LAB CYCLE 1

SUBMITTED BY,
SIJI JOSE
20MCA237

PROGRAM -1

AIM: To define a class ‘product’ with data members as pcode, pname and price. Create 3 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 which product has the lowest price using an if-else statement.
STEP 5: Print the details of the product.
STEP 6: Stop

PROGRAM CODE

product.java

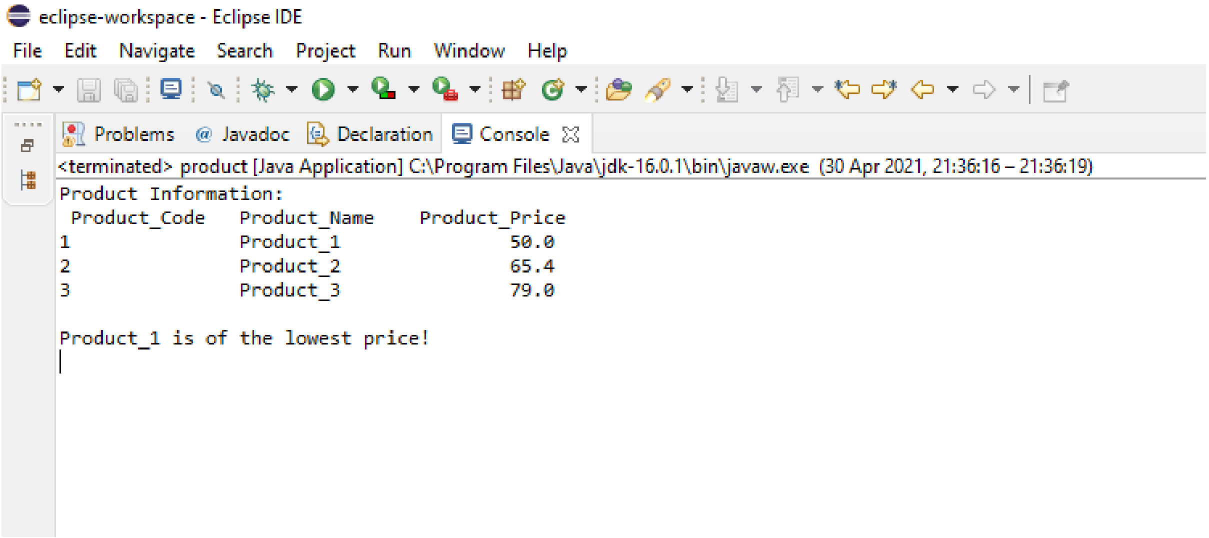
```
public class product{
            int pcode;
            String pname;
            double price;
            double lowest;
            void data(int c, String n, double p){
               pcode=c;
               pname=n;
               price=p;
            }
            void display(){

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("\nProduct_1 is of the lowest
price!");

            }
            else if(price2<=price1 && price2<=price3){
               System.out.println("\nProduct_2 is of the lowest
price!");

            }
```

```
                else{
                    System.out.println("\nProduct_3 is of the lowest
price!");

                }

               }
              public static void main(String[] args){
                    product obj1 = new product();
                    product obj2 = new product();
                    product obj3 = new product();
                    obj1.data(1,"Product_1",50.0);
                    obj2.data(2,"Product_2",65.40);
                    obj3.data(3,"Product_3",79.00);
                    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 obtains the output.

OUTPUT



eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Problems
Javadoc
Declaration
Console

<terminated> product [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (30 Apr 2021, 21:36:16 – 21:36:19)

```
Product Information:
 Product_Code    Product_Name     Product_Price
1               Product_1               50.0
2               Product_2               65.4
3               Product_3               79.0

