OBJECT ORIENTED PROGRAMMING LAB

SUBMITTED BY,

BLESSY P ROY

S2 MCA

ROLL NO :MCA214

LABCYCLE 1

PROGRAM 1

AIM

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

ALGORITHM

STEP 1: Start
STEP 2: Define a class name as a product with members pname, pcode and price.
STEP 3: Define objects to Class and add 3 products and values to each data using the object.
STEP 4: Check whether the product has the lowest price using if-else statement.
STEP 5: Print the details of the product.
STEP 6: Stop

PROGRAM CODE

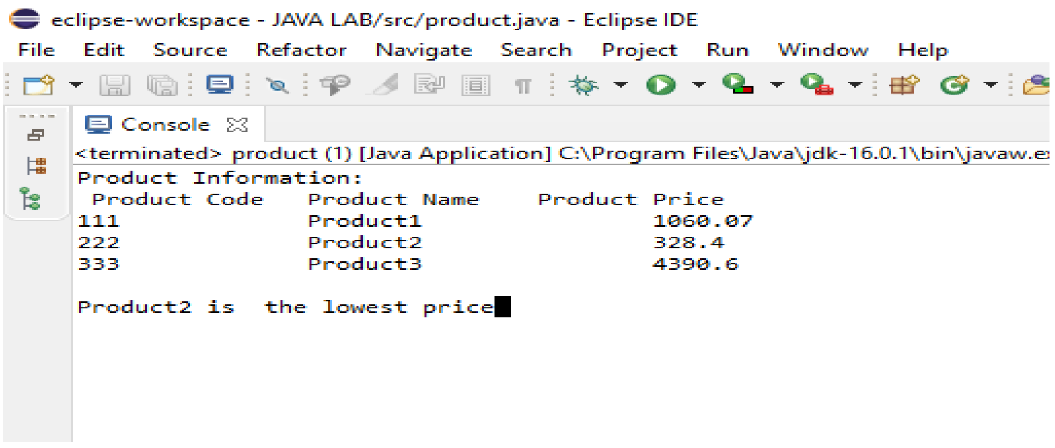
```
public class product{
                int pcode;
                String pname;
                double price;
                double lowest;
                void data(int c, String n, double p){
                   pcode=c;
                   pname=n;
                   price=p;
                }
                void display(){

System.out.println(pcode+"\t\t"+pname+"\t\t"+price);

                }
                static void findLowest(double price1,double
price2, double price3){
                if(price1<=price2 && price1<=price3){
                   System.out.println("\nProduct1 is  the lowest
price");

                }
                else if(price2<=price1 && price2<=price3){
                   System.out.println("\nProduct2 is  the lowest
price");
```

```
            }
            else{
                System.out.println("\nProduct3 is  the lowest
price");

            }

          }
          public static void main(String[] args){
                product obj1 = new product();
                product obj2 = new product();
                product obj3 = new product();
                obj1.data(111,"Product1",1060.07);
                obj2.data(222,"Product2",328.40);
                obj3.data(333,"Product3",4390.60);
                System.out.println("Product Information:\n
Product Code\tProduct Name\tProduct Price");
                obj1.display();
                obj2.display();
                obj3.display();
                findLowest(obj1.price,obj2.price,obj3.price);

            }
        }
```

RESULT

The above program is executed and obtained the output

OUTPUT


eclipse-workspace - JAVA LAB/src/product.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console

```
<terminated> product (1) [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.e
Product Information:
 Product Code     Product Name     Product Price
111              Product1                 1060.07
222              Product2                 328.4
333              Product3                 4390.6

Product2 is  the lowest price
```

PROGRAM 2

AIM

Read two matrices from the console and perform matrix addition.

ALGORITHM

```
STEP 1: Start
STEP 2: Declare matrix A[r][c];and matrix B[r][c];and matrix C[r][c]; r= no. of rows, c= no. of columns
STEP 3: Read r, c, A[][] and B[][]
STEP 4: Declare variable i=0, j=0
STEP 5: Repeat until i < r
        5.1: Repeat until j < c
             C[i][j] = A[i][j] + B[i][j]
             Set j=j+1
        5.2: Set i=i+1
STEP 6: C is the required matrix after addition
STEP 7: Stop
```

