*THEORY BASED STUDY MATERIALS FOR JAVA CODING – Developed by ANISOFT COMPUTER ACADEMY.*

What is a computer program?

Computers can perform a variety of tasks, however they cannot perform any of them on their own. As we know, computers have no commonsense and they cannot think. They need clear cut instructions to tell them *what to do, how to do & when to do*. A set of instructions to carry out these functions is called a *computer program.*

What is a programming language?

The communication between two parties, whether they are machines or human beings,always needs a common language or terminology. The language used in the communication of instructions to the computer is known as a programming language.

A computer program can be written using a programming language depending on the task to be performed and the knowledge of the person developing the program. There are many different programming languages available today such as C , C ++ ,JAVA,PYTHON,C# ETC.

A programming language is an artificial language designed to communicate instructions to a machine, particularly a computer. Programming languages can be used to create programs that control the behavior of a machine and/or to express algorithms.

The process of writing a program using a language is called *programming or coding*. The person who writes such instructions is called a *programmer.*

We know that natural languages such as English,Hindi or Bengali have a set of characters and use some rules known as *grammar* in framing sentences and statements. Similarly programming languages also have a set of characters and rules known as *syntax* which must be followed while writing a particular language.

*The evolution of programming languages.*

|  |  |  |
| --- | --- | --- |
| PERIOD | PROGRAMMING LANGUAGE | CHARACTERISTICS |
| 1940’s | Machine language | * Machine dependent * Faster execution * Difficult to use and understand * More prone to errors |
| 1950’s | Assembly Language | * Machine dependent * Faster execution * More prone to errors * Relatively simple to use |
| 1950 - 1970 | FORTRAN,LISP,COBOL,ALGOL 60,BASIC,APL | * High-level languages * Easy to develop and understand programs * Less prone to errors |
| 1970 - 1990 | C,C++,FORTH,PROLOG,SMALLTALK,ADA PERL,SQL. | * Very high-level languages * Easy to learn * Highly portable |
| 1990’S | JAVA,HTML,VB,PHP,XML,C# | * Internet-based languages * Object-oriented languages * More efficient * Reliable and robust |

JAVA

Java is an object-oriented programming language introduced by Sun Microsystems in the year 1995. It was originally developed in the year 1991 by James Gosling and his team. The syntax and semantics of java are somewhat similar to C ++. However it is regarded as more powerful than C++ and the other high-level languages. In the current scenario, java is the most dominant object-oriented programming language for developing web-based applications. Apart from the web-based applications java can be used to develop other types of applications such as desktop applications and embedded systems applications.

Java is a highly platform independent language because it uses the concept of just-in-time compilation. In this type of compilation, the java programs are not directly compiled into the native machine code. Instead an intermediate machine code called byte code is generated by the java compiler that can be interpreted on any platform with the help of a program called the java interpreter(JVM).

Some of the most significant characteristics of java are:

* It is a highly object-oriented and platform independent language.
* The programs written in this language are compiled and interpreted in two different phases.
* The programs written in this language are more robust and reliable.
* It is more secure as compared to other high-level programming languages because it does not allow the programmer to access the memory directly.
* It assists the programmers in managing the memory automatically with a feature called garbage collection.
* It also implements the concept of dynamic binding and threading in a better and efficient manner as compared to other object-oriented languages.

COMPILERS & INTERPRETERS  
A Compiler and Interpreter both carry out the same purpose – convert a high level language (like C, Java) instructions into the binary form which is understandable by computer hardware. They are the software used to execute the high level programs and codes to perform various tasks. Specific compilers/interpreters are designed for different high level languages. However both compiler and interpreter have the same objective but they differ in the way they accomplish their task i.e. convert high level language into machine language.  
  
