

Pokhara University Faculty of Management Studies

Course Code: CMP 175

Course Title: Object Oriented Programming (Java)

Nature of the Course: Theory & Practical

Level: Bachelor

Credits:3[Hrs]

Total Lectures: 48 hours

Program: BCSIT

Year: I Semester: II

1. Course Description:

This course offers a comprehensive introduction to Java programming, starting with an overview of object-oriented programming (OOP) concepts and the transition from procedure-oriented programming. Students learn the fundamentals of Java, including variables, data types, operators, control statements, and arrays. Moving on to OOP principles, the course covers classes, objects, encapsulation, constructors, static members, and inheritance. It also explores polymorphism, abstract classes, packages, and interfaces. Exception handling and stream manipulation are addressed, along with graphical user interface (GUI) programming using Swing components. Additionally, students delve into the benefits and implementation of generics for enhanced code reusability and type safety. Through hands-on exercises and projects, students develop practical Java programming skills essential for software development.

1. General Objectives:

- To understanding of fundamental programming concepts such as variables, data types, and control statements in Java.
- Develop proficiency in object-oriented programming principles including classes, objects, inheritance, and polymorphism.
- Learn to effectively handle exceptions and manipulate files in Java programs.
 - Acquire practical skills in GUI programming using Swing components and understand the benefits of using generics for code reusability and type safety.

2. Specific Objectives and Contents

Specific Objective	Contents
<ul style="list-style-type: none">• Recognize limitations of procedure-oriented programming versus object-oriented programming.• Highlight advantages and disadvantages of object-oriented programming, focusing on its key features.	Unit I: Introduction to Object-Oriented Programming (3 hrs.) 1.1 Problems in Procedure-Oriented Programming 1.2 Introduction to Object Oriented Programming, advantages and disadvantages 1.3 Features and concepts of Object-Oriented Programming 1.4 Basic Concept of OOP
<ul style="list-style-type: none">• Understand Java's features like platform independence and robustness.• Explain JVM components and Java program structure with naming conventions.	Unit II: Basic Java [9 hrs.] 2.1. Introduction to Java: Features of Java, 2.2 The Java Virtual Machine (JVM), Parts of Java program, Naming Conventions in Java

<ul style="list-style-type: none"> • Apply Java fundamentals like variables, data types, operators, arrays, and I/O operations in programming. 	2.3 Variables and constants 2.4 Data Types in Java 2.5 Operators in Java, 2.6 Reading and Displaying Output 2.7 Command Line Arguments. 2.8 Control Statements in Java 2.9 Array
<ul style="list-style-type: none"> • Create classes and objects in Java with proper instance variable initialization. • Understand and apply access specifiers and encapsulation to control class member access. • Implement constructors and grasp garbage collection concepts, including "this" pointer and static members in Java classes. 	Unit III: Object-Oriented Programming (6 hrs.) 3.1 Classes and Objects: Object Creation, Initializing the Instance Variables 3.2 Access Specifiers and Encapsulation 3.3 Constructors and Garbage collection 3.4 "this" pointer 3.5 static members
<ul style="list-style-type: none"> • Explain inheritance and its role in code reuse, covering inheritance types and implementation in Java. • Understand polymorphism principles, including method overriding, overloading, dynamic method invocation, and runtime polymorphism. • Describe abstract classes, methods, packages, interfaces, and lambda functions, demonstrating their usage in Java programming. 	Unit IV: Inheritance and Polymorphism (6 hrs.) 4.1 Inheritance and Reuse 4.2 Types of Inheritance and its Implementation 4.3 Polymorphism: Method Overriding, Method overloading, Dynamic method invocation and Run-time polymorphism 4.4 Abstract Classes: Abstract Method and Abstract Class 4.5 Packages and Interfaces 4.6 Lambda function
<ul style="list-style-type: none"> • Understand Java exceptions, including types and significance, and proficiently implement exception handling mechanisms for both built-in and user-defined exceptions in Java programs 	Unit V: Exception Handling (6 hrs.) 5.1 Introduction to Exception 5.2 Errors in Java Program 5.2 Types of Exceptions 5.3 Exception handling mechanism 5.4 Built-in Exceptions in Java 5.5 User defined exception in JAVA
<ul style="list-style-type: none"> • Understand Java streams, differentiating between byte and character streams. • Learn file management using I/O classes like File Input Stream, File Output Stream, Data Input Stream, Data Output Stream, Object Input Stream, and Object Output Stream. 	Unit VI: Stream in JAVA (6 hrs.) 6.1 Introduction to stream, byte stream and character stream 6.2 I/O class hierarchy 6.3 Manipulating file: <ul style="list-style-type: none"> ▪ File Input Stream, ▪ File Output Stream, ▪ Data Input Stream, ▪ Data Output Stream, ▪ Object Input Stream, ▪ Object Output Stream