PROGRAM CODE

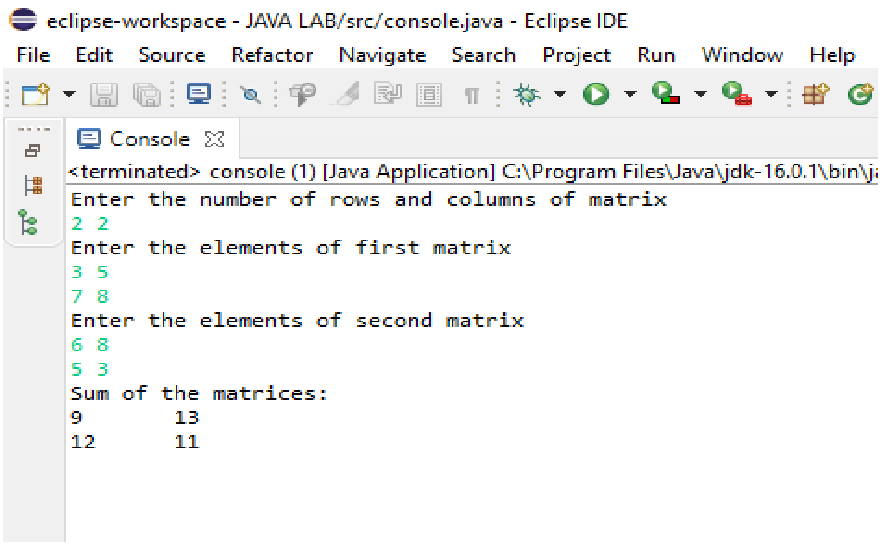
```
import java.util.Scanner;
class console
{
  public static void main(String args[])
  {
    int m, n, c, d;
    Scanner in = new Scanner(System.in);

    System.out.println("Enter the number of rows and
columns of matrix");
    m = in.nextInt();
    n = in.nextInt();

    int first[][] = new int[m][n];
    int second[][] = new int[m][n];
    int sum[][] = new int[m][n];

    System.out.println("Enter the elements of first
matrix");

    for (c = 0; c < m; c++)
```

```
      for (d = 0; d < n; d++)
        first[c][d] = in.nextInt();

    System.out.println("Enter the elements of second
matrix");

    for (c = 0 ; c < m; c++)
      for (d = 0 ; d < n; d++)
        second[c][d] = in.nextInt();

    for (c = 0; c < m; c++)
      for (d = 0; d < n; d++)
        sum[c][d] = first[c][d] + second[c][d];  //replace
'+' with '-' to subtract matrices

    System.out.println("Sum of the matrices:");

    for (c = 0; c < m; c++)
    {
      for (d = 0; d < n; d++)
        System.out.print(sum[c][d] + "\t");

      System.out.println();
    }
  }
}
```

RESULT

The above program is successfully executed and obtains the output.

OUTPUT



eclipse-workspace - JAVA LAB/src/console.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console

```
<terminated> console (1) [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\j
Enter the number of rows and columns of matrix
2 2
Enter the elements of first matrix
3 5
7 8
Enter the elements of second matrix
6 8
5 3
Sum of the matrices:
9       13
12      11
```

PROGRAM 3

AIM

Add complex numbers.

ALGORITHM

STEP 1: Start
STEP 2: Create a class with 2 data members and 2 functions.
STEP 3: First function is used to add values to variables.
STEP 4: Second function is used to add the complex numbers and return the value.
STEP 5: Define object to call the function and Print the result.
STEP 6: Stop

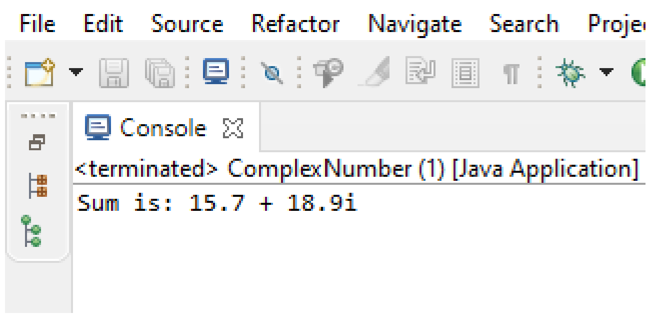
PROGRAM CODE

```
public class ComplexNumber{
  double real, img;

 ComplexNumber(double r, double i){
          this.real = r;
          this.img = i;
  }

  public static ComplexNumber
sum(ComplexNumber c1, ComplexNumber
c2)
  {

     ComplexNumber temp = new
ComplexNumber(0, 0);

     temp.real = c1.real + c2.real;
     temp.img = c1.img + c2.img;


     return temp;
   }
   public static void main(String args[]) {
          ComplexNumber c1 = new
ComplexNumber(6.8, 9);
          ComplexNumber c2 = new
ComplexNumber(8.9,9.9);
     ComplexNumber temp = sum(c1, c2);
     System.out.printf("Sum is: "+
temp.real+" + "+ temp.img +"i");
   }
}
```

RESULT

The above program is successfully executed and obtains the output.

OUTPUT



```
<terminated> ComplexNumber (1) [Java Application]
Sum is: 15.7 + 18.9i
```

PROGRAM 4

AIM

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

ALGORITHM

STEP 1: Start

STEP 2: Read a matrix using for loop.

