Week 4

1)
Maximum.java

// max (which finds maximum among three integers and returns the maximum integer)

// max (which finds maximum among three floating point numbers and returns the maximum among them)

// max (which finds the maximum in an array and returns it)

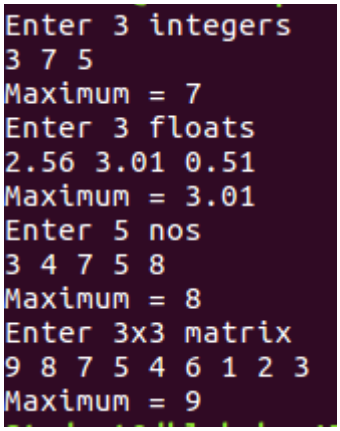
// max (which finds the maximum in a matrix and returns the result)

// 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.

```
package myPackages.p1;

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

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

	public static int max(int[] arr){
		int max = arr[0];
		for(int i = 0; i < arr.length; i++){
			if(arr[i] > max){
				max = arr[i];
			}
		}
		return max;
	}

	public static int max(int[][] arr, int r, int c){
		int max = arr[0][0];
		for(int i = 0; i < r; i++){
			for(int j = 0; j < c; j++){
				if(arr[i][j] > max){
					max = arr[i][j];
				}
			}
		}
		return max;
	}
}
```

maxtest.java

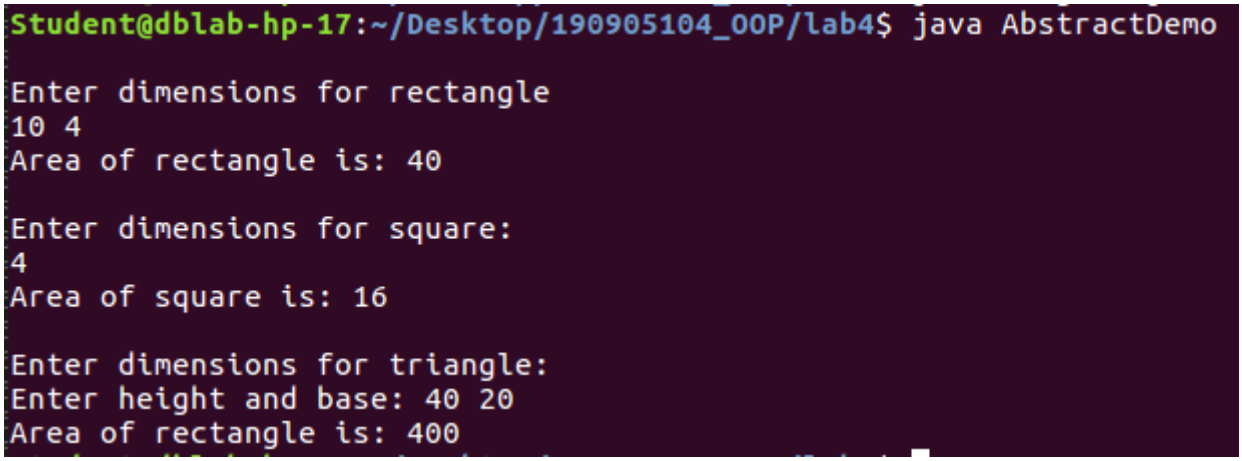
```
// To test the package

import java.util.Scanner;
import myPackages.p1.Maximum;

class MaxTest{
	public static void main(String []args){
		Scanner sc = new Scanner(System.in);
		int a, b, c;
		System.out.println("Enter 3 integers");
		a = sc.nextInt();
		b = sc.nextInt();
		c = sc.nextInt();
		System.out.println("Maximum = " + Maximum.max(a, b, c));

		float d, e, f;
		System.out.println("Enter 3 floats");
		d = sc.nextFloat();
		e = sc.nextFloat();
		f = sc.nextFloat();
		System.out.println("Maximum = " + Maximum.max(d, e, f));
		System.out.println("Enter 5 nos");
		int arr[] = new int[5];
		for(int i = 0; i < 5; i++)
			arr[i] = sc.nextInt();
		System.out.println("Maximum = " + Maximum.max(arr));
		System.out.println("Enter 3x3 matrix");
		int mat[][] = new int[3][3];
		for(int i = 0; i < 3; i++)
			for(int j = 0; j < 3; j++)
				mat[i][j] = sc.nextInt();
		System.out.println("Maximum = " + Maximum.max(mat, 3, 3));
	}
}
```



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)

```
import java.util.Scanner;

abstract class Figure{
	int dim1, dim2;
	Figure(int d1, int d2){
		dim1 = d1;
		dim2 = d2;
	}
	abstract void area();
}

class Rectangle extends Figure{
	Rectangle(int l, int b){
		super(l, b);
	}
	void area() {
		System.out.println("Area of rectangle is: " + super.dim1 * super.dim2);
	}
}

class Square extends Figure{
	Square(int s1){
		super(s1, s1);
	}
	void area() {
		System.out.println("Area of square is: " + super.dim1 * super.dim2);
	}
}

class Triangle extends Figure{
	Triangle(int b, int h){
		super(b, h);
	}
	void area() {
		System.out.println("Area of rectangle is: " + (super.dim1 * super.dim2) / 2);
	}
}

class AbstractDemo{
	public static void main(String []args) {
		Scanner sc = new Scanner(System.in);
		int a, b;

		Figure fig;
```

```
                System.out.println("\nEnter dimensions for rectangle");
                a = sc.nextInt();
                b = sc.nextInt();
                fig = new Rectangle(a, b);
                fig.area();

                System.out.println("\nEnter dimensions for square: ");
                a = sc.nextInt();
                fig = new Square(a);
                fig.area();

                System.out.println("\nEnter dimensions for triangle: ");
                System.out.print("Enter height and base: ");
                a = sc.nextInt();
                b = sc.nextInt();
                fig = new Triangle(a, b);
                fig.area();
        }
}
```



3)
//Design an interface called Series with the following methods

//Get Next (returns the next number in series)

//reset(to restart the series)

//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.

```
import java.util.Scanner;

interface Series{
        int get_next();
```

```
	void reset();
	void set_start(int start);
}

class ByTwos implements Series{
	int current;

	public int get_next(){
		current += 2;
		return current;
	}

	public void reset(){
		current = 0;
	}

	public void set_start(int n){
		current = n;
	}
}

public class SeriesDemo{
	public static void main(String []args){
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter start");
		int start = sc.nextInt();
		ByTwos series = new ByTwos();
		series.set_start(start);
		System.out.println("How many values do you want?");
		int n = sc.nextInt();
		for(int i = 0; i < n; i++){
			System.out.print(series.get_next() + " ");
		}
		System.out.println("\n Resetting series");
		series.reset();
		System.out.println("How many values do you want?");
		n = sc.nextInt();
		for(int i = 0; i < n; i++){
			System.out.print(series.get_next() + " ");
		}
		System.out.println("\n");
	}
}
```

```
Student@dblab-hp-17:~/Desktop/190905104_OOP/lab4$ java SeriesDemo
Enter start
4
How many values do you want?
5
6 8 10 12 14
 Resetting series
How many values do you want?
5
2 4 6 8 10
```