1. Consider the following code snippet.

if (aNumber >= 0)

if (aNumber == 0)

System.out.println("first string");

else System.out.println("second string");

System.out.println("third string");

* 1. What output do you think the code will produce if aNumber is 3?

Ans: second string

Third string

* 1. Write a test program containing the previous code snippet; make aNumber 3. What is the output of the program? Is it what you predicted? Explain why the output is what it is; in other words, what is the control flow for the code snippet?

Ans:

anumber is 3 hence the first if statement is true anumber>=0 hence it goes to second if statement then anumber is not equal to zero then the condition fails goes To else statement and the print the second string.Then the final println statement Is outside the first if statement hence it is also gets printed.

* 1. Using only spaces and line breaks, reformat the code snippet to make the control flow easier to understand.

Ans: if (aNumber >= 0)

if (aNumber == 0)

System.out.println("first string");

else

System.out.println("second string");

System.out.println("third string");

* 1. Use braces, { and }, to further clarify the code.

Ans: if (aNumber >= 0){

if (aNumber == 0)

{

System.out.println("first string");

}

Else{

System.out.println("second string");

}

}

System.out.println("third string”);

1. Which of the following is not OOPS concept in Java?  
   a) Inheritance  
   b) Encapsulation  
   c) Polymorphism  
   d) Compilation

Ans: d

Compilation is not oops concept in java. Where inheritance, encapsulation, polymorphism are of oops concept.

1. What is encapsulation? Please explain your answer with some example.

Ans:

Encapsulation can be achieved by Declaring all the variables in the class as private and writing public methods in the class to set and get the values of variables.

Encapsulation is used to hide the values or state of a structured data object inside a class, preventing unauthorized parties' direct access to them.

Car.java:

public class Car {  
 private int doors;  
 private int wheels;  
 private String model;  
 private String engine;  
 public void setmodel(String model) {  
 String valid = model.toLowerCase();  
 if (valid.equals("benz") || valid.equals("audi")) {  
 this.model = model;  
 } else {  
 this.model = "unkown";  
  
 }  
 }  
 public String getModel(){  
 return this.model;  
 }  
 }

Main.java:

public class Main {  
 public static void main(String[] args){  
 Car benz=new Car();  
 Car audi=new Car();  
 benz.setmodel("benz");  
 System.*out*.println("the car is"+ " "+benz.getModel());

}

}

Here the fields declared in class car can’t be accessed directly

Because they are declared private this is known as encapsulation

To hide the the data and not giving the access directly.here we use

Get and set methods to access the fields and methods present in the

Car class. Hence the in main class we create a object so that we can

Access The fields and the methods inside the car class by using get

Set methods we use benz.setmodel(“987”) then if condtion fails and   
 print model car is unknown.

1. What is Polymorphism? Please explain your answer with some example.

Ans:

The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form.

They are 2types of polymorphism

1. complie time polymorphism

achieved by method overloading.

class MultiplyFun {

     static int Multiply(int a, int b)

     {

         return a \* b;

     }

     static double Multiply(double a, double b)

     {

         return a \* b;

     }

}

class Main {

     public static void main(String[] args)

     {

         System.out.println(MultiplyFun.Multiply(2, 4));

         System.out.println(MultiplyFun.Multiply(5.5, 6.3));

     }

}

B)Run time polymorphism(achieved by method overriding)

class Parent {

     void Print()

     {

          System.out.println("parent class");

     }

}

class subclass1 extends Parent {

     void Print()

     {

         System.out.println("subclass1");

     }

}

class subclass2 extends Parent {

     void Print()

     {

         System.out.println("subclass2");

     }

}

class TestPolymorphism3 {

     public static void main(String[] args)

     {

         Parent a;

      a = new subclass1();

         a.Print();

         a = new subclass2();

         a.Print();

     }

}

1. Write a class like real world object you observed (Ex: Car, Mobile Phone etc) and also create a sample interface that defines its behaviour, then require your class to implement it. Omit one or two method from the class and note what kind of compilation error you can see.

Ans:

Here we implement the interface we should implement all the methods of interface.

Otherwise it gives the error or you can declare a abstract or we should implement all

The methods.

Interface Car{

Public void model();

Public void engine();

}

Class bike implements Car{

Public void model(){

}

Public void engine(){

}

}