Product_1 is of the lowest price!
```

PROGRAM -2

AIM: To read 2 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

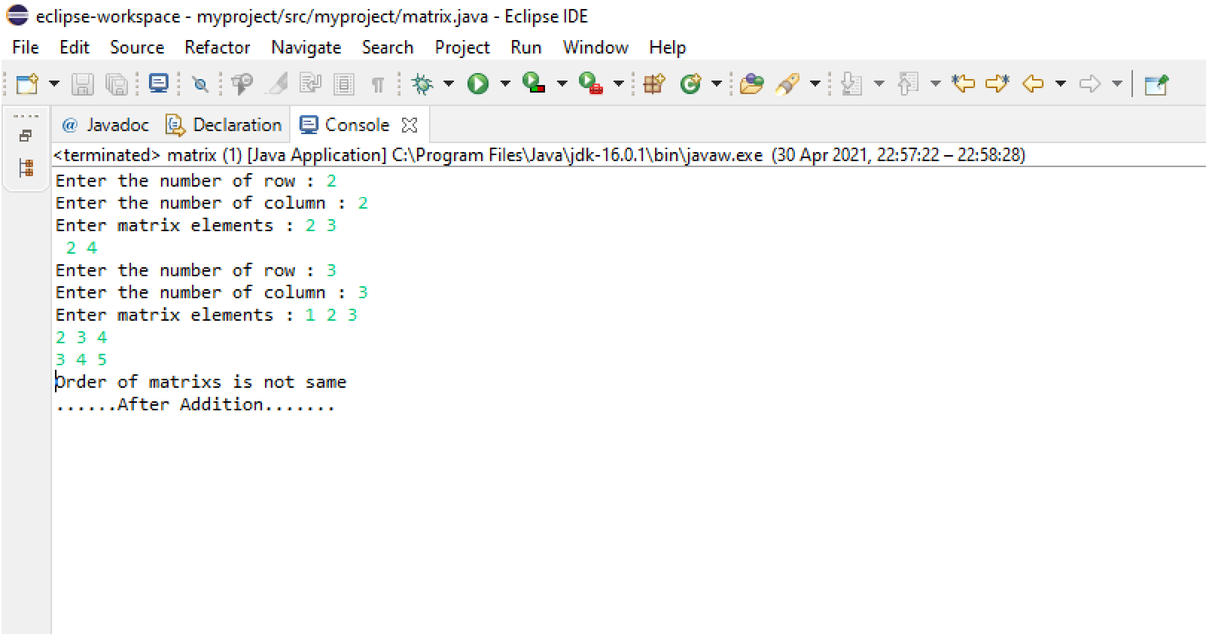
matrix.java

```java
package myproject;
import java.util.*;
public class matrix {
 int row;
 int column;
 int[][] array = new int[10][10];
 public void get_metrix(){
 int rc,cc;
 Scanner sc= new Scanner(System.in);
 System.out.print("Enter the number of row : ");
 this.row = sc.nextInt();
 System.out.print("Enter the number of column : ");
 this.column = sc.nextInt();
 System.out.print("Enter matrix elements : ");
 for(rc=0;rc<this.row;c++){
 for(cc=0;cc<this.column;c++){
 this.array[rc][cc] = sc.nextInt();
 }
 }
 }
 public static matrix sum(matrix c1, matrix c2) {
 int rc, cc;
 matrix temp = new matrix();
 if (c1.row == c2.row && c1.column == c2.column) {
 temp.row =c1.row;
 temp.column = c1.column;
 for (rc = 0; rc < c1.row; c++) {
 for (cc = 0; cc < c1.column; c++) {
 temp.array[rc][cc] = c1.array[rc][cc] + c2.array[rc][cc];
 }
 }
 }
 else {
 System.out.println("Order of matrices is not same ");
 }
 return temp;
 }
 public void display_matrix(){
 int rc,cc;
 for(rc=0;rc<this.row;c++){
 for(cc=0;cc<this.column;c++){
 System.out.print(this.array[rc][cc] + "\t" );

