

WEEK4 LAB4

1. Define a class Maximum with the following overloaded methods

- a)max (which finds maximum among three integers and returns the maximum integer)
- b)max (which finds maximum among three floating point numbers and returns the maximum among them)
- c)max (which finds the maximum in an array and returns it)
- d)max (which finds the maximum in a matrix and returns the result)

Place this in a package called p1. Let this package be present in a folder called "myPackages", which is a folder in your present working directory (eg: c:\student\3rdsem \ mypackages\p1). Write a main method to use the methods of Max class in a package p1.

Under Folder myPackages-> p1 ->Max.java

```
package p1;

public class Max
{
    public int max (int a, int b, int c)
    {
        return (a>b)?((a>c)?a:c):((b>c)?b:c);
    }

    public float max (float a, float b, float c)
    {
        return (a>b)?((a>c)?a:c):((b>c)?b:c);
    }

    public int max (int a[])
    {
        int max = 0;

        for (int i : a)

            if (i > max)

                max = i;
```

```
return max;
}
```

```
public int max(int a[])
{
    int max = 0;
    for (int b[] : a)
        for (int i : b)
            if (i > max)
                max = i;
    return max;
}
```

under folder myPackages→ pgm1.java

```
import p1.Max;

import java.util.Scanner;

public class pgm1

{

    public static void main(String[] args)

    {

        Scanner sc = new Scanner(System.in);

        Max max = new Max();

        int choice;

        System.out.println("\n\t\t-----\n");

        System.out.println("\n\t\t\t\t\tMAXIMUM , INTEGER NUMBER , FLOAT NUMBER , MAX IN ARRAY , MAX  
IN MATRIX \n ");

        System.out.println("\n\t\t\t\t\t-----\n\n");

        System.out.println("\n\t\t\t\t\t1-MAXIMUM INTEGER NUMBER");

        System.out.println("\n\t\t\t\t\t2-MAXIMUM FLOATING NUMBER");
```

```

System.out.println("\n\t\t\t\t3-MAXIMUM NUMBER IN ARRAY");

System.out.println("\n\t\t\t\t4-MAXIMUM MATRIX ");

System.out.println("\n\t\t\t-----\n\n");

System.out.println("\n\t\t\t\tEnter your choice = ");

choice = sc.next().charAt(0);

switch(choice)

{

case '1':

{

int a, b, c;

System.out.println("\n\t\t\t\tEnter the three integers = ");

a = sc.nextInt();

b = sc.nextInt();

c = sc.nextInt();

System.out.println("\n\t\t\t\tThe largest integer is = " + max.max(a, b, c));

}

break;

case '2':

{

float a, b, c;

System.out.println("\n\t\t\t\tEnter the three integers = ");

a = sc.nextFloat();

b = sc.nextFloat();

c = sc.nextFloat();

System.out.println("\n\t\t\t\tThe largest floating point number is = " + max.max(a, b, c));

}

break;

case '3' :

{

```

```

System.out.print("\n\t\t\t\tEnter number of integers in array = ");

int n = sc.nextInt();

int arr[] = new int[n];

System.out.print("\n\t\t\t\tEnter " + n + " elements is = ");

for(int i = 0; i < n; ++i)

    arr[i] = sc.nextInt();

System.out.println("\n\t\t\t\tLargest is = " + max.max(arr));

}

break;

case '4' :

{

System.out.print("\n\t\t\t\tEnter size of the matrix = ");

int r = sc.nextInt();

int c = sc.nextInt();

int mat[][] = new int[r][c];

System.out.print("\n\t\t\t\tEnter the matrix = ");

for(int i = 0; i < r; ++i)

for (int j = 0; j < c; ++j)

    mat[i][j] = sc.nextInt();

System.out.println("\n\t\t\t\tLargest is = " + max.max(mat));

}

break;

default :System.out.println("\n\t\t\t\tINVALID CHOICE ! \n\n");

break;

}

}

}

