

## CYCLE 1

### PROGRAM 1

Define a class 'product' with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.

CODE :

```
class product{
    String pcode;
    String pname;
    int price;
public product(String pcode_get,String pname_get, int price_get){
    pcode = pcode_get;
    pname = pname_get;
    price = price_get;
}
public int get_price(){
    return price;
}
public static void main(String[] args)
{
    product p_1 = new product("a123","tv",2);
    product p_2 = new product("b123", "radio",3);
    product p_3 = new product("c123","dvd",5);
```

```
System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n DATE : 24-3-
2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT ORIENTED
PROGRAMMING LAB");
System.out.println("-----OUTPUT-----");
if(p_1.price <= p_3.price && p_1.price <= p_2.price)
    System.out.println("lowest product id is : " +p_1.pcode);
if(p_3.price <= p_1.price && p_3.price <= p_2.price)
    System.out.println("lowest product id i : " + p_3.pcode);
if (p_2.price <= p_3.price && p_2.price <= p_1.price)
    System.out.println("Lowest product ID is : " + p_2.pcode);
}
}
```

OUTPUT :

```
sjcet@Z238-UL:~/jom/java/cycle1$ javac product.java
sjcet@Z238-UL:~/jom/java/cycle1$ java product

NAME : JOM BINOY
REG NO :SJC22MCA-2033
DATE : 24-3-2023
COURSE CODE : 20MCA132
COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----OUTPUT-----
lowest product id is : a123
```

## PROGRAM 2 :

Read 2 matrices from the console and perform matrix addition.

### CODE :

```
import java.util.Scanner;

public class MatrixAddition {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n DATE : 24-3-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB");
        System.out.println("-----OUTPUT-----");
        System.out.print("Enter the number of rows: ");
        int rows = input.nextInt();
        System.out.print("Enter the number of columns: ");
        int cols = input.nextInt();
        int[][] matrix1 = new int[rows][cols];
        int[][] matrix2 = new int[rows][cols];
        System.out.println("Enter the elements of the first matrix:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                matrix1[i][j] = input.nextInt();
            }
        }
        System.out.println("Enter the elements of the second matrix:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                matrix2[i][j] = input.nextInt();
            }
        }
        int[][] sum = new int[rows][cols];
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                sum[i][j] = matrix1[i][j] + matrix2[i][j];
            }
        }
        System.out.println("The sum of the matrices is:");
        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                System.out.print(sum[i][j] + " ");
            }
            System.out.println();
        }

        input.close();
    }
}
```

```

sjcet@Z238-UL:~/jom/java/cycle1$ javac MatrixAddition.java
sjcet@Z238-UL:~/jom/java/cycle1$ java MatrixAddition

NAME : JOM BINOY
REG NO : SJC22MCA-2033
DATE : 24-3-2023
COURSE CODE : 20MCA132
COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----OUTPUT-----
Enter the number of rows: 2
Enter the number of columns: 2
Enter the elements of the first matrix:
3
4
5
6
Enter the elements of the second matrix:
1
2
3
4
The sum of the matrices is:
4 6
8 10
sjcet@Z238-UL:~/jom/java/cycle1$

```

### PROGRAM 3 :

Add complex numbers

CODE :

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

```

public class ComplexAddition {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n DATE : 24-3-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB");
        System.out.println("-----OUTPUT-----");
        System.out.print("Enter the real part of the first complex number: ");
        double real1 = input.nextDouble();
        System.out.print("Enter the imaginary part of the first complex number: ");
        double imaginary1 = input.nextDouble();
        System.out.print("Enter the real part of the second complex number: ");
        double real2 = input.nextDouble();
        System.out.print("Enter the imaginary part of the second complex number: ");
        double imaginary2 = input.nextDouble();
        double realSum = real1 + real2;
        double imaginarySum = imaginary1 + imaginary2;
        System.out.println("The sum of the complex numbers is: " + realSum + " + " + imaginarySum + "i");

        input.close();
    }
}

```

```
}
```

OUTPUT :

```
sjcet@Z238-UL:~/jom/java/cycle1$ javac ComplexAddition.java
sjcet@Z238-UL:~/jom/java/cycle1$ java ComplexAddition

NAME : JOM BINOY
REG NO :SJC22MCA-2033
DATE : 24-3-2023
COURSE CODE : 20MCA132
COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----OUTPUT-----
Enter the real part of the first complex number: 25
Enter the imaginary part of the first complex number: 45
Enter the real part of the second complex number: 47
Enter the imaginary part of the second complex number: 65
The sum of the complex numbers is: 72.0 + 110.0i
```

PROGRAM 4 :

Read a matrix from the console and check whether it is symmetric or not.