STEP 3: Check the number of rows and columns are the same. If its same;

STEP 4: Check the symmetric elements are the same. If its same;

STEP 5: Print the matrix and Print its True.

STEP 6: Else print its false.

STEP 7: Stop

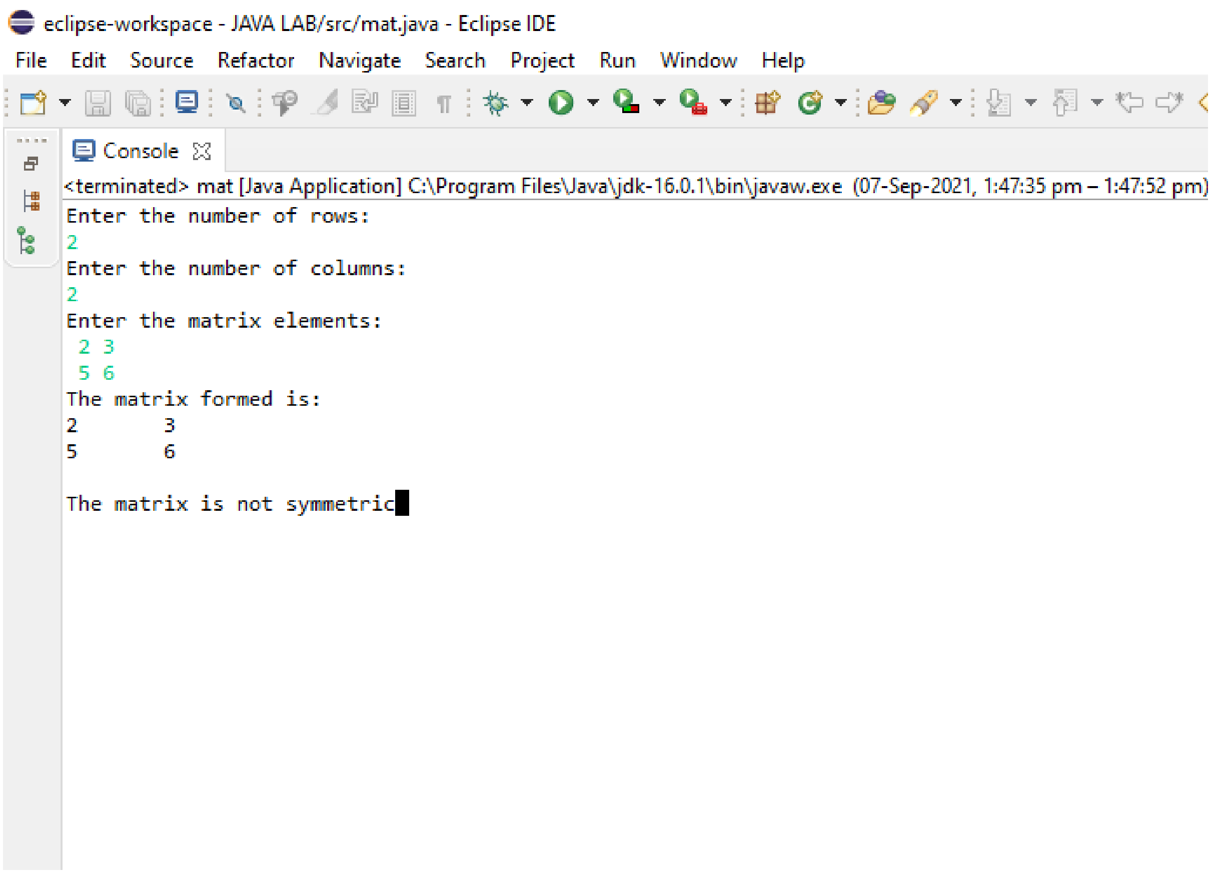
PROGRAM CODE

```
import java.util.*;
public class mat {
                        static void
checkSymmetric(int mat[][], int row,

                        int col)
            {
                        int i, j, flag = 1;
                        System.out.println("The
matrix formed is:");
                        for (i = 0; i < row; i++) {
                                    for (j = 0; j < col;
j++) {

System.out.print(mat[i][j] + "\t");
                                    }

System.out.println("");
                        }
int[][] transpose = new int[row][col];
for (i = 0; i < row; i++) {
        for (j = 0; j < col; j++) {
transpose[j][i] = mat[i][j];
                                    }
                        }
if (row == col) {
    for (i = 0; i < row; i++) {
        for (j = 0; j < col; j++) {
              if (mat[i][j] != transpose[i][j]) {
                flag = 0;
        break;
                                        }}

        if (flag == 0) {
System.out.print("\nThe matrix is not
symmetric");

break;
                                }
                        }
if (flag == 1) {
System.out.print("\nThe matrix is symmetric");
                        }
                }
else {
  System.out.print("\nThe matrix is not
symmetric");
                }
        }
public static void main(String args[])
        {

                Scanner sc = new
Scanner(System.in);

        int i, j, row, col, flag = 1;

System.out.print("Enter the number of rows:");
row = sc.nextInt();

System.out.print("Enter the number of
columns:");
        col = sc.nextInt();
                int[][] mat = new
int[row][col];

System.out.println("Enter the matrix elements:");

for (i = 0; i < row; i++) {
     for (j = 0; j < col; j++) {
mat[i][j] = sc.nextInt();
                        }
                }

        checkSymmetric(mat, row, col);
        }
}
```

RESULT

The above program is successfully executed and obtains the output.

