

CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY**DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY &
RESEARCH**

Department of Computer Science & Engineering

Subject Name: Java Programming**Semester: III****Subject Code: CSE201****Academic year: 2024-25****Part - I**

No.	AIM
1.	<p>Demonstration of installation steps of Java, Introduction Object Oriented Concepts comparison of Java with other object-oriented programming languages. Introduction to JDK, JRE, JVM, Javadoc, command line argument. Introduction to Eclipse or NetBeans IDE, or BlueJ and Console Programming.</p> <p><u>1. Installation of Java</u></p> <p>Steps to install Java Development Kit (JDK):</p> <ul style="list-style-type: none"><input type="checkbox"/> Download JDK:<ul style="list-style-type: none">• Go to the Oracle JDK download page: [Oracle JDK Downloads]• (https://www.oracle.com/java/technologies/javase-downloads.html).• Select the appropriate JDK version for your operating system (Windows macOS, Linux).• Download the installer package (.exe for Windows, .dmg for macOS, .tar.gz for Linux).<input type="checkbox"/> Install JDK:<ul style="list-style-type: none">• Windows: Double-click the downloaded .exe file and follow the installation instructions.

- macOS: Double-click the downloaded .dmg file, then drag and drop the JDK package icon to the Applications folder.
- Linux: Extract the downloaded .tar.gz file to a directory and follow the instructions in the README file for installation.

□ Set JAVA_HOME (Optional):

- Windows: Set the JAVA_HOME environment variable to the JDK installation directory.
- macOS/Linux: Add the JDK bin directory to your PATH and set JAVA_HOME in your shell profile (e.g., ~/.bash_profile, ~/.bashrc).

□ Verify Installation:

- Open a terminal or command prompt.
- Type `java -version` and `javac -version` to verify that Java runtime and compiler are installed correctly.

2. Introduction to Object-Oriented Concepts

- Object-oriented programming (OOP) revolves around the concept of objects, which are instances of classes. Key principles include:
 - Classes and Objects: Classes define the blueprint for objects.
 - Encapsulation: Bundling data (attributes) and methods (functions) that operate on the data within a single unit (class).
 - Inheritance: Mechanism where a new class (derived or child class) is created from an existing class (base or parent class).
 - Polymorphism: Ability of different objects to be treated as instances of the same class through method overriding and overloading.

3. Comparison of Java with Other Object-Oriented Programming Languages

- Java is often compared with languages like C++, C#, and Python in terms of syntax, features, and application domains. Key points of comparison include:
 - Syntax: Java has a C-style syntax with similarities to C++.
 - Memory Management: Java uses automatic garbage collection, unlike C++ which requires manual memory management.
 - Platform Independence: Java programs are compiled into bytecode, which can run on

any JVM, making it platform-independent.

- Libraries: Java has a rich standard library (Java API) comparable to those in C++ and C#.
- Community and Ecosystem: Java has a large developer community and extensive third-party libraries and frameworks.

4. Introduction to JDK, JRE, JVM, Javadoc, Command Line Arguments

- JDK (Java Development Kit): Includes tools for developing and running Java programs, including JRE and development tools such as javac (Java compiler).
- JRE (Java Runtime Environment): Includes JVM (Java Virtual Machine) and libraries required to run Java applications, but does not include development tools.
- JVM (Java Virtual Machine): Executes Java bytecode and provides a runtime environment for Java programs.
- Javadoc: Tool for generating API documentation from Java source code comments.
- Command Line Arguments: Parameters passed to a Java program when it is invoked from the command line.

5. Introduction to Eclipse or NetBeans IDE (Integrated Development Environment)

- Eclipse : A widely used open-source IDE for Java development, also supports other programming languages through plugins. Features include code editing, debugging, and version control integration.
- NetBeans: Another popular open-source IDE primarily for Java development, with features similar to Eclipse.

6. Introduction to BlueJ and Console Programming

- BlueJ : A lightweight IDE specifically designed for teaching and learning Java programming, providing a simplified interface and visualization tools for object-oriented concepts.
- Console Programming : Refers to writing Java programs that interact with users via text-based input and output through the console (command line interface).