CODE :

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

```
public class SymmetricMatrix {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n DATE : 24-3-2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB");
        System.out.println("-----OUTPUT-----");
        System.out.print("Enter the number of rows in the matrix: ");
        int numRows = input.nextInt();
        System.out.print("Enter the number of columns in the matrix: ");
        int numCols = input.nextInt();
        int[][] matrix = new int[numRows][numCols];
        System.out.println("Enter the matrix elements:");
        for (int i = 0; i < numRows; i++) {
            for (int j = 0; j < numCols; j++) {
                matrix[i][j] = input.nextInt();
            }
        }
        boolean isSymmetric = true;
        for (int i = 0; i < numRows; i++) {
            for (int j = 0; j < numCols; j++) {
                if (matrix[i][j] != matrix[j][i]) {
                    isSymmetric = false;
                    break;
                }
            }
        }
    }
}
```

```

        if (!isSymmetric) {
            break;
        }
    }
    if (isSymmetric) {
        System.out.println("The matrix is symmetric.");
    } else {
        System.out.println("The matrix is not symmetric.");
    }

    input.close();
}
}

```

OUTPUT :

```

sjcet@Z238-UL:~/jom/java/cycle1$ javac SymmetricMatrix.java
sjcet@Z238-UL:~/jom/java/cycle1$ java SymmetricMatrix

NAME : JOM BINOY
REG NO :SJC22MCA-2033
DATE : 24-3-2023
COURSE CODE : 20MCA132
COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----OUTPUT-----
Enter the number of rows in the matrix: 2
Enter the number of columns in the matrix: 2
Enter the matrix elements:
6
7
76
6
The matrix is not symmetric.

```

PROGRAM 5 :

CODE :

```

class Cpu {
    int price;

    Cpu(int p) {
        this.price = p;
    }

    class Processor {
        int cores;
        String manufacture;

        Processor(int n, String m) {
            this.cores = n;
            this.manufacture = m;
        }
    }
}

```

```

    void display() {
        System.out.println("No of Cores : " + this.cores);
        System.out.println("Processor manufactures : " + this.manufacture);
    }
}

static class Ram {
    int memory;
    String manufacture;

    Ram(int n, String m) {
        this.memory = n;
        this.manufacture = m;
    }

    void display() {
        System.out.println("Memory Size : " + this.memory);
        System.out.println("Memory manufactures : " + this.manufacture);
    }
}

void display() {
System.out.println("\n NAME : JOM BINOY \n REG NO :SJC22MCA-2033 \n DATE : 24-3-
2023 \n COURSE CODE : 20MCA132\n COURSE NAME : OBJECT ORIENTED
PROGRAMMING LAB");
System.out.println("-----OUTPUT-----");
    System.out.println("Price of CPU : " + this.price);
}

public static void main(String[] args) {
    Cpu intel = new Cpu(23000);
    Cpu.Processor i_processor = intel.new Processor(4, "intel");
    Cpu.Ram i_ram = new Ram(1024, "Asus");
    intel.display();
    i_processor.display();
    i_ram.display();
}
}

```

OUTPUT :

```

sjcet@Z238-UL:~/jom/java/cycle1$ javac Cpu.java
sjcet@Z238-UL:~/jom/java/cycle1$ java Cpu

NAME : JOM BINOY
REG NO :SJC22MCA-2033
DATE : 24-3-2023
COURSE CODE : 20MCA132
COURSE NAME : OBJECT ORIENTED PROGRAMMING LAB
-----OUTPUT-----
Price of CPU : 23000
No of Cores : 4
Processor manufactures : intel
Memory Size : 1024
Memory manufactures : Asus

```