WEEK 2

Introduction and software requirements to run java programs
1# What are the softwares that help to run java programs.
2# What is JDK and JRE.
3# What is eclipse IDE.
4# How to run the java program in eclipse/netbeans IDE.

1. Software Requirements for Running Java Programs

To develop and run Java programs effectively, several essential software components are required:

Core Requirements:

- **Java Development Kit (JDK)**: Essential for developing Java applications. JDK is a comprehensive software package that includes the tools and libraries needed for Java development
- **Java Runtime Environment (JRE)**: Required to execute Java programs. The JRE is suitable for users who only need to run Java programs without developing them
- **Integrated Development Environment (IDE)**: Software applications that provide comprehensive facilities for Java development

2. Understanding JDK and JRE

Java Development Kit (JDK)

The JDK is a superset of the JRE, and contains everything that is in the JRE, plus tools such as the compilers and debuggers necessary for developing applets and applications. The JDK includes:

JDK contains tools required to write Java programs, and JRE to execute them. It includes a compiler, Java application launcher, Appletviewer, etc.

Java Runtime Environment (JRE)

The JRE does not contain tools and utilities such as compilers or debuggers for developing applets and applications. The JRE is designed specifically for running Java applications.

Key Difference: JDK is the development platform, while JRE is for execution. If you're developing Java programs, you need JDK. If you're only running existing Java applications, JRE is sufficient.

3. Eclipse IDE Overview

Eclipse IDE is one of the most popular integrated development environments for Java programming. Eclipse supports Java 24 and provides the necessary tooling for development. It offers improved code folding with custom regions, enhanced instance of clean-ups, options to collapse stack frames and navigate to variable declarations.

Eclipse Advantages:

- Mature and stable platform
- Large community support
- Extensive plugin ecosystem
- Industry-standard tool used by many organizations
- Excellent performance with large projects

4. Running Java Programs in Eclipse and NetBeans IDE

Running Java Programs in Eclipse IDE:

Step 1: Installation and Setup

1. Download and install JDK from Oracle's official website
2. Download Eclipse IDE for Java Developers from eclipse.org
3. Extract and launch Eclipse
4. Configure workspace location

Step 2: Creating a Java Project

1. File → New → Java Project
2. Enter project name (e.g., "MyFirstProject")
3. Select JRE version
4. Click "Finish"

Step 3: Creating a Java Class

1. Right-click on project → New → Class
2. Enter class name (e.g., "HelloWorld")
3. Check "public static void main(String[] args)" option
4. Click "Finish"

Step 4: Writing and Running Code

1. Write your Java code in the editor
2. Save the file (Ctrl+S)
3. Right-click in editor → Run As → Java Application

4. Or use the Run button (green play icon) in toolbar
5. Output appears in Console view

Running Java Programs in NetBeans IDE:

Step 1: Installation and Setup

1. Download and install JDK
2. Download Apache NetBeans from netbeans.apache.org
3. Install and launch NetBeans
4. NetBeans automatically detects installed JDK

Step 2: Creating a Java Project

1. File → New Project
2. Select "Java" category → "Java Application"
3. Enter project name and location
4. Check "Create Main Class" option
5. Click "Finish"

Step 3: Writing and Running Code

1. NetBeans automatically creates a main class with template code
2. Edit the code in the main editor window
3. Save the file (Ctrl+S)
4. Right-click on project → Run
5. Or press F6 to run the main project
6. Or click the Run button in toolbar

IDE Comparison

NetBeans is slightly easier to use than Eclipse, so it's also a good choice for beginner developers. Both Eclipse and NetBeans are the best IDEs used in order to develop Java applications.

NetBeans Advantages:

- More user-friendly for beginners
- Native Maven integration
- Cleaner, more intuitive interface

Eclipse Advantages:

- More powerful and customizable
- Larger plugin ecosystem
- More widely used in industry