Introduction to Java

Java

Java programming language was originally developed by Sun Microsystems.

Java is guaranteed to be Write Once, Run Anywhere.

Java

Object Oriented − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.

Platform Independent − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

Simple − Java is designed to be easy to learn. If you understand the basic concept of OOP, it would be easy to master.

Secure − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

Java

Robust − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.

Multithreaded − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.

High Performance − With the use of Just-In-Time compilers, Java enables high performance.

Dynamic − Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

JVM

JVM is short for Java Virtual Machine.

It is a platform-independent execution environment that converts Java bytecode into machine language and executes it.

Most programming languages compile source code directly into machine code that is designed to run on a specific microprocessor architecture or operating system, such as Windows or UNIX.

Java is compile once and run anywhere language.

OOP

The object-oriented is a programming paradigm where the program logic and data are weaved.

OOP languages permit higher level of abstraction for solving real-life problems. The traditional procedural language (such as C and Pascal) forces you to think in terms of the structure of the computer (e.g. memory bits and bytes, array, decision, loop) rather than thinking in terms of the problem you are trying to solve. The OOP languages (such as Java, C++ and C#) let you think in the problem space, and use software objects to represent and abstract entities of the problem space to solve the problem.

Classes and Objects

What is an Object?

Any entity is an Object either in real world or in programming world

Then what is Class?

Blue print of a particular type of Object

Program Struct

```java
public class BankAccount
{
        private int AccountNumber;
        private boolean Active;
        private double Balance;

        public BankAccount(int AccountNumber, double Balance, boolean Active)
        {
                this.AccountNumber = AccountNumber;
                this.Balance = Balance;
                this.Active = Active;
        }

        public void Withdraw(double amount)
        {
                Balance = Balance - amount;
        }

        public void Deposit(double amount)
        {
                Balance = Balance + amount;
        }

        public void getBalance()
        {
                System.but.println("Balance is "+Balance);
        }

}

public class Test
{
        public static void main(String args[])
        {
                BankAccount myAccount = new BankAccount(12345, 125.50, true);
                myAccount.Deposit(500);
                myAccount.getBalance();
        }
}
```