 }
 System.out.println("");
 }
 }
 public static void main(String[] args) {
 matrix first = new matrix();
 matrix second = new matrix();
 matrix temp = sum(first, second);
 first.get_metrix();
 second.get_metrix();
 temp = sum(first,second);
 //first.display_matrix();
 //second.display_matrix();
 System.out.println("......After Addition.......");
 temp.display_matrix();

 }
}
```

RESULT

The above program is executed and obtains the output.

OUTPUT



eclipse-workspace - myproject/src/myproject/matrix.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Javadoc
Declaration
Console

```
<terminated> matrix (1) [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (30 Apr 2021, 22:57:22 – 22:58:28)
Enter the number of row : 2
Enter the number of column : 2
Enter matrix elements : 2 3
 2 4
Enter the number of row : 3
Enter the number of column : 3
Enter matrix elements : 1 2 3
2 3 4
3 4 5
Order of matrixs is not same
......After Addition.......
```

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

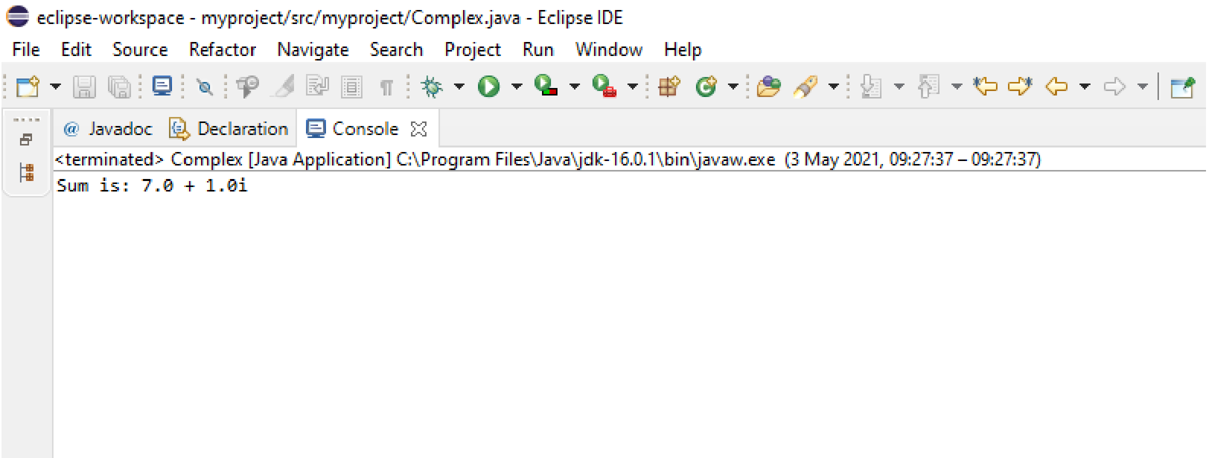
complex.java

```
package myproject;
public class Complex{
   double real, img ;
   Complex(double r, double i){
            this.real = r;
            this.img = i;
   }
   public static Complex sum(Complex c1, Complex c2)
   {
        Complex temp = new Complex(0, 0);
        temp.real = c1.real + c2.real;
        temp.img = c1.img + c2.img;
        return temp;
    }
    public static void main(String args[]) {
            Complex c1 = new Complex(6, 2);
            Complex c2 = new Complex(1, -1);
        Complex temp = sum(c1, c2);
        System.out.printf("Sum is: "+ temp.real+" + "+ temp.img
+"i");
    }
}
```

RESULT

The above program is executed and obtains the output.

OUTPUT


eclipse-workspace - myproject/src/myproject/Complex.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Javadoc
Declaration
Console

