Module 6 – Core Java

1. Introduction to Java Q:

What is Java?

Ans: Java is a high-level, class-based, object-oriented programming language developed by James Gosling at Sun Microsystems in 1995. It is designed to have minimal implementation dependencies and allows programs to run on any platform with a Java Virtual Machine (JVM).

What is the history of Java?

Ans: The Java project began in 1991 under the name Oak. It was renamed Java in 1995 and officially released by Sun Microsystems. Oracle later acquired Sun Microsystems in 2010. Over time, Java has evolved with regular updates, maintaining backward compatibility.

What are the key features of Java?

Ans: Platform Independent, Object-Oriented, Simple, Secure, Robust, Multithreaded, Portable, High Performance (via JIT compiler), and Distributed.

What is JVM, JRE, and JDK?

Ans: JVM executes Java bytecode; JRE contains JVM and libraries; JDK includes JRE plus development tools like the compiler (javac).

How do you set up the Java environment?

Ans: Install JDK, configure JAVA_HOME and PATH variables, and verify installation with java -version and javac -version.

What is the structure of a Java program?

Ans: A program includes package declaration, import statements, class definition, main method, and code statements.

2. Data Types, Variables, and Operators

Q: What are data types in Java?

Ans: Java provides primitive (byte, short, int, long, float, double, char, boolean) and non-primitive (String, arrays, classes, interfaces) data types.

Q: What is a variable?

Ans: A variable is a named memory location used to store data. Example: int age = 25;

Q: What are the types of operators in Java?

Ans: Arithmetic, Relational, Logical, Assignment, Unary, and Bitwise operators.

Q: What is type conversion and casting?

Ans: Type conversion (implicit) occurs automatically; type casting (explicit) requires manual conversion using syntax like (int)value.

3. Control Flow Statements

Q: What are control flow statements?

Ans: They determine the sequence in which statements are executed.

Q: What are conditional statements?

Ans: Include if, if-else, nested if, and switch-case for decision making.

Q: What are loops?

Ans: For, while, and do-while loops are used to execute blocks repeatedly.

Q: What are break and continue?

Ans: Break terminates a loop; continue skips to the next iteration.

4. Classes and Objects

Q: What is a class?

Ans: A blueprint for creating objects containing data fields and methods.

Q: What is an object?

Ans: An instance of a class that holds real values.

Q: What are constructors?

Ans: Special methods used to initialize objects.

Q: What is constructor overloading?

Ans: Having multiple constructors with different parameter lists.

Q: What is the 'this' keyword?

Ans: Refers to the current class instance, used to distinguish variables.

Q: What is encapsulation?

Ans: Wrapping data and methods together and controlling access through getters and setters.

5. Methods in Java

Q: What are methods?

Ans: Blocks of code that perform tasks, defined with parameters and return types.

Q: What is method overloading?

Ans: Defining multiple methods with the same name but different parameter lists.

Q: What are static methods and variables?

Ans: Static members belong to the class rather than individual objects.

6. OOP Concepts

Q: What are the main OOP principles?

Ans: Encapsulation, Inheritance, Polymorphism, and Abstraction.

Q: What is inheritance?

Ans: Allows one class to acquire the properties and methods of another.

Q: What is polymorphism?

Ans: The ability of an object to take many forms, achieved via method overriding or overloading.

Q: What is abstraction?

Ans: Hiding implementation details and showing only essential features.

7. Constructors and Destructors

Q: What are the types of constructors?

Ans: Default, parameterized, and copy constructors (emulated).

Q: What is constructor overloading?

Ans: Having multiple constructors with different parameter types.

Q: What is garbage collection?

Ans: Automatic memory management where the JVM reclaims unused objects.

8. Arrays and Strings

Q: What are arrays?

Ans: A collection of elements of the same data type stored in contiguous memory.

Q: What are types of arrays?

Ans: One-dimensional and multi-dimensional arrays.

Q: What are strings in Java?

Ans: Sequences of characters represented by the String class.

Q: What are StringBuffer and StringBuilder?

Ans: Mutable string-handling classes for efficient manipulation of strings.

Q: What are common string methods?

Ans: length(), charAt(), substring(), equals(), compareTo(), etc.

9. Inheritance and Polymorphism

Q: What are types of inheritance?

Ans: Single, multilevel, and hierarchical inheritance.

Q: What is method overriding?

Ans: A subclass provides a new implementation for a method defined in the parent class.

Q: What is dynamic binding?

Ans: Method call resolution occurs at runtime.

Q: What is the super keyword?

Ans: Used to refer to the parent class constructor or methods.

10. Interfaces and Abstract Classes

Q: What is an abstract class?

Ans: A class that cannot be instantiated and may contain abstract methods.

Q: What is an interface?

Ans: A reference type containing abstract methods, allowing multiple inheritance.

Q: How do you implement multiple interfaces?

Ans: By separating them with commas in the class declaration.

11. Packages and Access Modifiers

Q: What is a package?

Ans: A namespace that organizes related classes and interfaces.

Q: What are types of packages?

Ans: Built-in (java.util, java.io) and user-defined packages.

Q: What are access modifiers?

Ans: private, default, protected, and public determine visibility of members.

Q: How to import a package?

Ans: Use the 'import' statement before class definition.

12.Exception Handling

Q: What are exceptions?

Ans: Events that disrupt program execution; can be checked or unchecked.

Q: What are try, catch, finally?

Ans: Blocks used to handle exceptions and ensure cleanup operations.

Q: What are throw and throws?

Ans: throw: used to manually throw exceptions; throws: declares exceptions in method signature.

Q: What is a custom exception?

Ans: User-defined exception class extending Exception or RuntimeException.

13.Multithreading

Q: What is a thread?

Ans: A lightweight subprocess executing independently.

Q: How to create threads?

Ans: By extending Thread class or implementing Runnable interface.

Q: What is synchronization?

Ans: Ensures that only one thread accesses shared resources at a time.

Q: What are wait(), notify(), and notifyAll()?

Ans: Used for inter-thread communication to manage thread states.

14.File Handling

Q: What is file handling?

Ans: The process of reading from and writing to files using the java.io package.

Q: What are FileReader and FileWriter?

Ans: Used for character-based file reading and writing.

Q: What are BufferedReader and BufferedWriter?

Ans: Provide efficient reading/writing using buffers.

Q: What is serialization?

Ans: Converting an object into a byte stream for storage or transmission.

15. Collections Framework

Q: What is the Collections Framework?

Ans: A set of classes and interfaces for storing and manipulating data groups.

Q: What are the main interfaces?

Ans: List, Set, Map, and Queue.

Q: What are common implementations?

Ans: ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap.

Q: What are iterators?

Ans: Used to traverse collections sequentially.

16.Java I/O (Input/Output)

Q: What are streams?

Ans: Streams are sequences of data used for input and output operations.

Q: What are InputStream and OutputStream?

Ans: Used for reading and writing binary data.

Q: How to handle file I/O operations?

Ans: By using FileInputStream and FileOutputStream for byte-based data transfer.