OOPS

LAB-10

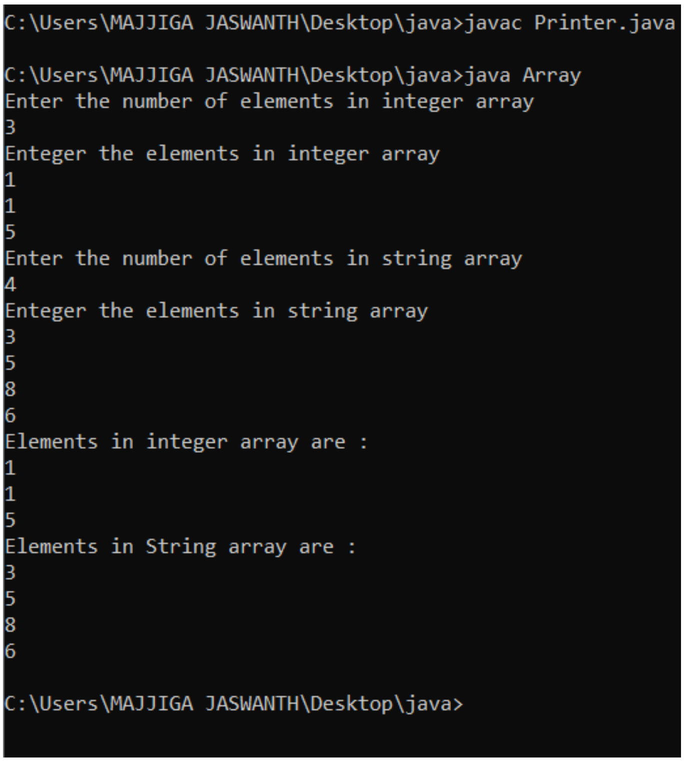
Q1)

Let's say you have an integer array and a string array. You have to write a single method printArray that can print all the elements of both arrays. The method should be able to accept both integer arrays or string arrays.

Code:

```
import java.util.*;

class printer{

public <T> void printArray(T[] arr){

for(int i = 0; i<arr.length-1;i++){

System.out.println(arr[i]);


}

System.out.println(arr[arr.length-1]);

}

}

class Array{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

printer p = new printer();

System.out.println("Enter the number of elements in integer array");

int n = sc.nextInt();

Integer intArray[] = new Integer[n];

System.out.println("Enteger the elements in integer array");

for(int j = 0;j<n;j++){

intArray[j] = sc.nextInt();

}
System.out.println("Enter the number of elements in string array");

int N = sc.nextInt();

String strArray[] = new String[N];

System.out.println("Enteger the elements in string array");

for(int k = 0;k<N;k++){

strArray[k] = sc.next();

}

System.out.println("Elements in integer array are :");

p.printArray(intArray);

System.out.println("Elements in String array are :");

p.printArray(strArray);


}

}
```

Output:



```
C:\Users\MAJJIGA JASWANTH\Desktop\java>javac Printer.java

C:\Users\MAJJIGA JASWANTH\Desktop\java>java Array
Enter the number of elements in integer array
3
Enteger the elements in integer array
1
1
5
Enter the number of elements in string array
4
Enteger the elements in string array
3
5
8
6
Elements in integer array are :
1
1
5
Elements in String array are :
3
5
8
6

C:\Users\MAJJIGA JASWANTH\Desktop\java>
```

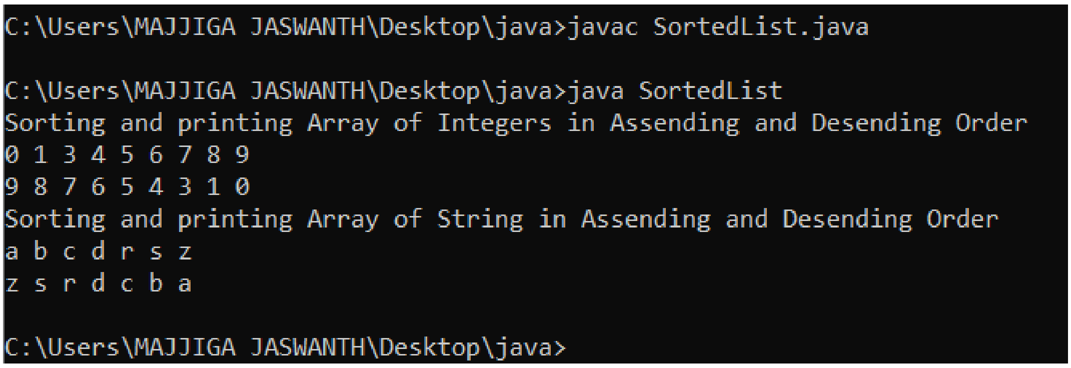
Q2) Develop a Generic SortedList class to sort the array of objects (either Numeric or String) and to print the sorted list in forward and reverse order.

