

Course Outcome 1 (CO1):

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 {
    int pcode;
    String pname;
    double price;
}

public class Main{
    public static void main(String[] args) {
        System.out.println("Name: VISWAJITH P S\nReg:
SJ22MCA_2057\nDate: 24/03/2023\nCourse Code:
20MCA132");
        System.out.println("-----");

        Product p1 = new Product();
        p1.pcode = 1;
        p1.pname = "Product 1";
        p1.price = 10.0;

        Product p2 = new Product();
        p2.pcode = 2;
        p2.pname = "Product 2";
        p2.price = 20.0;

        Product p3 = new Product();
        p3.pcode = 3;
        p3.pname = "Product 3";
        p3.price = 5.0;
```

```

        Product lowestPriceProduct = p1;

        Product[] products = {p1, p2, p3};
        for (Product product : products) {
            if (product.price < lowestPriceProduct.price) {
                lowestPriceProduct = product;
            }
        }

        System.out.println("Product with lowest price:");
        System.out.println("Product code: " +
lowestPriceProduct.pcode);
        System.out.println("Product name: " +
lowestPriceProduct.pname);
        System.out.println("Product price: " +
lowestPriceProduct.price);
    }
}

```

OUTPUT:

```

sjcet@HP-Z238:~/appu$ javac Main.java
sjcet@HP-Z238:~/appu$ java Main
Name: VISWAJITH P S
Reg: SJ22MCA-2057
Date: 24/03/2023
Course Code: 20MCA132
-----
Product with lowest price:
Product code: 3
Product name: Product 3
Product price: 5.0

```

2. Read 2 matrices from the console and perform matrix addition.

CODE:

```
import java.util.Scanner;

public class MatrixAdd
{
    public static void main(String[] args)
    {
        System.out.println("Name : VISWAJITH P S \nReg No :  
SJC22MCA-2057 \nDate : 24/03/2023 \nCourse code : 20MCA132");

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

        Scanner input = new Scanner(System.in);
        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 values of matrix 1:");

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

```
matrix1[i][j] = input.nextInt();
```

```
}
```

```
}
```

```
System.out.println("Enter the values of matrix 2:");
```

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

```
{
```

```
for (int j = 0; j < cols; j++)
```

```
{
```

```
matrix2[i][j] = input.nextInt();
```

```
}
```

```
}
```

```
int[][] result = new int[rows][cols];
```

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

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

```
result[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(result[i][j] + " ");

    }

    System.out.println();

}

}

}

```

OUTPUT:

```

sjcet@HP-Z238:~/appu$ javac MatrixAdd.java
sjcet@HP-Z238:~/appu$ java MatrixAdd
Name : VISWAJITH P S
Reg No : SJC22MCA-2057
Date : 24/03/2023
Course code : 20MCA132
-----
Enter the number of rows: 2
Enter the number of columns: 2
Enter the values of matrix 1:
1
2
3
4
Enter the values of matrix 2:
8
7
6
5
The sum of the matrices is:
9 9
9 9

```

3. Add complex numbers

CODE:

```
public class ComplexNumber
{
    private double real;
    private double imaginary;

    public ComplexNumber(double real, double imaginary)
    {
        this.real = real;
        this.imaginary = imaginary;
    }

    public ComplexNumber add(ComplexNumber other)
    {
        double newReal = this.real + other.real;
        double newImaginary = this.imaginary + other.imaginary;
        return new ComplexNumber(newReal, newImaginary);
    }

    public String toString()
    {
        if (imaginary >= 0)
        {
            return real + " + " + imaginary + "i";
        } else
        {
            return real + " - " + (-imaginary) + "i";
        }
    }

    public static void main(String[] args)
    {
        System.out.println("Name : VISWAJITH P S \nReg No : SJC22MCA-2057 \nDate : 24/03/2023 \nCourse code : 20MCA132");
        System.out.println("-----");
        ComplexNumber num1 = new ComplexNumber(3, 4);
        ComplexNumber num2 = new ComplexNumber(1, -2);
        ComplexNumber sum = num1.add(num2);
        System.out.println("Sum: " + sum);
    }
}
```

```
}  
}
```

