How Java works

How Java works

- java’s platform is achieved by the use of the java virtual machine
- consist of one or more files with a .java extension
  - these are plain old text files
- when the program is complied the .java files are fed to the complier which produces a .class file got each .java file
- the .class file contains Java bytecode
- bytecode is like machine language, but it is intended for the Java VM not a specific chip such as a Pentium or PowerPC chip
- to run a java program the bytecode in a .class file is fed to an interpreter which converts the byte code to machine code for a specific chip (IA-32, PowerPC)
- some people refer to the interpreter as the “Java Virtual Machine” (JVM)
- the interpreter is a platform specific bc it takes platform independent bytecode and produces machine language instructors for a particular chip
- java program could be run an any type of computer that has a JVM written for it
  - pc, mac, unix, linux

Object oriented programming

program paradigm based on the concepts of “objects” which can contain data and code: data in the form of fields (often known as attributes or properties)

- classes- blueprint of an object - contains attributes, methods, and initializes values an object should have
- attributes- properties of an object. ex: Color, Size, BuildingMaterialsUsed, Cost, RustProof, WaterProof, RequiredAccessories,

- methods- equivalent to a function in a non OOP methodology. has instructions to change the contents of attributes to do a particular task in a class
- encapsulation- mechanism of wrapping the data and the code acting on the data as a single unit. the data of a class will be hidden from other classes and can be accessed only through the methods of their current class/
- abstraction

java exercise below

1. Come up with class names and member variables and member methods for a car sales man. This should help the sales man to pull up features in features in different models of cars , various colors availiable, the base sale price

A plan on how you will implement:

***create as many classes as possible-

class Vehicle {

truck,

suv,

van,

sedan,

motorcycle,

}

class Model {

}

class EngineType {

}

class New {

}

class Used {

}

class Price {

}

```
// 1,1,2,3,5,8,13....
// GenerateFibSeries(nth)

class Naturalnumbers{
  void generatenumbers(int arg1){
    for(int i = 0 ;i<=arg1;i++)
      System.out.print(i);
    System.out.println("End of Natural numbers");
  }
}
public class HelloWorld {
public static void main(String[] args) {

    Fibonacci obj1 = new Fibonacci();
    Naturalnumbers obj2 = new Naturalnumbers();
    System.out.print("Printing fibonacci series");
    obj1.generateFibonacciSeries();
    System.out.print("Printing natural numbers");
    obj2.generatenumbers(100);

    //obj1.i = 10;


  }

}

class Fibonacci {

  private int j = 1;
  private int i =1;
  private int k = i + j;
  void generateFibonacciSeries(){
    System.out.print(i + "," + j + ","+ k + ",");
    for (int n = 1; n<=25 ;n++) {
      i=j;
      j=k;
      k=i+j;
      System.out.print(k + "," );

    }
    System.out.println("End of Fibonacci");
```

```
  }

}
```

mr. k email

naganandk@sprucebots.com

connect with mr.k on linkedin