Code:

```
import java.util.*;


public class SortedList{


     public static void main(String[] args)
     {


          Integer arr[] = { 3, 4, 0, 8, 6,9,7,5,1};



          Cl<Integer> obj1 = new Cl<Integer>(arr);
        System.out.println("Sorting and printing Array of
Integers in Assending and Desending Order");
        obj1.assending();
        obj1.desending();




          String arr2[] = {"a","b","c","d","z","r","s"};
          Cl<String> obj2 = new Cl<String>(arr2);
           System.out.println("Sorting and printing Array of
String in Assending and Desending Order");
           obj2.assending();
        obj2.desending();


     }
}
```

```
class Cl<T>  {

     T[] values;



     Cl(T[] obj)
     {
      values = obj;
     }




     public void assending()
     {
        Cl<T>a=new Cl<T>(values);
        Arrays.sort(values);
         for(int i=0;i<values.length;i++)
     {
         System.out.print(values[i]+" ");
     }
         System.out.println();
     }

     public void desending()
     {
          Cl<T>a=new Cl<T>(values);
        Arrays.sort(values);
         for(int i=values.length-1;i>=0;i--)
     {
         System.out.print(values[i]+" ");
     }
      System.out.println();
     }




}
```

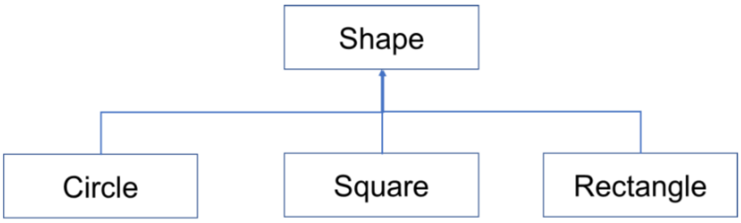
Output:



```
C:\Users\MAJJIGA JASWANTH\Desktop\java>javac SortedList.java

C:\Users\MAJJIGA JASWANTH\Desktop\java>java SortedList
Sorting and printing Array of Integers in Assending and Desending Order
0 1 3 4 5 6 7 8 9
9 8 7 6 5 4 3 1 0
Sorting and printing Array of String in Assending and Desending Order
a b c d r s z
z s r d c b a

C:\Users\MAJJIGA JASWANTH\Desktop\java>
```

Q3)

Create following class hierarchy:



Develop a Generic class “Node” to work with different shapes, like circle, rectangle, square. Nodeclass contains,

–an attribute area, which is to be initialized with appropriate shape’s area value by calling calcArea()method of the relevant class(e.g., Circle), and

–also has a method “compareArea()” to compare the areas of different shapes with wild-card parameter to determine which shape is largest or smallest in terms of area.

Formula to calculate area:Circle:

$A=\pi*r2$Where r is radius

Square: $A=Length*Width$, Where Length and Width are same

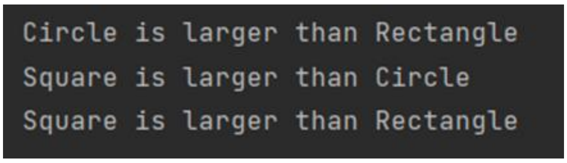
Rectangle: $A=Length*Width$

Code:

```
import java.lang.Math;
public class Node<T extends Shape>
{
    Shape s=new Shape();;
    double area;
    static void compareArea(Node<?> obj1,Node<?> obj2)
    {
        if(obj1.area>obj2.area)
        {
            System.out.println(obj1.s.getClass().getName()+" is 
larger than "+obj2.s.getClass().getName());
        }
        else if(obj1.area<obj2.area)
        {
            System.out.println(obj2.s.getClass().getName()+" is 
larger than "+obj1.s.getClass().getName());
        }
        else
        {
            System.out.println(obj2.s.getClass().getName()+" and 
"+obj1.s.getClass().getName()+" have same area.");
        }
    }
    public static void main(String args[])
    {
        Node<Circle> cir=new Node<Circle>();
        Circle c=new Circle();
        cir.s=c;
        cir.area=c.calcArea();
        Node<Rectangle> rec=new Node<Rectangle>();
        Rectangle r=new Rectangle();
        rec.s=r;
        rec.area=r.calcArea();

        Node<Square> sqr=new Node<Square>();
        Square sq=new Square();
        sqr.s=sq;
        sqr.area=sq.calcArea();

        compareArea(cir,rec);
        compareArea(cir,sqr);
        compareArea(rec,sqr);
    }
}
class Shape
{
    public double calcArea()
    {
        return 0.0;
    }
}
class Circle extends Shape
{
    public double calcArea()
    {
        double radius=2.0;
        return (Math.PI)*radius*radius;
    }
```

```
}
class Rectangle extends Shape
{
    public double calcArea()
    {
        double length=5.0;
        double width=2.0;
        return length*width;
    }
}
class Square extends Shape
{
    public double calcArea()
    {
        double length=5.0;
        double width=length;
        return length*width;
    }
}
```

Output:



```
Circle is larger than Rectangle
Square is larger than Circle
Square is larger than Rectangle
```