```
<terminated> Complex [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (3 May 2021, 09:27:37 – 09:27:37)
Sum is: 7.0 + 1.0i
```

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 a 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

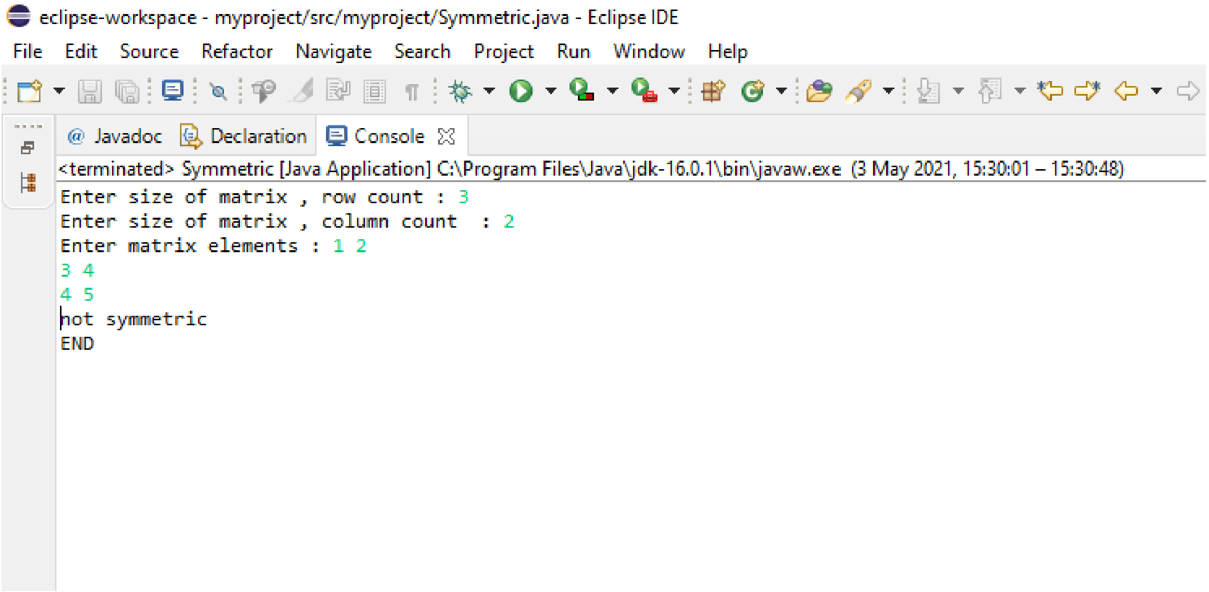
symmetric.java

```
package myproject;
import java.util.*;
public class Symmetric{
    int row;
    int column;
    int[][] array = new int[10][10];

    public void get_metrix() {
        int rc, cc;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of matrix , row count : ");
        this.row = sc.nextInt();
        System.out.print("Enter size of matrix , column count  : ");
        this.column = sc.nextInt();
        System.out.print("Enter matrix elements : ");
        for (rc = 0; rc < this.row; rc++) {
            for (cc = 0; cc < this.column; cc++) {
                this.array[rc][cc] = sc.nextInt();
            }
        }
    }

    public void is_symmetric() {
        int rc, cc, flag = 0;
        int[][] transpose = new int[10][10];

        for (rc = 0; rc < this.row; rc++) {
            for (cc = 0; cc < this.column; cc++) {
                transpose[cc][rc] = array[rc][cc];
            }
        }

        for (rc = 0; rc < this.row; rc++) {
            for (cc = 0; cc < this.column; cc++) {
                if (this.array[rc][cc] != transpose[rc][cc]) {
                    flag = 1;
                }
            }
        }
        if (flag == 0) {
            System.out.println("symmetric");
        } else {
            System.out.println("not symmetric");
        }

    }

    public static void main(String[] args) {
        Symmetric first = new Symmetric();
        first.get_metrix();
        first.is_symmetric();
        System.out.println("END");
    }
}
```

RESULT

The above program is executed and obtains the output.

OUTPUT



eclipse-workspace - myproject/src/myproject/Symmetric.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Javadoc
Declaration
Console

