OOP IN JAVA

COURSE CODE: BCA 153
YEAR/SEMESTER: I/II
WORKLOAD: 6 HOURS A WEEK

CREDIT HOURS: 3
LECTURE: 3 Hrs.
PRACTICAL: 3 Hrs.

Course description:

This course provides an in-depth introduction to object-oriented programming (OOP) principles and their implementation using the Java programming language. This course covers the different concepts of object-oriented programming such as classes, objects, inheritance, polymorphism, abstraction, encapsulation, exception handling, multithreading collections, generics, file handling and advanced concept of OOP. The course emphasizes both theoretical understanding and hands-on programming skills.

Course objectives:

The main objective of this course to provide students both the theoretical foundation and practical knowledge of programming using Java. After the completion of the course, students will be able to:

- understand OOP principles and implement them in Java.
- install java compiler and IDE ,compilation and run of java program
- design and develop Java applications using classes, objects, and interfaces.
- use exception handling, collections, and generics and file I/O in real-world programs.
- apply Advanced OOP concept (Design pattern, Lambda, Streams API) for efficient coding.
- build a small-scale Java project using OOP best practices.

Prerequisite:

Conceptual understating on programming logic and technique.

Course Contents

Unit-1 Introduction to Java and OOP concept **4Hrs.**

1.1 History, Feature or Buzzwords of Java
1.2 Java Architecture: JVM, JDK and JRE

Humanities and Social Sciences
Dean's Office
T.U

1.3 Procedural-oriented Vs. Object-oriented Programming

1.4 Setting up java environment and IDE in Local machine

1.5 Sample java programs

1.6 Compiling and running java program

1.7 Command-line arguments

1.8 Scanner class for input

1.9 Handling common errors

Unit-2 Basic of Java Programming **8Hrs.**

2.1 Writing comments and its type

2.2 Java token: keywords, identifier, literal, operators and separators

2.3 Data types: primitive and user-defined data type.

2.4 Variable declaration and assignment, expression

2.5 Control statements: selection statements, looping statement and jump statements.

2.6 Arrays: single dimension array, multi-dimensional array (Rectangular and Jagged)

2.7 Type conversion and casting

2.8 Garbage Collection

2.9 String: creation, concatenation, comparison, modification, changing case and searching

2.10 String Buffer Class

Unit-3 Class and Objects in Java **8 Hrs.**

3.1 Defining class, adding method to class, creating object and calling function/method

3.2 Abstraction and Encapsulation

3.3 Constrictors and its type (Default, Parameterized and copy)

3.4 'this' keyword

3.5 Static fields and methods

3.6 More on method: passing by value, by reference

3.7 Recursion

3.8 Nested and inner class

3.9 Variable length arguments

3.10 Package: Defining and importing package

Unit-4 Inheritance and Polymorphism **6 Hrs.**

4.1 Inheritance basics

4.2 Inheritance Type(Single-level, Multi-level, Multiple and Hierarchical)

4.3 'super' keyword

4.4 Polymorphism: Method overloading and method overriding

4.5 Object class

4.6 'final' keyword

4.7 Abstract class and methods

4.8 Access control (private, protected, default and public)

4.9 Interface: Defining, implementing and applying interface

Unit-5 Exception Handling and Multithreading **6 Hrs.**

5.1 Basic exceptions, proper use of exceptions

5.2 Exception hierarchy

5.3 Exception handling keywords: try, catch, throw, throws and finally

5.4 Java's built-in exceptions

5.5 User-defined exceptions

5.6 Multithreading basics

5.7 Thread class and Runnable interface

5.8 Thread priorities

5.9 Thread synchronization and inter-thread communication

Unit-6 File Handling in Java **6 Hrs.**

6.1 Console and File I/O

6.2 Reading and writing file using byte stream

6.3 Reading and writing file using character stream

6.4 Serialization and deserialization

6.5 RandomAccessFile class

Unit-7 Collections and Generics **6 Hrs.**

7.1 Wrapper class and associate methods

7.2 Java collection framework

7.2.1 List, Set, Map interface

7.2.2 ArrayList, LinkedList, HashSet, HashMap and TreeSet Class

7.3 Accessing collections: Iterator/comparator

7.4 Defining generic class and methods

7.5 Using wildcard arguments

7.6 Generic interface and generic hierarchy

7.7 Some generic restrictions

Unit-8 Advanced OOP concepts in Java **4 Hrs.**

8.1 Design pattern: singleton, factory, observer pattern

8.2 Lambda expression

8.3 Stream API: Introduction

8.4 Optional class

8.5 Method references

Laboratory Work **48 Hrs.**

Practical report contains: Theory, source code and output.

- To install Java SE and IDE (Notepad++, NetBeans or Eclispe) on local Machine.
- To run and compile java program and implement concept of command-line argument and scanner class.
- To implement java basic concepts: Token, control statements, arrays, type casting.
- To implement method provided by String and StringBuffer class.
- To define class, object and implement other object-oriented feature like Encapsulation, Inheritance, polymorphism and Abstraction.
- To handle built-in and user-defined exception and implement multithreading concepts.
- To create, write and read file using byte stream, character stream class and use concept of serialization, deserialization and RandomAccessFile.
- To implement java collection Framework and generic concepts.
- To implement advanced OOP concepts in java
- After end of course student must submit programming project on object-oriented concept of Java.

Required readings

Balagurusamy, E. (2024). *Programming with Java* (7th ed.). McGraw Hill.

Deitel, P. J., & Deitel, H. M. (2024). *Java how to program: An objects-natural approach* (12th ed.). Pearson.

Horstmann, C. S. (2024). *Core Java, Volume I: Fundamentals* (13th ed.). Addison-Wesley Professional.

Schildt, H., & Coward, D. (2024). *Java: The complete reference* (13th ed.). McGraw Hill.

and Social Science