OUTPUT


eclipse-workspace - JAVA LAB/src/mat.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console

```
<terminated> mat [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (07-Sep-2021, 1:47:35 pm – 1:47:52 pm)
Enter the number of rows:
2
Enter the number of columns:
2
Enter the matrix elements:
 2 3
 5 6
The matrix formed is:
2       3
5       6

The matrix is not symmetric
```

PROGRAM 5

AIM

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.

ALGORITHM

STEP 1: Start

STEP 2: Create a class CPU with members as price and a class processor.

STEP 3: Class processors contain members as cores, manufacture and nested class Ram.

STEP 5: Class Ram contains members as memory and manufactures.

STEP 6: Create objects for each class and Print its details.

STEP 7: Stop

PROGRAM CODE

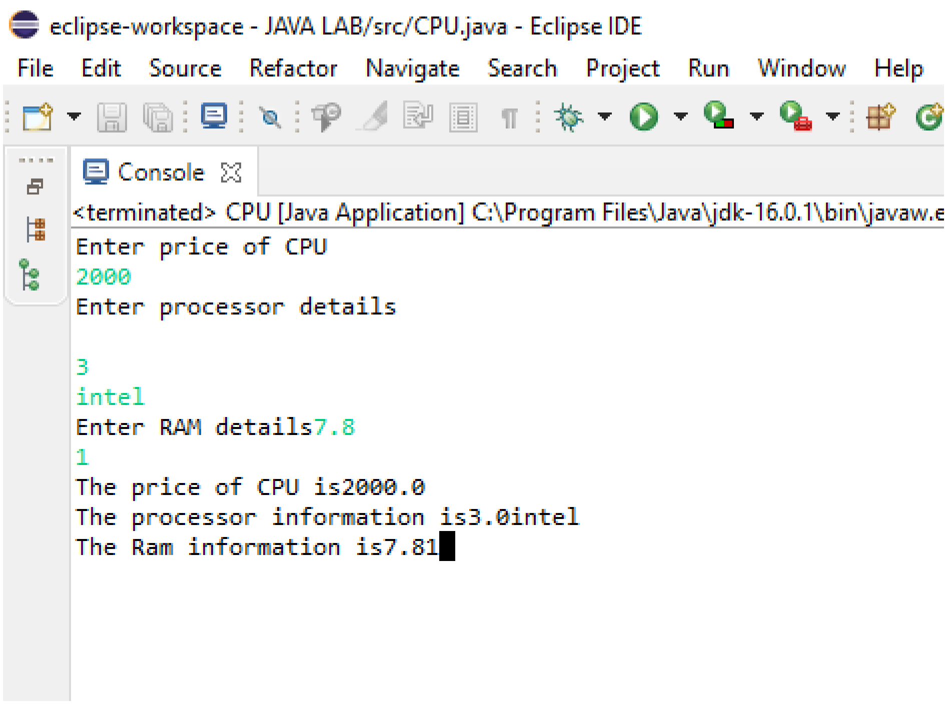
```
import java.util.Scanner;
import java.lang.String;

public class CPU {
        double price;
  public class processor{
          float ncores;
          String manufacturer;
          void pinfo(float a,String
processorname) {
               ncores=a;
               manufacturer=processorname;
          System.out.println("The processor
information is" +ncores+ "" +manufacturer);
  }
  }
  static class ram{
          float memory;
          String manufacturer;
          void prinfo(float b,String ramname) {
                  memory=b;
                  manufacturer=ramname;

          System.out.println("The Ram
information is" +memory+ "" +manufacturer);

  }
  }
```

```
	public static void main(String[] args) {
		CPU obj=new CPU();
		CPU.processor obj1=obj.new
processor();
		CPU.ram obj2=new CPU.ram();
		Scanner sc=new
Scanner(System.in);
		System.out.println("Enter price
of CPU");
		obj.price=sc.nextInt();
		System.out.println("Enter
processor details");
		float a=sc.nextFloat();
		Scanner sc1=new
Scanner(System.in);
		String
processorname=sc1.nextLine();
		System.out.print("Enter RAM
details");
		float b=sc.nextFloat();
		String
ramname=sc1.nextLine();
		sc.close();
		sc1.close();
		System.out.println("The price
of CPU is"+obj.price);
		obj1.pinfo(a, processorname);
		obj2.prinfo(b, ramname);
	}

}
```

RESULT

The above program is successfully executed and obtains the output.

