ASSIGNMENT 23

Q1)Write a program of static keyword with all possible ways(variable,method,class,block)

```
package SaticDemo;

import java.sql.*;
public class StaticClass {

    //this class is working on database part
    static class Database {

        //this variable hold the database password

        static String DBpassword = "root";

        //this static block make a database connection
        static {
            System.out.println("Assume that Connection is created inside static block So Connection is created
Done....");

        }

        //this method return connection Class Object

        public static String DatabaseConnection()
        {
            return "Connection Object";
        }
    }

    public static void main(String[] args) {
     StaticClass s = new StaticClass();
        System.out.println(Database.DatabaseConnection());
    }
}
```

```
Assume that Connection is created inside static block So Connection is created Done....
Connection Object
```

Q2)Explore storage of static methods and static variable in java

Before the java 8 version , static variables of the class were stored in the separate section of the non-heap memory name as a memory area created by jvm after compilation of the class Method area section was used to store static variable of the class,meta data of the class etc.Wheras, non static methods are variables were stored in the heap memory. After java 8 version static variables are stored in the heap memory

Q3)Prove pratically=>

i)Can we Overload static methods?

```
package SaticDemo;

public class OverloadStatic {
   //static method without parameter
   static void staticVariableHoBe()
   {
      System.out.println("Hello i am static method without parameter");
   }
   //static method with one parameter and return type is string
   static String staticVariableHoBe(int a)
   {
      System.out.println("Hello i am static method with one parameter");
      return "Kya Haal hai Bhai";
   }
   public static void main(String[] args) {
      OverloadStatic.staticVariableHoBe();
      System.out.println(OverloadStatic.staticVariableHoBe(12));
   }
}
```

```
Hello i am static method without parameter
Hello i am static method with one parameter
Kya Haal hai Bhai

Process finished with exit code 0
```

ii)Can we Override static methods?

We can't override static method because static method is call through class name for better understanding check the example given below

```
package SaticDemo;
class Parent
{
   static void staticVariableHoBe()
   {
      System.out.println("Hello i am static method of parent class");
   }
}
public class OverloadStatic extends Parent{


   //static method with one parameter and return type is string
   static void staticVariableHoBe()
   {
      System.out.println("Hello i am static method of child class");
   }
   public static void main(String[] args) {
  OverloadStatic.staticVariableHoBe();
  Parent.staticVariableHoBe();
   }
}
```

```
Hello i am static method of child class
Hello i am static method of parent class

Process finished with exit code 0
```

iii)Why main method is declared as static?

Main method is static so that complier can call it without the creation of an Object of the class.

4)is There any error in the below code snippet ?if yes identify the error and give the reason behind it.

```
class Demo
{
  void m1(Demo  d)
  {
    System.out.println("instance method");
  }
  static void  m1(Demo  d1)
  {
    System.out.println("Static  method");
  }
}
```

so first we can overload static and non static method but we have to pass different parameter

5)Will the following code snippet compile fine? if yes what will be the output after excution

```
package SaticDemo;

public class OverloadStatic {
  private static int x=10;
  static
  {
    x++;//post increment  first initialized  then increment  so value is still 10 // runs only once
  }
  static
  {
   ++x; //pre increment  first increment than initialized  so value will  be 12 runs only once
  }
   {
     x--;// post decrement and instance block will  execute for all  object so it will  execute three time  so the value  is 9
   }
   public static void main(String[]  args) {
    OverloadStatic o1=new OverloadStatic();
      OverloadStatic o2=new OverloadStatic();
      OverloadStatic o3=new OverloadStatic();
      System.out.println(x);// 9
   }
}
```

```
9

Process finished with exit code 0
```