**PRACTICAL 1**

**Aim:**

Introduction to Object Oriented Concepts, comparison of Java with other object-oriented programming languages. Introduction to JDK, JRE, JVM, javadoc, Bytecode, Compiler, Interpreter, Scripting Language, Programming Language, Hypertext Language, command line argument.

**Object Oriented Concepts**

|  |
| --- |
| **Class:** Class is a blueprint of data and functions or methods. Class does not take any space |

Class is a user defined data type like structures and unions in C.

By default, class variables are private but in case of structure it is public. in above example person is a class.  
  
**Encapsulation and Data abstraction:** Wrapping up (combing) of data and functions into a single unit is known as encapsulation. The data is not accessible to the outside world and only those functions which are wrapping in the class can access it. This insulation of the data from direct access by the program is called data hiding or information hiding.

Data abstraction refers to, providing only needed information to the outside world and hiding implementation details. For example, consider a class Complex with public functions as getReal() and getImag(). We may implement the class as an array of size 2 or as two variables. The advantage of abstractions is, we can change implementation at any point, users of Complex class wont’t be affected as out method interface remains same. Had our implementation be public, we would not have been able to change it.  
  
**Inheritance:** Inheritance is the process by which objects of one class acquire the properties of objects of another class. It supports the concept of hierarchical classification. Inheritance provides re usability. This means that we can add additional features to an existing class without modifying it.

**Polymorphism:** Polymorphism means ability to take more than one form. An operation may exhibit different behaviors in different instances. The behavior depends upon the types of data used in the operation.  
C++ supports operator overloading and function overloading.  
Operator overloading is the process of making an operator to exhibit different behaviors in different instances is known as operator overloading.  
Function overloading is using a single function name to perform different types of tasks.  
Polymorphism is extensively used in implementing inheritance.

**Dynamic Binding:** In dynamic binding, the code to be executed in response to function call is decided at runtime. C++ has virtual functions to support this.

**Message Passing:** Objects communicate with one another by sending and receiving information to each other. A message for an object is a request for execution of a procedure and therefore will invoke a function in the receiving object that generates the desired results. Message passing involves specifying the name of the object, the name of the function and the information to be sent.

**JDK:** JDK (Java Development Kit) is a bundle of software component. JDK is an implementation of either of Java SE, Java EE or Java ME.  It includes the Java Runtime Environment (JRE), an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc) and other tools needed in Java development.

**JRE:** JRE (Java Runtime Environment) contains JVM, class libraries, and other supporting files. It does not contain any development tools such as compiler, debugger, etc. Actually, JVM runs the program, and it uses the class libraries, and other supporting files provided in JRE. If you want to run any java program, you need to have JRE installed in the system

**JVM:** In java when we compile our code then output not in exe file but it is a class file and class file consist of java byte codes which are understandable by JVM (Java Virtual Machine). JVM interprets the byte code to machine code depending upon the underlying operation system and hardware combination. The JVM is called “virtual” because it provides a machine interface that does not depend on the underlying operating system and machine hardware architecture. JVM is platform dependent. It is responsible for all the things like garbage collection, array bounds checking, etc.

**What is JIT Compiler (Just- In Time Compiler)?**

JIT compiler is a component of JRE (Java Runtime Environment). It improves the performance of Java Application’s by compiling bytecode to native machine code at run time. When a Java program is run JVM launched that interprets the byte code and provides result. At run time, the JVM loads the class files, determines the semantics of each individual bytecode, and performs the appropriate computation.

**Scripting languages**

A scripting or script language is a programming language that supports scripts: programs written for a special run-time environment that automate the execution of tasks that could alternatively be executed one-by-one by a human operator. Scripting languages are often interpreted (rather than compiled).

**Programming Language**

A programming language is a formal language that specifies a set of instructions that can be used to produce various kinds of output. Programming languages generally consist of instructions for acomputer. Programming languages can be used to create programs that implement specific algorithms.

**Hypertext Language**

HTML (Hypertext Markup Language) is a text-based approach to describing how content contained within an HTML file is structured. This markup tells a web browser how to display the text, images and other forms of multimedia on a webpage

**Command line argument**

Command line argument is a parameter supplied to the program when it is invoked. Command line argument is an important concept in C programming. It is mostly used when you need to control your program from outside. Command line arguments are passed to the main() method.