OUTPUT


eclipse-workspace - JAVA LAB/src/CPU.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console

```
<terminated> CPU [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.e
Enter price of CPU
2000
Enter processor details

3
intel
Enter RAM details7.8
1
The price of CPU is2000.0
The processor information is3.0intel
The Ram information is7.81
```

LABCYCLE 2

PROGRAM 1

AIM

Program to Sort strings.

ALGORITHM

```
STEP 1: Start
STEP 2.Enter the limit n
STEP 3: Enter n  String
STEP 4: Enter the names with for loop
STEP 5: for i = 0 ;i < n ;i++ then
STEP 6: if names[i].compareTo(names[j])>0)
STEP 7: swap temp,names[i] & names[i],names[j] & names[j],temp
STEP 8:Print the array
STEP 9: Stop
```

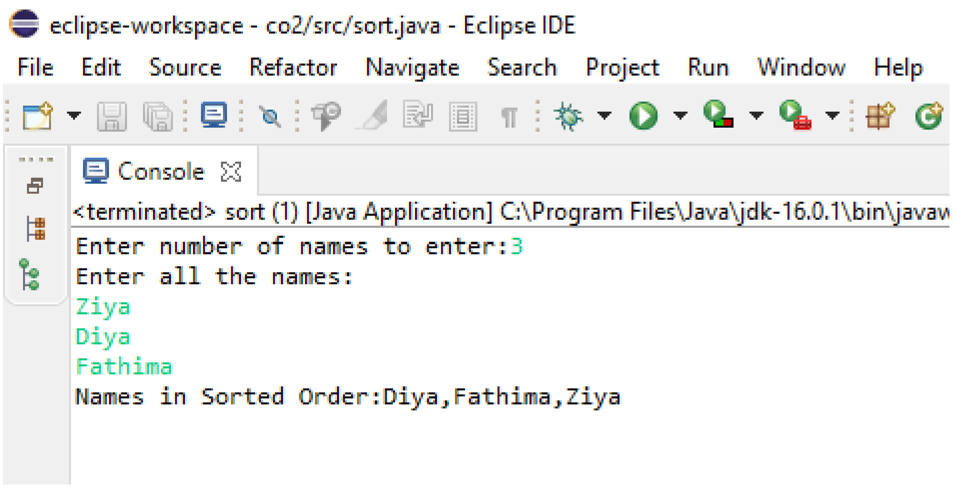
PROGRAM CODE

```java
import java.util.Scanner;
public class sort
{
    public static void main(String[] args)
    {
        int n;
        String temp;
        Scanner s = new Scanner(System.in);
        System.out.print("Enter number of names to
enter:");
        n = s.nextInt();
        String names[] = new String[n];
        Scanner s1 = new Scanner(System.in);
        System.out.println("Enter all the names:");
        for(int i = 0; i < n; i++)
        {
            names[i] = s1.nextLine();
        }
        for (int i = 0; i < n; i++)
        {
            for (int j = i + 1; j < n; j++)
            {
                if (names[i].compareTo(names[j])>0)
                {
                    temp = names[i];
                    names[i] = names[j];
                    names[j] = temp;
                }
            }
        }
        System.out.print("Names in Sorted Order:");
        for (int i = 0; i < n - 1; i++)
        {
            System.out.print(names[i] + ",");
        }
        System.out.print(names[n - 1]);
    }
}
```

RESULT

The above program is successfully executed and obtains the output.

OUTPUT


eclipse-workspace - co2/src/sort.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console

```
<terminated> sort (1) [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw
Enter number of names to enter:3
Enter all the names:
Ziya
Diya
Fathima
Names in Sorted Order:Diya,Fathima,Ziya
```

PROGRAM 2

AIM

Search an element in an array.

ALGORITHM

```
STEP 1: Start
STEP 2: Enter limit n
STEP 3: Enter  n elements to ar[]
STEP 4: Enter searching key
STEP 5: increment i until i<n
STEP 6: if a[i]==key
STEP 7: Print element found
STEP 8: Else Print not found
STEP 9: Stop
```

