# JAVA PROGRAMMING ASSIGNMENT – 3

***PART A – THEORY QUESTIONS***

**1 Explain single, multilevel, and hierarchical inheritance.?**

- Inheritance is the main concept in the java in that the super class / parent class behavior / methos , function , are replicate in to the other class / subclass / child class is class the INHERITANCE.

- SINGLE INHERITANCE in that a single class can inheritance for base class or parent class thy can get method and property, function from the based super parent class,

  The best example is the parent have one child and this child can get life lessons from thare parents.

  Suppose the vehicle is the class and car is the sub class as a result care inherent to vehicles car has all property which has in vehicles (colour , wheels,etc).

- MULTILEVEL INHERITANCE in that one base class is its has one sub class that one sub class also has a one sub class and thy have property like based class it called multilevel Inheritance.

  Ex. Is grandfather => father => son

  Vechicle is the class and car is the a sub class and sport car is also a class then the sport car also inheritance to vechicle calss and share the property or method functions.

- HIERACHICAL INHERITANCE in that one base calss have has multiple sub class and thay can belong to the same based class it call hierarchical inheritance.

  Ex.care => father => son 1 =>son2.

**2 What is polymorphism?**

- the one methode name but thay have different differently behavior for object its called polymorphism and java support the polymorphism.

  The same method call can perform different actions for different objects.

Polymorphism in Java is the ability of an object to take multiple forms, where the same method call invokes different implementations at runtime.

**3 Difference between compile-time and runtime polymorphism.**

- ⇨ Compile time Polymorphism Alos called static Polymorphism

  Compile time Polymorphism is slow as compare to runtime

  It is less flexible

  Its achive while using methpde overloding

  Decision is mad at the compile time

- ⇨ **Run**time Polymorphism it also called dynamic Polymorphism

  Runtime Polymorphism is a fast as compre to compile time

  It more flexible

  It achibve while using methode overriding

  Decision is mad at the run time

4. What is method overloading?

- ⇨ Multiple methods have same name and different parametrs in same class it called methode overloding
- ⇨ And its decide when at the compile time depend on its parameters

**5. What is method overriding?**

- ⇨ Methode overring is thet the child class has its own implementation which is already defin in the parent class its called methode overriding.

## ***PART B – PROGRAMMING QUESTIONS***

1 Write a program to find the sum of digits of a number.?

```
package demo;
import java.util.*;
public class classwork{
        public static void main(String[] args){
                Scanner sc = new Scanner (System.in);


                 int temp,n,rev=0;
                 System.out.println("=====ENTER THE NUMBER=====");
                int number= sc.nextInt();


                 int sum = 0;


            while (number!= 0) {
              sum = sum + (number % 10);
              number = number / 10;
            }


            System.out.println("=====Sum of digits===== " + sum);
                      }
                    }
```

**2. Write a program to check if a number is prime.**

```
import java.util.Scanner;

public class classwork{

  public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

 System.out.print("=====Enter a number=====");

    int num = sc.nextInt();


    boolean isPrime = true;


    if (num <= 1) {

      isPrime = false;

    } else {

      for (int i = 2; i <= num / 2; i++) {

        if (num % i == 0) {

          isPrime = false;

          break;

        }

      }

    }


    if (isPrime) {

      System.out.println(num + "=====is a Prime number=====");

    } else {

      System.out.println(num + "=====is NOT a Prime number======");

    }

 }  }
```

**3 Write a program to print all prime numbers between 1 and 100.**

```
        Import java.util.scanner;

        public class classwork {

       public static void main(String[] args) {

        for (int num = 2; num <= 100; num++) {

      boolean isPrime = true;


      for (int i = 2; i <= num / 2; i++) {

         if (num % i == 0) {

            isPrime = false;

            break;

         }

      }


      if (isPrime) {

         System.out.print(num + " ");

      }

    }

  }

}
```

**4 Write a program to print multiplication table of a number.**

```
package demo;

import java.util.*;

public class classwork{

        public static void main(String[] args){

                Scanner sc = new Scanner (System.in);


                System.out.print("=====Enter a number===== ");
```

```
                       int a = sc.nextInt();


                       System.out.println("=====Multiplication table of=====");


                       for (int i = 1; i <= 10; i++) {
                           System.out.println(a + " x " + i + " = " + (a * i));
                       }
                   }
               }
```

**5 Write a program to count the number of digits in a number.**

```
import java.util.Scanner;
public class classwork {
    public static void main(String[] args) {


        Scanner sc = new Scanner(System.in);


        System.out.print("Enter a number: ");
        int num = sc.nextInt();


        int count = 0;


        if (num == 0) {
            count = 1;
        } else {
            while (num != 0) {
                num = num / 10;
                count++;
            }
        }

        System.out.println("Number of digits = " + count);

        sc.close();
    }
}
```