Q.1 Write a java program to multiply two given matrices.

Program:

```
public static void main(String[] args) {

    int[][] matrix1={{2,3,5},{1,1,2},{3,5,4}};
    int[][] matrix2={{1,2,4},{2,3,5},{1,4,3}};
    int matrix3[][]=new int[3][3];
    int i,j,k;
    System.out.println("multiply of matrices:");
    for (i=0;i<3;i++)   {

      for(j=0;j<3;j++)   {

      matrix3[i][j]=0;

      for(k=0;k<3;k++)  {

        matrix3[i][j]=matrix3[i][j]+matrix1[i][k]*matrix2[k][j];
      }

      }   }
    for(i=0;i<3;i++)
    {
      for(j=0;j<3;j++){
        System.out.print(matrix3[i][j]+" ");
      }
      System.out.println(" ");
    }

     }   }
```

Output:

multiply of matrices:

```
13   33   38
5    13   15
17   37   49
```

Q.2 Write a java program to implement.

a) Method overloading and constructors overloading .

Program:

```
//constructor overloading:
public class Myinfo {
  String  name;
   int  age;
   boolean  isadult;
   Myinfo(){
      name="jayant";
      age=23;
      isadult=true;
  }
   Myinfo(String name,int age){
      this.name=name;
      this.age=age;
      isadult=false;
   }
   Myinfo(String name,int age,boolean isadult){
      this.name=name;
      this.age=age;
      this.isadult=isadult;
   }
   public  void getdetails(){
      System.out.println("name : "+name);
      System.out.println("age : "+age);
      System.out.println("Isadult : "+isadult);
   }
   public static void main(String[] args) {
      Myinfo person1=new Myinfo();
      Myinfo person2=new Myinfo("imran",22,true);
```

```
        Myinfo person3=new Myinfo("pankaj",17);
        System.out.println("person1:");
        person1.getdetails();
        System.out.println("person2:");
        person2.getdetails();
        System.out.println("person3:");
        person3.getdetails();
    }   }
```

Output:

person1:

name : jayant

age : 23

Isadult : true

person2:

name : imran

age : 22

Isadult : true

person3:

name : pankaj

age : 17

Isadult : false

//method overloading:

Program:

```
public class Area {
   float getarea(float radius){
     return 3.14f*(radius*radius)   }

   int getarea(int side){
        return (side*side);   }

   double getarea(int base,int height){
        return 0.5*base*height;}

  public static void main(String[] args) {
     // TODO code application logic here
```

```
        Area obj = new Area();

        System.out.println("Area of circle: "+ obj.getarea(2.7f));
        System.out.println("Area of square: "+ obj.getarea(5));
        System.out.println("Area of triangle: "+ obj.getarea(4,6));
    }}
```

Output:

Area of circle: 22.890602
Area of square: 25
Area of triangle: 12.0

b) Write a java program to implement method overriding.

Program:

```
class A {

    void print(){

        System.out.println("inside class A");

    }   }

class B extends A  {

    @Override

    void print() {

        System.out.println("inside class B");

    }   }

public class MethodOverriding {

     public static void main(String[] args) {

     A a = new A();

     B b= new B();

     a.print();

     b.print();     }   }
```

Output:

inside class A

inside class B

Q.3 Write a java program to check whether a given string is a palindrome.

Program:

```
import java.util.Scanner;
public class Palindrom {
   public static void main(String[] args) {
      String str,reverse="" ;
      Scanner obj = new Scanner(System.in);
      System.out.println("enter string: ");
      str = obj.nextLine();
      int len = str.length();
      for (int i=len-1;i>=0;i--){
         reverse=reverse+str.charAt(i);
      }
      if(str.equals(reverse)){
         System.out.println("string is palindrom !");
      }
      else{
        System.out.println("string is not palindrom")    }
   }
}
```

Output:

```
enter string:
wow
string is palindrom !
```

Q.4 Write a java program for sorting a given list of names in ascending order .

Program:

```
import java.util.Arrays;


class NameSorter{
public static void main(String args[]){
        String[] names = {"sumit","imran","sandhya","pankaj","jayant"};
        System.out.println("sorted name:");
        for(String name:names)
        {
                System.out.println(name);
        }
                }
}
```

Output:

```
sorted name:
imran
jayant
pankaj
sandhya
sumit
```

Q.5 Write a java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given Shape.

Progam:

```
abstract class  Shape {
   int x =5,y=12;
    abstract void  printArea();  }
 class Circle extends Shape {
    void printArea(){
      System.out.println("area of circle is:"+3.14*x*x);
   } }
class Rectangle extends Shape {
    void printArea(){
      System.out.println("area of Ractangle:"+ x*y);
   } }
class Triangle extends Shape{
    void printArea(){
       System.out.println("area of triangle:"+0.5*x*y)    }  }
public class Abstractclass {
     public static void main(String[] args) {
       Circle circle = new Circle();
       Rectangle rectangle = new Rectangle();
      Triangle triangle = new Triangle();
```

```
    circle.printArea();

    rectangle.printArea();

    triangle.printArea()    }   }
```

Output:

area of circle is: 78.5

area of Ractangle: 60

area of triangle: 30.0

Q.6 Write a Java program to show uses of Vector class.

Program:

```
import java.util.Vector;
public class VectorUse {
public static void main (String args[] ) {
//create a vector of string
Vector<String> vector = new Vector <>();
vector.add("Apple ");
vector.add("Banana ");
vector.add("Cherry ");
vector.add("Orange ");
vector.add("Watermelon");
System.out.println("Elements in the Vector: ");
for (String fruit: vector){
System.out.println(fruit);
}}}
```

Output:

Elements in the Vector: Apple

Banana

Cherry

Orange

Watermelon