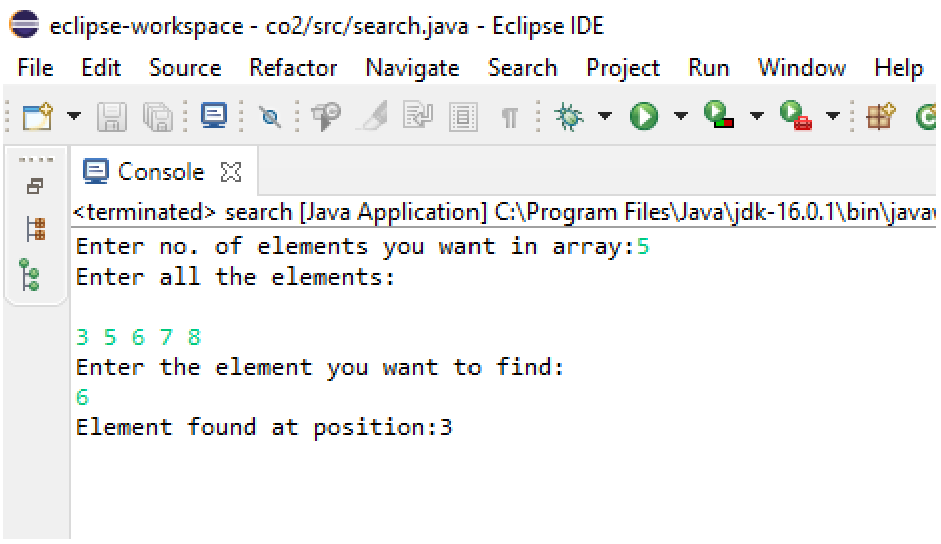
PROGRAM CODE

```
import java.util.Scanner;
public class search
{
   public static void main(String[] args)
   {
      int n, x, flag = 0, i = 0;
      Scanner s = new Scanner(System.in);
      System.out.print("Enter no. of elements you want in array:");
      n = s.nextInt();
      int a[] = new int[n];
      System.out.println("Enter all the elements:");
      for(i = 0; i < n; i++)
      {
         a[i] = s.nextInt();
      }
      System.out.print("Enter the element you want to find:");
      x = s.nextInt();
      for(i = 0; i < n; i++)
      {
         if(a[i] == x)
         {
                flag = 1;
                break;
            }
            else
            {
                flag = 0;
            }
        }
        if(flag == 1)
        {
            System.out.println("Element found
at position:"+(i + 1));
        }
        else
        {
            System.out.println("Element not
found");
        }
    }
}
```

RESULT

The above program is successfully executed and obtained the output

OUTPUT


eclipse-workspace - co2/src/search.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console

```
<terminated> search [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw
Enter no. of elements you want in array:5
Enter all the elements:

3 5 6 7 8
Enter the element you want to find:
6
Element found at position:3
```

PROGRAM 3

AIM

Perform string manipulations.