```

OUTPUT :

```
student@lplab-Lenovo-Product: ~/190905514/myPackages
student@lplab-Lenovo-Product:~/190905514/myPackages$ clear
student@lplab-Lenovo-Product:~/190905514/myPackages$ ls
p1  pgm1.class  pgm1.java
student@lplab-Lenovo-Product:~/190905514/myPackages$ javac pgm1.java
student@lplab-Lenovo-Product:~/190905514/myPackages$ java pgm1

-----

MAXIMUM , INTEGER NUMBER , FLOAT NUMBER ,MAX IN ARRAY , MAX IN MATRIX

-----

1-MAXIMUM INTEGER NUMBER
2-MAXIMUM FLOATING NUMBER
3-MAXIMUM NUMBER IN ARRAY
4-MAXIMUM MATRIX

-----

Enter your choice =
1
Enter the three integers =
23 14 2

The largest integer is = 23
student@lplab-Lenovo-Product:~/190905514/myPackages$
```

```

student@lplab-Lenovo-Product: ~/190905514/myPackages
student@lplab-Lenovo-Product:~/190905514/myPackages$ java pgm1

-----

MAXIMUM , INTEGER NUMBER , FLOAT NUMBER ,MAX IN ARRAY , MAX IN MATRIX

-----

1-MAXIMUM INTEGER NUMBER
2-MAXIMUM FLOATING NUMBER
3-MAXIMUM NUMBER IN ARRAY
4-MAXIMUM MATRIX

-----

Enter your choice =
2

Enter the three integers =
34.0 3.0 56.0

The largest floating point number is = 56.0
student@lplab-Lenovo-Product:~/190905514/myPackages$

```

```

student@lplab-Lenovo-Product: ~/190905514/myPackages
student@lplab-Lenovo-Product:~/190905514/myPackages$ java pgm1

-----

MAXIMUM , INTEGER NUMBER , FLOAT NUMBER ,MAX IN ARRAY , MAX IN MATRIX

-----

1-MAXIMUM INTEGER NUMBER
2-MAXIMUM FLOATING NUMBER
3-MAXIMUM NUMBER IN ARRAY
4-MAXIMUM MATRIX

-----

Enter your choice =
3

Enter number of integers in array = 5
Enter 5 elements is = 34 23 3 4 -90

Largest is = 34
student@lplab-Lenovo-Product:~/190905514/myPackages$

```

```

student@lplab-Lenovo-Product: ~/190905514/myPackages
student@lplab-Lenovo-Product:~/190905514/myPackages$ java pgm1

-----

MAXIMUM , INTEGER NUMBER , FLOAT NUMBER ,MAX IN ARRAY , MAX IN MATRIX

-----

1-MAXIMUM INTEGER NUMBER
2-MAXIMUM FLOATING NUMBER
3-MAXIMUM NUMBER IN ARRAY
4-MAXIMUM MATRIX

-----

Enter your choice =
4

Enter size of the matrix = 3 3
Enter the matrix = 1 2 3 4 5 6 7 8 9

Largest is = 9
student@lplab-Lenovo-Product:~/190905514/myPackages$

```

2. Create an abstract class Figure with abstract method area and two integer dimensions. Create three more classes Rectangle, Triangle and Square which extend Figure and implement the area method. Show how the area can be computed dynamically during run time for Rectangle, Square and Triangle to achieve dynamic polymorphism. (Use the reference of Figure class to call the three different area methods),

pgm3.java

```

abstract class figure {

    double dimension1;

    double dimension2;

    abstract double area();

}

```

```
class Rectangle extends figure {  
  
    Rectangle(double c, double d) {  
  
        dimension1 = c;  
  
        dimension2 = d;  
  
    }  
  
    @Override  
  
    double area() {  
  
        return dimension2 * dimension1;  
  
    }  
  
}
```

```
class Triangle extends figure {  
  
    Triangle(double a, double b) {  
  
        dimension1 = a;  
  
        dimension2 = b;  
  
    }  
  
    @Override  
  
    double area() {  
  
        return dimension2 * dimension1;  
  
    }  
  
}
```



```
class Square extends figure {

    @Override

    double area() {

        return dimension1 * dimension2;

    }

}

public class pgm3 {

    public static void main(String[] args) {

        Rectangle rob = new Rectangle(4, 5);

        Triangle tob = new Triangle(4, 5);

        Square sob = new Square();

        figure f;

        f=rob;

        System.out.println("Area of Rectangle is = " + rob.area());

        System.out.println("Area of Triangle is = " + tob.area());

        System.out.println("Area of Square is = " + f.area());

    }

}
```

