

## Assignment No. 02 (Unit 2)

PAGE NO. 14  
DATE: / /

Q1>

Create a class named Matrix which will have each of the following :

- 1 - get the number of rows
- 2 - get the number of columns
- 3 - Set the elements of the matrix at given position (i,j)
- 4 - Adding two matrices. If the matrices are not addable, "Matrices cannot be added", will be displayed.
- 5 - Multiplying the two matrices.

→

```
import java.util.Scanner;
```

```
class Matrix {  
    int rows;  
    int columns;  
    int [][]matrix;  
    int [][] product;  
    int [][] sum;
```

```
Final Scanner sc = new Scanner(System.in);
```

```
void getRowsC() {  
    rows = sc.nextInt();  
}
```

```
void Matrix() {  
    getRowsC();  
    getColumnsC();
```

```

matrix = new int [rows] [columns];
sum = new int [rows] [columns];
product = new int [rows] [columns];
    
```

```

Scanner sc = new Scanner (System.in);
    
```

```

for (int i=0; i<rows; i++) {
    
```

```

        for (int j=0; j<columns; j++) {
            
```

```

            System.out.println("Enter element at index [" +
                + i + "][" + j + "] : ");
        
```

```

        matrix [i] [j] = sc.nextInt();
    
```

```

}
    
```

```

}
    
```

```

}
    
```

```

}
    
```

```

void getColumns() {
    
```

```

    System.out.print("Enter number of columns : ");
    
```

```

    columns = sc.nextInt();
    
```

```

}
    
```

```

* Matrix <int> = {
    
```

```

    void setAt (int i, int j, int value) {
        
```

```

            matrix [i] [j] = value;
        
```

```

    }
    
```

```

void add (Matrix a) {
    
```

```

if (this.rows == a.rows && this.columns == a.columns) {
    
```

```

        for (int i=0; i<this.rows; i++) {
    
```

```
for(int j=0; j<columns; j++) {  
    sum[i][j] = this.matrix[i][j] + a.matrix[i][j];  
}  
else if (rows < a.rows || columns < a.columns) {  
    System.out.println("Matrix cannot be added");  
}  
else if (rows > a.rows || columns > a.columns) {  
    System.out.println("Multiplication not possible");  
}  
else {  
    void multiply(Matrix a) {  
  
        for(int i=0; i<rows; i++) {  
            for(int j=0; j<columns; j++) {  
                try {  
                    product[i][j] = matrix[i][j] *  
                        a.matrix[i][j];  
                } catch (Exception e) {}  
            } finally {  
                continue;  
            }  
        }  
    }  
}
```

3

display (product); (return sm) unqsm

3

```
void display (int [][] a) {
    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix[i].length; j++) {
            System.out.print(a[i][j] + " ");
        }
        System.out.println();
    }
}
```

System.out.println();

3

3 : s : variable for entering value

3 s : [0][0] value to elements value

3 : [0][1] value to elements value

class Q1 { to enter the elements value

..... : to print the value

public static void main (String [] args) {

..... System.out.println ("Enter Matrix 1 : ");

2 : Matrix m1 = new Matrix ();

2 : System.out.println ("Enter Matrix 2 : ");

2 : Matrix m2 = new Matrix ();

3 : [0][0] value to elements value

m1.add (m2); add method

m1.multiply (m2); 8.3

System.out.println ("Matrix after setting element at given position!");

```
m1.setAt(0,0,11);  
m1.display(m1.matrix);  
3
```

Output:

CMD tom ~ i0 = 1.000

javac P1.java

java P1

Enter Matrix 1:

Enter number of rows : 2

Enter number of columns : 2

Enter element at index [0][0] : 1

Enter element at index [0][1] : 2

Enter element at index [1][0] : 3

Enter element at index [1][1] : 4

Enter Matrix 2:

Enter number of rows : 2

Enter number of columns : 2

Enter element at index [0][0] : 5

Enter element at index [0][1] : 6

Enter element at index [1][0] : 7

Enter element at index [1][1] : 8

Addition Result:

6 8

10 12

Multiplication Result:

14 16

28 32

Matrix after Setting element at given position:

11 2

3 4

- Q2) Write a program to print the area and perimeter of a triangle having sides of 3, 4, and 5 units by creating a class named Triangle without any parameters in its constructor.

Code:

```
class Triangle {
```

```
    Triangle () {
```

```
        area (3, 4, 5);
```

```
        peri (3, 4, 5);
```

```
    }
```

```
    void area (int a, int b, int c) {
```

```
        float s = (a+b+c)/2.0;
```

```
        float A = (float) Math.sqrt(s*(s-a)*(s-b)*(s-c));
```

```
        System.out.println ("Area of triangle :: " + A +
```

```
            " Units");
```

```
    }
```

```
void peri(int a, int b, int c) {
```

```
    System.out.println("Perimeter of triangle" + (a+b+c)
        + " Units");
```

Writing code to 3 reme print the result

3

5 11

4 8

```
class P2 {
```

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

```
    Triangle t = new Triangle();
```

3

Output:

[END]

javac P2.java

java P2

Area of triangle : 6.0 units

Area of

Perimeter of triangle : 12 units.

- Q.3) Create a class 'Complex' with separate methods for each operation whose real and imaginary parts are entered by user. WAP which prints the sum, difference and product of two complex numbers.