ALGORITHM

STEP 1: Start

STEP 2: Enter string 1,string 2 and string 3

STEP 3: Perform string operations

STEP 4: Display the output

STEP 5: Stop

PROGRAM CODE

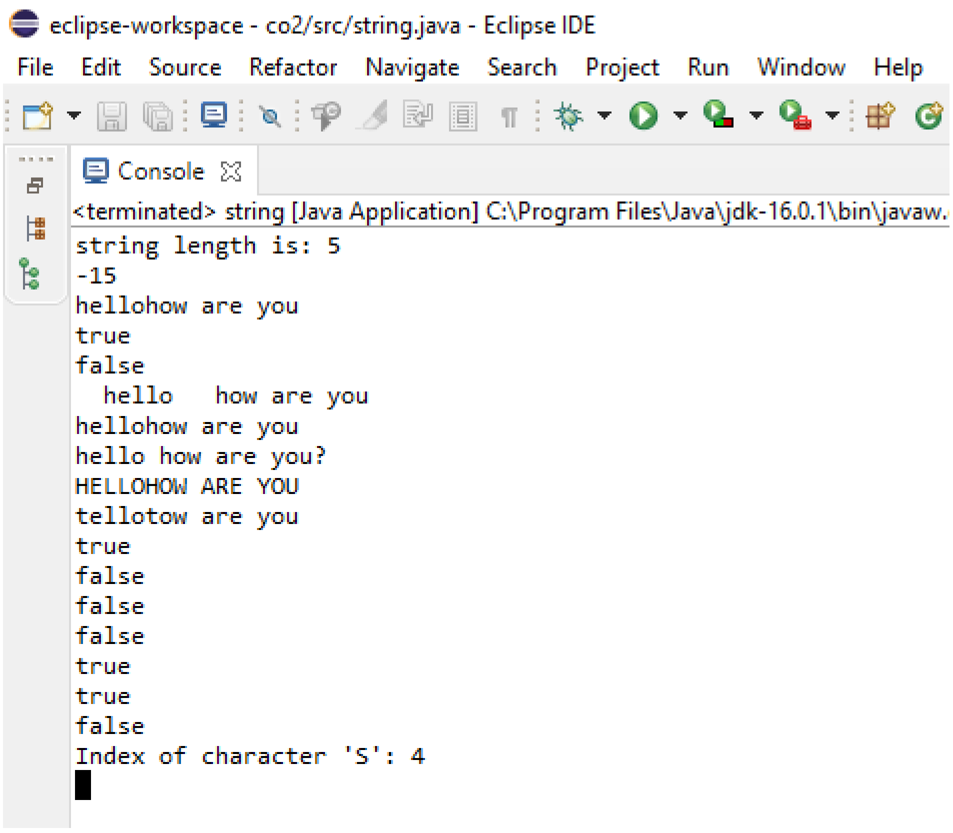
```
public class string {

        public static void main(String[] args) {
                String s1="hello";
                String s2="whatsup";

                //1.LENGTH
                System.out.println("string length is:
"+s1.length());
                //2.COMPARETO
              System.out.println(s1.compareTo(s2));
                //3.CONCATENATION
                s1=s1.concat("how are you");
                System.out.println(s1);
                //4.ISEMPTY
                String s3="";
                System.out.println(s3.isEmpty());
                System.out.println(s1.isEmpty());
                //5.TRIM
                String s4="  hello   ";
                System.out.println(s4+"how are
you");
             System.out.println(s4.trim()+"how are
you");
                //6.LOWERCASE
                String s5="HELLO HOW Are
You?";
                String s5lower=s5.toLowerCase();
                System.out.println(s5lower);
```

```
                    //7.UPPERCASE
                    String s1upper=s1.toUpperCase();
                    System.out.println(s1upper);
                    //8.REPLACE
                    String
replaceString=s1.replace('h','t');
                    System.out.println(replaceString);
                    //9.CONTAINS
                    String name=" hello how are you
doing";
System.out.println(name.contains("how are you"));
System.out.println(name.contains("fine"));
                  //10.EQUALS
             String s6="hello";
System.out.println(s1.equalsIgnoreCase(s2));
System.out.println(s1.equalsIgnoreCase(s6));
                 //11.ENDSWITH
             String s7="hello how are you";
             System.out.println(s7.endsWith("u"));
             System.out.println(s7.endsWith("you"));
             System.out.println(s7.endsWith("how"));
              //12.INDEXOF
             String s8 = "RockStar";
             System.out.println("Index of character
'S': " + s8.indexOf('S'));
             }

}
```

RESULT

The above program is successfully executed and obtained the output

OUTPUT


eclipse-workspace - co2/src/string.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console

```
<terminated> string [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.
string length is: 5
-15
hellohow are you
true
false
   hello    how are you
hellohow are you
hello how are you?
HELLOHOW ARE YOU
tellotow are you
true
false
false
false
true
true
false
Index of character 'S': 4
```

PROGRAM 4

AIM

Program to create a class for employee having attributes eNo,eName,eSalary. Read n employ information and search for an employee given eNo,using the concept of Array of Objects.