COMPILER  
A [compiler](http://www.engineersgarage.com/articles/what-is-compiler-tutorial) is a piece of code that translates the high level language into machine language. When a user writes a code in a high level language such as Java and wants it to execute, a specific compiler which is designed for Java is used before it will be executed. The compiler scans the entire program first and then translates it into machine code which will be executed by the computer processor and the corresponding tasks will be performed

INTERPRETER  
Interpreters are not much different than compilers. They also convert the high level language into machine readable binary equivalents. Each time when an interpreter gets a high level language code to be executed, it converts the code into an intermediate code before converting it into the machine code. Each part of the code is interpreted and then execute separately in a sequence and an error is found in a part of the code it will stop the interpretation of the code without translating the next set of the codes.    
  
The main differences between compiler and interpreter are listed below:

* The interpreter takes one statement then translates it and executes it and then takes another statement. While the compiler translates the entire program in one go and then executes it.
* Compiler generates the error report after the translation of the entire page while an interpreter will stop the translation after it gets the first error.
* Compiler takes a larger amount of time in analyzing and processing the high level language code comparatively interpreter takes lesser time in the same process.
* Besides the processing and analyzing time the overall execution time of a code is faster for compiler relative to the interpreter.

How does JAVA work?

Diagram showing java compilation process

**JRE and JDK**

Sun Microsystems provides two principal software products in the Java Platform.

Java SE Runtime Environment (JRE)

The JRE provides the libraries, Java virtual machine, and other components necessary for you to run applets and applications written in the Java programming language. This runtime environment can be redistributed with applications to make them free-standing.

Java SE Development Kit (JDK)

The JDK includes the JRE plus command-line development tools such as compilers and debuggers that are necessary or useful for developing applets and applications.

**Java Virtual Machine**

The Java virtual machine is an abstract computing machine that has an instruction set and manipulates memory at run time. The Java virtual machine is ported to different platforms to provide hardware- and operating system-independence.

When we compile a java file output is not an .exe but a .class file. It consists of java byte code which is understandable by JVM.  
JVM interprets the byte code into the machine code depending on the underlying OS and hardware. The JVM is called ‘virtual’ because it provides a machine interface between the java byte code and the underlying OS and hardware. This independence from hardware and OS is called “Platform Independence” or WORA (Write Once Run Anywhere) feature of java.  
Programs written in java are compiled by java compiler (java) and it generates a byte code.JVM converts the byte code into the machine code for the computer.

**Two most important concepts in JAVA (CLASS & OBJECTS)**What is an OBJECT?

In real world, we come across various objects in our day to day life such as a car,a pen,a phone and many others. An object is an identifiable entity with some characteristics and behavior of its own.

Example:-

A car is an object – It has many characteristics such as: -  
 1) Model  
 2) Color  
 3) Engine type  
 - It also has some behaviors such as :-   
 1) Start / Stop  
 2) Increase/Decrease Speed  
 3) Change gear

A software object is not much different from a real world object. A software object has characteristics or state which is represented by *data members* / fields (variables) and behavior /functions represented by *methods.*  
OBJECTS are basic run-time entities in an object oriented system. The fundamental idea behind an object oriented language is to combine both data members (variables) and the member functions (method) that operate on the data into a single unit known as object.

Diagram Showing Objects

What is a CLASS?

Class is the central point of OOP and that contains the data and code with behavior. In java everything happens within the class and it describes a set of objects with common behavior. An object is an instance of a class.  
A class is a blueprint from which individual objects can be created. A class is a group of objects that share common properties, relationships and behaviors.

***OBJECT 1***

|  |
| --- |
| **ORANGE** |
| *COLOR* |
| *FLAVOUR* |
| *ORIGIN* |

Example of CLASS:-

|  |
| --- |
| **FRUIT** |
| *COLOR*  *DATA* |
| *FLAVOUR* |
| *ORIGIN* |

*ORANGE*

***CLASS***

|  |
| --- |
| **MANGO** |
| *COLOR* |
| *FLAVOUR* |
| *ORIGIN* |

*INDIA*

*SWEET*

*GREEN*

*SPAIN*

*SOUR*

***OBJECT 2***