OUTPUT :

```
student@lplab-Lenovo-Product: ~/190905514/week4
student@lplab-Lenovo-Product:~/190905514/myPackages$ ls
p1  pgm1.class  pgm1.java
student@lplab-Lenovo-Product:~/190905514/myPackages$ cd ..
student@lplab-Lenovo-Product:~/190905514$ ls
myPackages  week3  week4
student@lplab-Lenovo-Product:~/190905514$ cd week4]
bash: cd: week4]: No such file or directory
student@lplab-Lenovo-Product:~/190905514$ cd week4
student@lplab-Lenovo-Product:~/190905514/week4$ ls
lab6  pgm1.java  pgm2.java  pgm3.java
student@lplab-Lenovo-Product:~/190905514/week4$ javac pgm3.java
student@lplab-Lenovo-Product:~/190905514/week4$ java pgm3
Area of Rectangle is = 20.0
Area of Triangle is = 20.0
Area of Square is = 20.0
student@lplab-Lenovo-Product:~/190905514/week4$
```

3. Design an interface called Series with the following methods
a. Get Next (returns the next number in series)
b. reset (to restart the series)
c. set Start (to set the value from which the series should start)
Design a class named By Twos that will implement the methods of the interface Series such that it generates a series of numbers, each two greater than the previous one. Also design a class which will include the main method for referencing the interface

pgm2.java

```
import java.util.Scanner;

import static java.lang.System.exit;

interface series {
    void getNext();

    void reset();

    void setStart();
}

class byTwo implements series {
    int num1;
    int num2;

    byTwo(int a, int b) {
        num1 = a;
        num2 = b;
    }

    public void reset() {
        num1 = 0;
        num2 = 0;
    }
}
```

```

        System.out.println("Starting point reset to Zero");
    }

    public void getNext() {
        for (int i = 0; i < num2; i++) {
            num2 = num2 - 2;
            System.out.println("Number one is = " + num2);
        }
    }

    public void setStart() {
        System.out.println("Enter starting point : ");
        Scanner scanner = new Scanner(System.in);
        int num1 = scanner.nextInt();
        System.out.println("Enter the number of terms : ");
        int num2 = scanner.nextInt();
        System.out.println("Strating point is = "+num1);
        System.out.println("Number of term is = "+num2);

    }

}

public class pgm2 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int num1 = 0;
        int num2 = 0;
        int n = 1;
        byTwo bobj = new byTwo(num1, num2);
        while (n != 0) {
            System.out.println("-----");
            System.out.println("1-Set Start");
            System.out.println("2-getNext");
            System.out.println("3-Reset");
            System.out.println("4-Exit");
            System.out.println("-----");
            System.out.println("Enter the choice : ");
            int choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    bobj.setStart();
                    break;
                case 2:
                    bobj.getNext();
                    break;
                case 3:
                    bobj.reset();
                    break;
                case 4:
                    exit(0);
                    break;
                default:
            }
        }
    }
}

```

```
System.out.println("Invalid choice !");
```

```
}  
}  
}  
}
```

OUTPUT :

```
student@lplab-Lenovo-Product: ~/tofik/week3  
student@lplab-Lenovo-Product:~/tofik/week3$ javac pgm2.java  
student@lplab-Lenovo-Product:~/tofik/week3$ java pgm2  
-----  
1-Set Start  
2-getNext  
3-Reset  
4-Exit  
-----  
Enter the choice :  
1  
Enter starting point :  
2  
Enter the number of terms :  
3  
Strating point is = 2  
Number of term is = 3  
-----  
1-Set Start  
2-getNext  
3-Reset  
4-Exit  
-----  
Enter the choice :  
2  
-----  
1-Set Start  
2-getNext  
3-Reset  
4-Exit  
-----  
Enter the choice :  
3
```

```
student@lplab-Lenovo-Product: ~/tofik/week3  
-----  
1-Set Start  
2-getNext  
3-Reset  
4-Exit  
-----  
Enter the choice :  
3  
Starting point reset to Zero  
-----  
1-Set Start  
2-getNext  
3-Reset  
4-Exit  
-----  
Enter the choice :  
1  
Enter starting point :  
5  
Enter the number of terms :  
6  
Strating point is = 5  
Number of term is = 6  
-----  
1-Set Start  
2-getNext  
3-Reset  
4-Exit  
-----  
Enter the choice :  
4  
student@lplab-Lenovo-Product:~/tofik/week3$
```

