**Assignment: (Core Java)**

**1 : Introduction to Java**

**Que.1 History of Java**

**Ans.1** The history of Java began in 1991 with James Gosling and a team at Sun Microsystems. Initially, the language was focused on creating a language for consumer electronics like interactive TVs. It was originally called "Oak."

**Renaming:** James Gosling and his team had to rename their project to JAVA in 1995.

**Platform Support:** It supports both console and desktop applications.

**Multisource Work**: It also supports multisource work.

**Official Launch & Rise to Prominence (1995-2000):**

Java 1.0 was officially launched by Sun Microsystems on May 23, 1995, with the slogan "Write Once, Run Anywhere (WORA)," emphasizing platform independence through the Java Virtual Machine (JVM).

**Early Adoption**:

Netscape Navigator integrated Java support, which significantly increased its visibility and the popularity of Java applets for dynamic web content.

**Key Updates:**

Java 1.1 (1997) introduced improvements like inner classes and JDBC.

Java 2 (1998) marked a significant evolution with the introduction of J2SE, J2EE, and J2ME editions.

**Que.2 Features of Java (Platform Independent, Object-Oriented, etc.)**

**1. Simple:** Java's syntax is designed to be easy to learn and understand, especially for programmers familiar with C++.

**2. Object-oriented:** Java is based on the object-oriented programming paradigm, meaning it organizes software design around objects rather than functions and logic.

**3. Interpreted:** While Java code is compiled into bytecode, this bytecode is then interpreted by the Java Virtual Machine (JVM) at runtime.

**4. Secure & Robust:** Java incorporates features like exception handling, garbage collection, and strong type checking to enhance security and create robust applications.

**5. Dynamic:** Java is capable of dynamically linking new class libraries, methods, and objects at runtime.

**6. High Performance:** Java achieves high performance through the use of Just-In-Time (JIT) compilers that convert bytecode into native machine code at runtime.

**7. Multithreading:** Java supports multithreading, allowing multiple parts of a program to execute concurrently, improving efficiency.

**8. Platform Independent:** Java's "write once, run anywhere" (WORA) principle means compiled Java bytecode can run on any platform with a compatible JVM, regardless of the underlying hardware or operating system.

**9. Portable:** Related to platform independence, Java bytecode can be easily moved between different computing environments and executed without modification.

**Que.3 Understanding JVM, JRE, and JDK**

**Ans.3 1. JVM (Java Virtual Machine)**

* The JVM (Java Virtual Machine) is a virtual machine that allows a computer to run Java programs.
* It takes Java bytecode (compiled .class files) and converts it into machine code that your computer’s operating understands.
* JVM makes Java **platform-independent** – "Write once, run anywhere."

**2.** **JRE (Java Runtime Environment)**

* The JRE is a software package that provides everything you need to run Java programs.
* It includes the JVM, important Java libraries, supporting files.
* But it does NOT include development tools like the compiler.

**3. JDK (Java Development Kit)**

* The JDK is a complete software development kit used to develop, compile, debug, and run Java programs.
* It includes the JRE and development tools like the compiler, debugger, documentation tool, etc.

**Que.4 Setting up the Java environment and IDE (e.g., Eclipse, IntelliJ)**

**Step 1: Install Java (JDK)**

* Download JDK
* Install JDK
* Set Environment Variables (Windows)
* Open System Properties → Advanced → Environment Variables.
* Under System Variables, add:
  + JAVA\_HOME = C:\Program Files\Java\jdk-21
  + Edit Path → Add %JAVA\_HOME%\bin

**Step 2: Install an IDE (for writing Java code)**

**1. Eclipse IDE**

* Download from Eclipse.
* Choose Eclipse IDE for Java Developers.
* Install and open Eclipse → Select a workspace folder.
* Create a new project: File → New → Java Project.

**2. IntelliJ IDEA (Recommended)**

* Download from [JetBrains IntelliJ IDEA](https://www.jetbrains.com/idea/download/).
* Free Community Edition is enough for Java SE.
* Install and open IntelliJ → Create New Project → Java.
* Set JDK path if not auto-detected.

**Que.5 Java Program Structure (Packages, Classes, Methods)**

**ANS.5** A Java program is typically divided into packages → classes → methods → statements.

1. Package

* A package is a namespace to organize related classes.
* Think of it like a folder for your Java files.

2. Class

* Every Java program must have at least one class.
* The file name must match the public class name.
* A class contains fields (variables) and methods (functions).

3. Main Method

* The JVM starts execution from the main() method.

public static void main(String[] args)

* public → accessible to JVM
* static → runs without creating an object
* void → returns nothing
* String[] args → command-line arguments

4. Methods

* Functions inside a class.
* Used to define behavior.

5. Statements

* Actual code that executes instructions.