PRACTICAL 1.1

Introduction to object oriented concepts :

Object-oriented programming (OOP) is a paradigm that organizes software design around data, or objects, rather than actions and logic. It emphasizes modularity and reusability, allowing for efficient code maintenance and scalability. Key principles include encapsulation, inheritance, polymorphism, and abstraction, which collectively enable clearer, more manageable software architecture and facilitate collaborative development.

Comparison of java with C++ :

1. Platform Independence: Java is platform-independent due to its bytecode compilation and JVM execution, whereas C++ code must be compiled separately for each platform.

2. Memory Management: Java uses automatic garbage collection, while C++ requires manual memory management with new and delete operators.

3. Object-Oriented Features: Java emphasizes strict object-oriented principles with features like interfaces and abstract classes, whereas C++ offers more flexibility with multiple inheritance and operator overloading.

4. Standard Libraries: Java provides a consistent and extensive standard library, whereas C++ offers a powerful but sometimes more complex standard library that gives developers greater control over optimization and performance.

Introduction to JDK :

The Java Development Kit (JDK) is a software development kit used for developing Java applications. It includes tools such as javac (Java compiler), java (Java interpreter), and javadoc (Java documentation generator). JDK also provides libraries, APIs, and documentation necessary for Java development.

Introduction to JRE :

The Java Runtime Environment (JRE) is a software package that provides the runtime environment necessary to execute Java applications. It includes the Java Virtual Machine (JVM), core libraries, and other components required to run Java bytecode. JRE does not include development tools like compilers and debuggers, which are part of the JDK (Java Development Kit).

Introduction to JVM, Javadoc and Command line argument :

1. JVM (Java Virtual Machine):

JVM is a crucial part of the Java Runtime Environment (JRE) responsible for executing Java bytecode. It abstracts hardware and operating system specifics, enabling Java programs to run identically on any platform with a compatible JVM.

2. Javadoc:

Javadoc is a tool provided by JDK for generating API documentation in HTML format from Java source code. It parses special comments (/** ... */) in the code to produce structured and easily navigable documentation for classes, methods, and fields.

3. Command Line Arguments:

Command line arguments are parameters passed to a program when it's executed from the command line or terminal. In Java, these arguments are stored as strings in the `args` parameter of the `main` method (`public static void main(String[] args)`), allowing programs to accept input or configuration from users or other programs.\

Introduction to Eclipse or NetBeans IDE :

Eclipse IDE:

Eclipse is a popular open-source integrated development environment (IDE) primarily used for Java development but supports various languages through plugins. It offers features like code editing, debugging, version control integration, and a rich ecosystem of plugins.

NetBeans IDE:

NetBeans is another open-source IDE primarily designed for Java development but also supports multiple programming languages. It provides features such as code editing, debugging, profiling, and project management tools, all integrated into a user-friendly interface.

BlueJ and Console Programming:

BlueJ:

BlueJ is an integrated development environment (IDE) specifically designed for teaching and learning Java programming. It provides a simplified interface with visual tools to interact with objects, making it ideal for beginners.

Console Programming:

Console programming involves writing programs that interact primarily through a text-based command line interface (CLI), typically used for simple input/output operations and debugging in languages like Java and C++.