<ul style="list-style-type: none"> • Design GUI creation with Swing components, including buttons, labels, text fields, and text areas, and gain expertise in event-driven programming for mouse and key events 	Unit VII: GUI Programming with Swing [8 hrs.] 7.1 Introduction to graphical user interface (GUI) programming in java 7.2 Using Swing components (buttons, labels, text fields, text area, etc.) 7.3 Event-driven programming and event handling: Mouse event and key event 7.4 Building Simple Interactive Applications
<ul style="list-style-type: none"> • Develop the skill of using generics in Java and demonstrate proficiency in implementing generic classes, methods, and constructors • Apply polymorphism concepts within generic classes and methods 	Unit VIII: Generics (4 hrs.) 8.1 Advantages of using Generics 8.2 Generic classes 8.3 Generic methods 8.4 Generic constructors 8.5 Polymorphism in generic

3. List of practical

1. Program to define the structure of a basic JAVA program
2. Program to define the data types, variables, operators, arrays, and control structures
3. Program to define class and constructors. Demonstrate constructors and garbage collection
4. Program to define class, methods, and objects. Demonstrate method overloading
5. Program to define inheritance and show method overriding.
6. Program to demonstrate Packages.
7. Program to demonstrate I/O operations.
8. Program to demonstrate Exception Handling
9. Program to demonstrate Layout managers
10. Implementation of generic concepts

Note: Creating a project based on Java using object-oriented concepts

4. Methods of Instruction

- Lecture
- Group discussion
- Question-answers
- Demonstration and discussion
- Presentations
- Guest lectures
- Group work/project work
- Problem solving
- Simulation
- Tutorials

5. Evaluation System and Students' Responsibilities

Evaluation System

The internal evaluation of a student may consist of assignments, attendance, term exams, lab reports and projects etc. The tabular presentation of the internal evaluation is as follows:

Internal Evaluation	Weight	Marks	External Evaluation	Marks
Theory		30	Semester End	50
Attendance & Class Participation	10%			
Assignments	20%			
Presentations/Quizzes	10%			
Internal Assessment	60%			
Practical		20		
Attendance & Class Participation	10%			
Lab Report/Project Report	20%			
Practical Exam/Project Work	40%			
Viva	30%			
Total Internal		50		
Full Marks: 50 + 50 = 100				

Students' Responsibilities

Each student must secure at least 45% marks separately in internal assessment and practical evaluation with 80% attendance in the class to appear in the semester-end examination. Failing to get such a score will be given NOT QUALIFIED (NQ) to appear the Semester-End Examinations. Students are advised to attend all the classes, formal exam, test, etc. and complete all the assignments within the specified time period. Students are required to complete all the requirements defined for the completion of the course.

Prescribed Books and References Text Books

1. Schildt, H. (2006). The Complete Reference JAVA (7th ed.). Tata McGraw
2. Balaguruswamy, E. (2009). Programming with JAVA: A Primer (3rd ed.). TATA McGrawHill Company

References

1. Liang, D. (2014). Introduction to Java Programming (Comprehensive Version) (7th ed.). Pearson
2. Malhotra, S., & Chaudhary, S. (2014). Programming in Java. Oxford University Press
3. Lowe, D., Murach, J., & Steelman, A. (2005). Murach's Beginning Java 2. SPD