and handling result sets.
16	Advanced Inheritance and Interfaces: Tutorials on creating subclasses, implementing interfaces, and understanding multiple inheritance in Java.
17	Servlet Development: A comprehensive guide to creating Java servlets, understanding their life cycle, and processing HTTP requests and responses.
18	JSP (JavaServer Pages) Essentials: Tutorials on developing dynamic web pages using JSP, including form handling and database connectivity.

19	Concurrency and Multithreading: Exploring multithreading in Java, including synchronization, thread pools, and concurrent data structures.
20	Design Patterns: Detailed tutorials on the Singleton, Factory, and Abstract Factory design patterns, with practical examples.

6. Practical Work

S.N.	Practical Works
1	Basic Java Application: Create a simple Java application that demonstrates your understanding of Java's syntax, variables, and control structures.
2	Object-Oriented Programming: Develop a Java program that showcases object-oriented principles such as inheritance, encapsulation, and polymorphism.
3	GUI Application with Swing: Build a graphical user interface (GUI) application using Swing, featuring interactive components like buttons, text fields, and labels.
4	JavaFX Project: Create a JavaFX application that utilizes JavaFX layout managers, UI controls, and event handling to build a modern user interface.
5	Socket Programming: Develop a client-server application using socket programming, allowing communication between two Java applications over a network.
6 8	JDBC Database Application: Create a Java application that connects to a database using JDBC, performs CRUD (Create, Read, Update, Delete) operations, and handles exceptions. Servlet-Based Web Application: Build a web application using Java servlets to handle HTTP requests and display dynamic content on a web page using JSP a well
9	Concurrency Demonstration: Develop a Java program that illustrates the concept of multithreading, showing how threads can run concurrently and safely.

10	Hibernate Integration: Create a Java application that integrates Hibernate for Object-Relational Mapping (ORM) with a database, demonstrating CRUD operations.
11	Spring Boot Project: Develop a small Spring Boot application that showcases the use of Spring Boot's features, such as auto-configuration and dependency injection.

7. Evaluation system and students' responsibilities

Internal Evaluation

In addition to the formal end-semester exam(s), the internal (formative) evaluation of a student may consist of quizzes, assignments, lab reports, projects, class participation and presentation etc. The tabular presentation of the internal evaluation is as follows. The components may differ according to the nature of the subjects.

Internal Evaluation	**Weight**	**Marks**	**External Evaluation**	**Marks**
Theory		30	Semester-End examination	50
Attendance & Class Participation	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%			
Practical		20		
Attendance & Class Participation	20%			
Lab Report/Project Report	30%			
Practical Exam/Project Work	30%			

Viva	20%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Student requirements:
Each student must secure at least 45% marks in internal evaluation with 80% attendance in the class in order to appear in the semester-end examination. Failing to get such a score will be equated with NOT QUALIFIED (NQ) and the student will not be eligible to appear in the End-Semester examinations. Students are advised to attend all the classes and complete all the assignments within the specified time period. Failure of a student to attend a formal exam, quiz, test, etc. won't qualify him/her for re-exam. ***Students are required to complete all the requirements defined for the completion of the course***

8. Prescribed Books and References

1. Dietel H.M and Dietel P.J., Java: How to Program, Third Edition, Pearson Education Asia
2. Naughton Java 2: The Complete Reference, Tata McGraw Hill
3. Balagurusamy E., Programming in Java: 2nd Edition, Tata McGraw Hill