Practical 1 ::

This Program is prepared by 21CE118 Shruti Sangani

Aim :: Introduction to Object Oriented Concepts, comparison of Java with other object oriented programming languages. Introduction to JDK, JRE, JVM, javadoc, command line argument

Answer ::

1) Introduction to Object Oriented Concept

Ans. :: Object-oriented programming is a programming paradigm based on the concept of "objects", which can contain data and code: data in the form of fields, and code, in the form of procedures. A feature of objects is that an object's own procedures can access and often modify the data fields of itself.

The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

Concept ::

- Class
- Object
- Polymorphism
- Inheritance
- Encapsulation
- Data Abstraction
- Dynamic Binding
- Message passing

2) Comparison of JAVA with other Object Oriented Programming Language

Ans. :: Java is one of the most popular and widely used programming language and platform. A platform is an environment that helps to develop and run programs written in any programming language. Java uses both compiler as well as interpreter

- Python is a high-level language. It fully supports object-oriented programming. Python is not a pure object-oriented language.
- Python is an interpreted language whereas Java is not an interpreted language, it is a compiled language.
- Python is a scripting language whereas JAVA is a low-level implementation language.
- Python is easy to use whereas JAVA is not as simple as Python. Programmers prefer to use python instead of Java because python contains less line of code whereas Java is just opposite to it.

C++

- Java was basically derived from C++.
- C++ is both procedural and object-oriented programming language whereas Java is a pure object-oriented language.
- Both the languages have different objectives which means it has many differences too.
- The main objective of C++ is to design a system of programming.
- Java doesn't support operator overloading but C++ does support it.
- C++ also extends the C programming language whereas Java is basically created to support network computing.

RUBY

- Ruby and Java are object-oriented languages and also they are strongly typed.
- Java is statically typed whereas Ruby is dynamically typed.
- Both languages have a different method for executing the code. Java first converts the code into machine language so that it can be understood by it and because of this Java code runs faster than Ruby's code.

3) Introduction to JDK, JRE, JVM, javadoc, command line argument

Ans. ::

JDK : The Java Development Kit (JDK) is one of three core technology packages used in Java programming, along with the JVM (Java Virtual Machine) and the JRE (Java Runtime Environment). It's important to differentiate between these three technologies, as well as understanding how they're connected:

- The JVM is the Java platform component that executes programs.
- The JRE is the on-disk part of Java that creates the JVM.
- The JDK allows developers to create Java programs that can be executed and run by the JVM and JRE.

JRE : JRE (Java Runtime Environment) is an installation package that provides an environment to only run(not develop) the java program(or application)onto your machine. JRE is only used by those who only want to run Java programs that are end-users of your system.

JVM : JVM **(**Java Virtual Machine) is a very important part of both JDK and JRE because it is contained or inbuilt in both. Whatever Java program you run using JRE or JDK goes into JVM and JVM is responsible for executing the java program line by line, hence it is also known as an interpreter.

Command Line Argument : Java command-line argument is an argument i.e. passed at the time of running the Java program. In the command line, the arguments passed from the console can be received in the java program and they can be used as input. The users can pass the arguments during the execution bypassing the command-line arguments inside the main() method. When command-line arguments are supplied to JVM, JVM wraps these and supplies them to args[]. It can be confirmed that they are wrapped up in an args array by checking the length of args using args.length.