OUTPUT:

```
sjcet@HP-Z238:~/appu$ javac ComplexNumber.java  
sjcet@HP-Z238:~/appu$ java ComplexNumber  
Name : Vishal P S  
Reg No : SJC22MCA-2056  
Date : 24/03/2023  
Course code : 20MCA132  
-----  
Sum: 4.0 + 2.0i
```

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

CODE:

```
import java.util.Scanner;
public class Symmetric{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Name : VISWAJITH P S\nReg.no : 
SJC22MCA-2056\nDate : 24/03/2032\nCourse code : 
20MCA132\n-----");
        System.out.print("Enter the number of rows/columns: ");
        int n = scanner.nextInt();
        int[][] matrix = new int[n][n];
        System.out.println("Enter the elements of the matrix:");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                matrix[i][j] = scanner.nextInt();
            }
        }
        boolean isSymmetric = true;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < i; j++) {
                if (matrix[i][j] != matrix[j][i]) {
                    isSymmetric = false;
                    break;
                }
            }
        }
        if (isSymmetric) {
            System.out.println("The matrix is symmetric.");
        } else {
            System.out.println("The matrix is not symmetric.");
        }
    }
}
```


OUTPUT:

```
sjcet@HP-Z238:~/appu$ javac Symmetric.java
sjcet@HP-Z238:~/appu$ java Symmetric
Name : VISWAJITH P S
Reg.no : SJC22MCA-2057
Date : 24/03/2032
Course code : 20MCA132
-----
Enter the number of rows/columns: 2
Enter the elements of the matrix:
1
2
3
4
The matrix is not symmetric.
```

5. Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

CODE:

```
public class cpu{
    int price;
    class processor{
        int cores;
        String producer;
        processor(int noC, String manu){
            cores=noC;
            producer=manu;
            System.out.println("Name: VISWAJITH P S\nReg:
SJ22MCA_2057\nDate: 24/03/2023\nCourse Code: 20MCA132");
            System.out.println("-----");
        }
        void display(){
            System.out.println("\nProcessor info");
            System.out.println("No. of Cores = "+cores);
            System.out.println("Manufacturer = "+producer+"\n");
        }
    }
    static class ram{
        int mem;
        String manuf;
        ram(int memory,String producer ){
            mem=memory;
            manuf=producer;
        }
        void display(){
            System.out.println("\nRAM info");
            System.out.println("Memory = "+mem+" GB");
            System.out.println("Manufacturer = "+manuf+"\n");
        }
    }
    public static void main(String[] args) {
        cpu ram obj1= new cpu.ram(8,"Intel");
        cpu obj2 = new cpu();
    }
}
```

```
        cpu.processor obj3 = obj2.new processor(8,"Samsung");
        obj1.display();
        obj3.display();
    }
}
```

OUTPUT:

```
4 ERROR
sjcet@HP-Z238:~/appu$ javac cpu.java
sjcet@HP-Z238:~/appu$ java cpu
Name: VISWAJITH P S
Reg: SJ22MCA_2057
Date: 24/03/2023
Course Code: 20MCA132
-----

RAM info
Memory = 8 GB
Manufacturer = Intel

Processor info
No. of Cores = 8
Manufacturer = Samsung
```

6. Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

CODE:

```
import java.util.Scanner;
public class symmetric {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the Number of rows of the
Matrix");
        int row = sc.nextInt();
        System.out.println("Enter the Number of Columns of the
Matrix");
        int col = sc.nextInt();
        int matrix[][] = new int[row][col];
        int i,j;
        boolean state=true;
        for(i=0;i<row;i++){
            for(j=0;j<col;j++){
                System.out.println("Enter the Element at
M("+i+", "+j+"));
                matrix[i][j] = sc.nextInt();
            }
        }
        for(i=0;i<row;i++){
            for(j=0;j<col;j++){
                if(matrix[i][j]!=matrix[j][i]){
                    state=false;
                    break;
                }
            }
        }
    }
}
```

```
if(state){  
    System.out.println("Matrix is Symmetric");  
}  
else{  
    System.out.println("Matrix is Antisymmetric");  
}  
}  
}
```