**14-01-2025**

* **Basic concepts of Java**

**Features of Java**

1. **Platform Independence**: Java programs can run on any operating system without modification, thanks to the Java Virtual Machine (JVM).
2. **Object-Oriented Programming (OOP)**: Java follows OOP principles like inheritance, polymorphism, encapsulation, and abstraction, making code reusable and modular.
3. **Security**: Java provides a secure execution environment by using features like bytecode verification, class loaders, and a security manager.
4. **Robustness**: Java has strong memory management and exception handling mechanisms, reducing crashes and errors.
5. **Multithreading**: Java supports concurrent execution of two or more tasks, improving efficiency.
6. **High Performance**: With Just-In-Time (JIT) compilation, Java programs run faster compared to traditional interpreted languages.
7. **Portability**: Java code is compiled into bytecode, which can be executed on any platform with a JVM.
8. **Automatic Memory Management**: Java uses garbage collection to automatically manage memory, reducing the risk of memory leaks.

**JVM, JRE, and JDK**

1. **JVM (Java Virtual Machine)**: It is an abstract machine that converts Java bytecode into machine code for execution. It enables platform independence.
2. **JRE (Java Runtime Environment)**: It includes JVM and libraries necessary to run Java applications but does not contain development tools.
3. **JDK (Java Development Kit)**: It includes JRE, compiler, and other development tools required for writing, compiling, and running Java programs.

**Writing and Running a Simple Java Program**

**Write a Java Program:**

public class Test {

public static void main(String[] args) {

System.out.println("LPU is in Jalandhar");

}

}

**Compile the Program:**

* + Open the command prompt and navigate to the file location.
  + Run: javac Test.java (This compiles the program and creates a Test.class file.)

1. **Run the Program:**
   * Execute: java Test
   * Output: LPU is in Jalandhar

**Primitive and Non-Primitive Data Types**

1. **Primitive Data Types**:
   * int: Stores integer values (e.g., 10, -5, 0).
   * float: Stores floating-point numbers (e.g., 3.14, -2.5).
   * char: Stores a single character (e.g., 'A', 'b').
   * boolean: Stores true or false values.
   * double: Stores larger floating-point numbers (e.g., 3.14159).
   * long: Stores large integer values (e.g., 123456789L).
   * short: Stores small integer values (e.g., 1000).
   * byte: Stores very small integer values (e.g., 127).
2. **Non-Primitive Data Types**:
   * String: A sequence of characters (e.g., "Hello, World!").
   * Arrays: A collection of elements of the same type (e.g., int[] numbers = {1, 2, 3};).
   * Objects: Instances of user-defined classes.

**Type Casting**

1. **Implicit Casting (Widening Conversion)**:
   * Automatically converts smaller data types to larger ones.
   * Example:

int num = 10;

double convertedNum = num; // int to double

1. **Explicit Casting (Narrowing Conversion)**:
   * Manually converts larger data types to smaller ones.
   * Example:

double num = 10.5;

int convertedNum = (int) num; // double to int