ALGORITHM

STEP 1: START

STEP 2: Enter the limit

STEP 3: Enter the details of employee

STEP 4: Assign the value using object

STEP 5: Display the value using object

STEP 6: Print the details

STEP 7: Stop

PROGRAM CODE

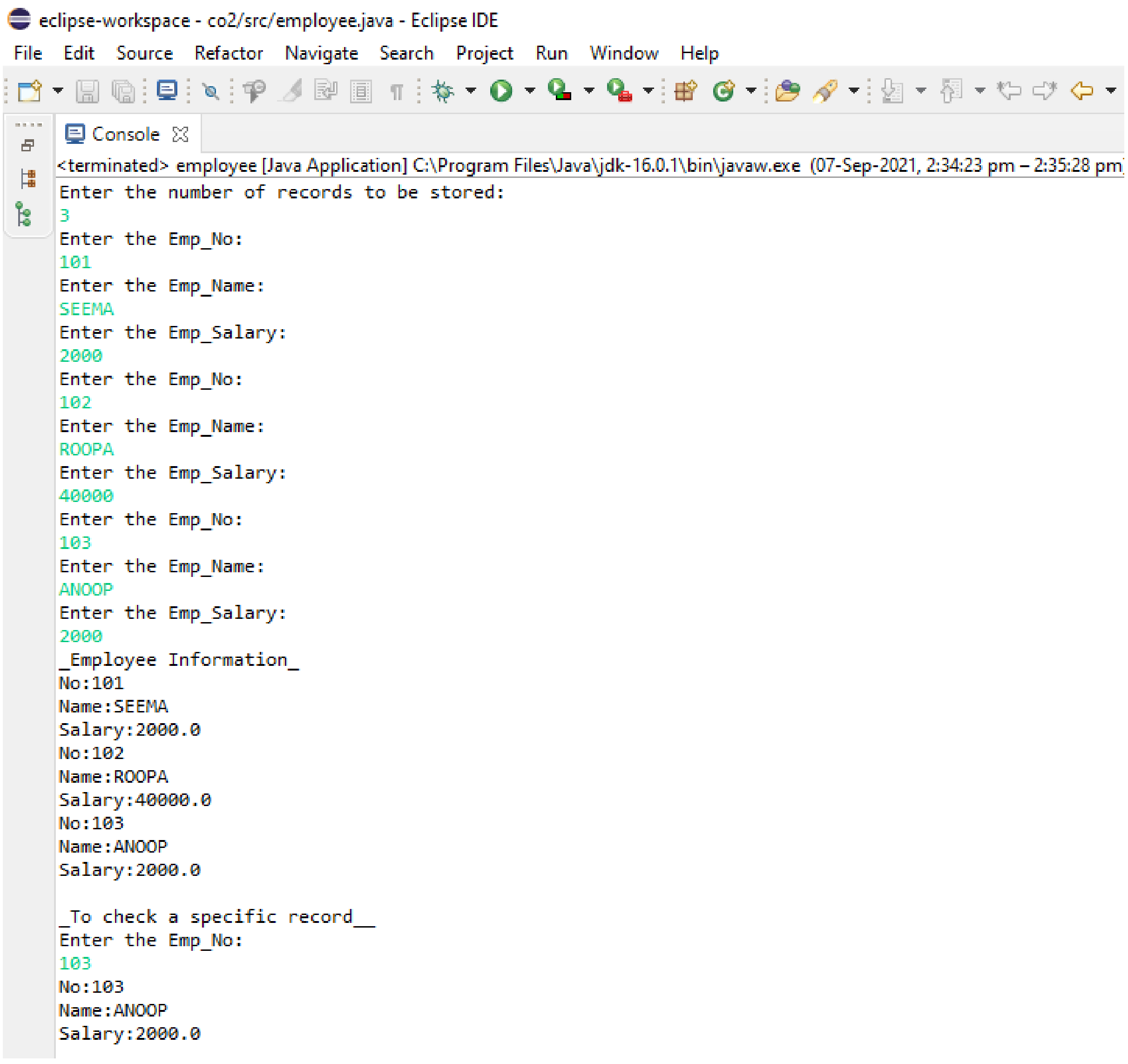
```
import java.util.*;
public class employee{

        int[] eNo = new int[20];
        int count,i,e;
        String[] eName = new String[50];
        float[] eSalary = new float[20];


        void getinfo(int c){
          Scanner s = new Scanner(System.in);
          count=c;
           for(i=0;i<c;i++){
              System.out.println("Enter the
Emp_No:");
              eNo[i]=s.nextInt();
              System.out.println("Enter the
Emp_Name:");
              eName[i]=s.next();
              System.out.println("Enter the
Emp_Salary:");
              eSalary[i]=s.nextFloat();
          }
        }
 void printinfo(int c){
          count =c;
```

```
                System.out.println("_Employee
Information_");
                for(i=0;i<count;i++)
                {
                  System.out.println("No:"+eNo[i]);
                  System.out.println("Name:"+eName[i]);

System.out.println("Salary:"+eSalary[i]);
                }
            }

            void displayinfo(int emp_no, int c) {
                  int flag=0;
                  e = emp_no;
                  count = c;
                  for(i=0;i<count;i++)
                  {
                           if(eNo[i]==e)
                           {
System.out.println("No:"+eNo[i]);

System.out.println("Name:"+eName[i]);

System.out.println("Salary:"+eSalary[i]);
                        flag++;
                           }

                  }
                  if(flag==0)
                           System.out.println("Record Not
Found!");
            }

            public static void main(String[] args){
                employee obj = new employee();
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the number of
records to be stored:");
                obj.count = sc.nextInt();
                obj.getinfo(obj.count);
                obj.printinfo(obj.count);
                System.out.println("\n_To check a specific
record__");
                System.out.println("Enter the Emp_No:");
                int e = sc.nextInt();
                obj.displayinfo(e,obj.count);
                sc.close();
            }  }
```

RESULT

The above program is successfully executed and obtains the output.

OUTPUT


eclipse-workspace - co2/src/employee.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console

```
<terminated> employee [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (07-Sep-2021, 2:34:23 pm – 2:35:28 pm)
Enter the number of records to be stored:
3
Enter the Emp_No:
101
Enter the Emp_Name:
SEEMA
Enter the Emp_Salary:
2000
Enter the Emp_No:
102
Enter the Emp_Name:
ROOPA
Enter the Emp_Salary:
40000
Enter the Emp_No:
103
Enter the Emp_Name:
ANOOP
Enter the Emp_Salary:
2000
_Employee Information_
No:101
Name:SEEMA
Salary:2000.0
No:102
Name:ROOPA
Salary:40000.0
No:103
Name:ANOOP
Salary:2000.0

_To check a specific record__
Enter the Emp_No:
103
No:103
Name:ANOOP
Salary:2000.0
```