→

Code:

```

import java.util.Scanner;
class Complex {
    int a, b;
}

```

Complex() {

```
Scanner sc = new Scanner(System.in);
```

System.out.println("Enter the real part of equation:");

```
a = sc.nextInt();
```

```
b = sc.nextInt();
```

System.out.println("Enter the imaginary part of");

equation : ");

```
b = sc.nextInt();
```

}

void add(Complex c1) {

System.out.println("\nAddition: \n");

```
a = a + c1.a; b = b + c1.b;
```

```
System.out.println(" + " + a + "i" + " + " + b + "j");
```

```
System.out.println("-----");
```

```
System.out.println(" " + (a + c1.a) + " + " + (b + c1.b) + "j");
```

```
System.out.println("-----");
```

3 ((5))

((5))

void subtract(Complex c1) {

System.out.println("\nA - Subtraction : \n");

```
a = a - c1.a; b = b - c1.b;
```

```

System.out.println("- " + "c1.a" + " + " + c1.b + "j");
System.out.println("-----");
System.out.println(" " + "(a - c2.a) + " + "( " + "c2.b" + " ) + " + "j");
System.out.println("-----");
3

```

void Multiply (Complex c1) {

```
System.out.println("In Multiplication : \n");
```

```
System.out.println(" " + a + " + " + b + "j ");
```

```
System.out.println("x " + "c1.a" + " + " + "c1.b" + "j ");
```

```
System.out.println("-----");
```

```
System.out.println(" : " + "(a x " + "c1.a" + " ) + " + "(b x c1.b) + "j ");
```

```
System.out.println("-----");
```

3

class Q3 {

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

```
Complex c1 = new Complex();
```

```
Complex c2 = new Complex();
```

```
c1.add(c2);
```

```
c1.subtract(c2);
```

```
c2.multiply(c2);
```

3

Output:

javac Q3.java

java Q3

Enter the real part of equation:

1

Enter the imaginary part of equation:

2

Enter the real part of equation:

2

Enter the imaginary part of equation.

Addition:

$$1 + 2j$$

$$+ 2 + 2j$$

$$\underline{3 + 4j}$$

Subtraction:

$$1 + 2j$$

$$- 2 + 2j$$

$$\underline{-1 + 0j}$$

Multiplication :

$$1 + 2j$$

$$\times 2 + 2j$$

$$\underline{1 + 2j}$$

$$+ 2 + 4j$$

$$-----$$

$$2 + 4j$$

- Q4) Write a program to create a room class and display data, the attributes of this class is roomno, roomtype, roomarea, and ACMachine. In this class the methods are setdata and displaydata.

→ Code:

Class Room {

    int roomNo;

    String roomType;

    float roomArea;

    boolean AC machine;

    void setdata( int roomNo, String roomType, float roomArea,

, boolean ACMachine) {

        this.roomNo = roomNo;

        this.roomType = roomType;

        this.roomArea = roomArea;

        this.ACmachine = ACMachine;

}

```

    void displayData() {
        System.out.println("room no : " + roomNo);
        System.out.println("Room type : " + roomType);
        System.out.println("Room area : " + roomArea);
        System.out.println("AC facility : " + acMachine ?
            "Available" : "Unavailable"));
    }
}

```

3 (Output for screen print) ~~Output~~

```
class Q4 { Room = smarit
```

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

```
Room r1 = new Room();
```

```
r1.setData(101, "Deluxe Suite", 250.56F, true);
```

```
r1.displayData();
```

3

~~Output~~

3 (Output) ~~Output~~ bin code ~~Output~~

(smarit + " : smart") ~~Output~~.~~Output~~

Output : (101) ~~Output~~.~~Output~~

Output : (101) ~~Output~~.~~Output~~

javac Q4.java

java Q4

Roomno : 101 bin code ~~Output~~

Room type : Deluxe Suite ~~Output~~

Room Area : 250.56 ~~Output~~.~~Output~~

AC facility : Available ~~Output~~

Q5) Write a program to demonstrate the use of static, this and final keyword.

```
→ (Author + " : " + name) printing due module
code: → (Author + " : " + name) printing due module
final class Student {
    final String mathsTeacher = "Laxmi ma'am";
    static String name;
    static int rollNo;
```

```
void setData(String name, int rollNo) {
    this.name = name;
```

~~this.rollNo = rollNo;~~

~~3.0000 - 2 = 2 mod~~

class Q5 {

```
public static void display(Student Q) {
    System.out.println("Name : " + Q.name);
```

```
    System.out.println("Roll No : " + Q.rollNo);
```

```
    System.out.println("Maths : " + Q.mathsTeacher);
```

}

```
public static void main(String[] args) {
    Student student = new Student();
```

```
    student.setData("Pratyay Dhand", 1907011);
```

```
    display(student);
```

}

3

(C:\In0) &gt;0 .shl fromapieea

Output: with statements od mapping most to file (EP  
leads updated to CMD) all given algorithm has shown

javac Q5.java	←
java Q5	: sh2)

Execution mode - spooling

Name : Pratyay Dhond

Roll No : 19070111 from 2do

: Maths : I Laxmi Ma'am?

: 32.0 = Hard tool9 hard fast tool9

(d tool9 c tool9) new tool9 hard

Execution mode - spooling

3. sortA struct element tools

3 (hard tool9 align tool9) new tool9

: aligned \* align number

{}

{}

3 sortA struct element tools

3 (hard tool9 align tool9) new tool9

: align \* align \* 19 native

{}

{}

3 sortA struct element tools

3 hard tool9 align tool9 new tool9

Tool number