```
<terminated> Symmetric [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (3 May 2021, 15:30:01 – 15:30:48)
Enter size of matrix , row count : 3
Enter size of matrix , column count  : 2
Enter matrix elements : 1 2
3 4
4 5
not symmetric
END
```

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

cpu.java

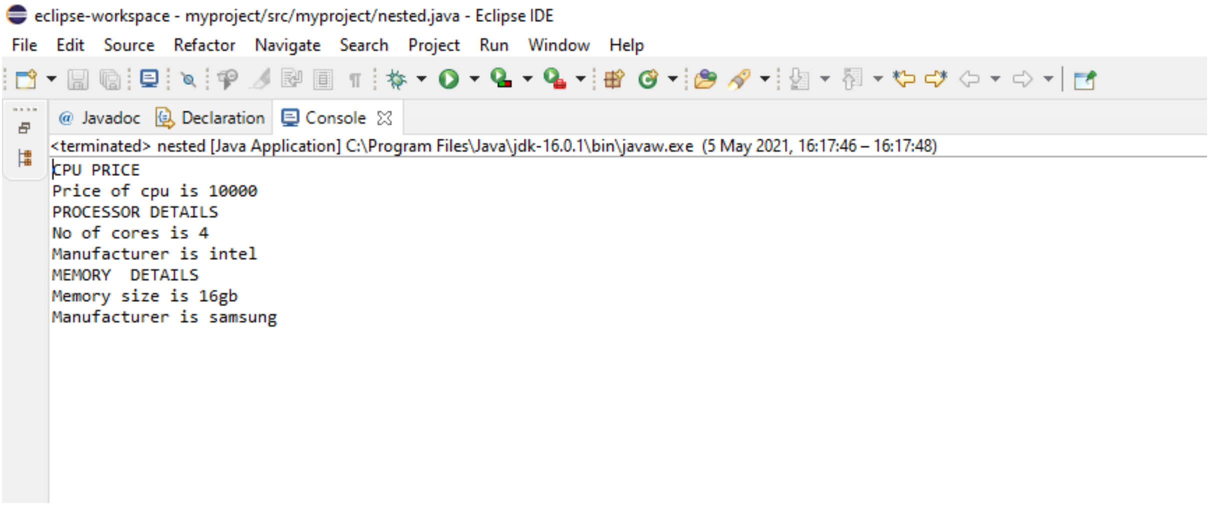
```
package myproject;

class cpu
{
        int price = 10000;
        public class processor
        {
                int cores = 4;
                String manufacturer = "intel";
        }
        static class ram
        {
                int memory size = 16;
                String manufacturer = "samsung";
        }
}
public class nested
{
  public static void main (String[] args)
  {
        System.out.println("CPU PRICE");
        cpu obj1 = new cpu();
        System.out.println("Price of cpu is "+obj1.price);
        System.out.println("PROCESSOR DETAILS");
        cpu.processor obj2 = obj1.new processor();
        System.out.println("No of cores is "+ obj2.cores);
        System.out.println("Manufacturer is "+ obj2.manufacturer);
        System.out.println("MEMORY  DETAILS");
```

	cpu.ram obj3 = new cpu.ram();
	System.out.println("Memory size is "+ obj3.memory size"gb");
	System.out.println("Manufacturer is "+ obj3.manufacturer);
	}
	}

RESULT

The above program is executed and obtains the output.

OUTPUT


eclipse-workspace - myproject/src/myproject/nested.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Javadoc
Declaration
Console

```
<terminated> nested [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (5 May 2021, 16:17:46 – 16:17:48)
CPU PRICE
Price of cpu is 10000
PROCESSOR DETAILS
No of cores is 4
Manufacturer is intel
MEMORY  DETAILS
Memory size is 16gb
Manufacturer is samsung
```

LAB CYCLE 2

PROGRAM -1

AIM: Program to Sort strings

ALGORITHM

STEP 1: Start
STEP 2: Initialize an array of strings
STEP 3: Compare between each string using compareTo() and swap the string elements accordingly.
STEP 4:Print the sorted array of strings
STEP 5: Stop

PROGRAM CODE

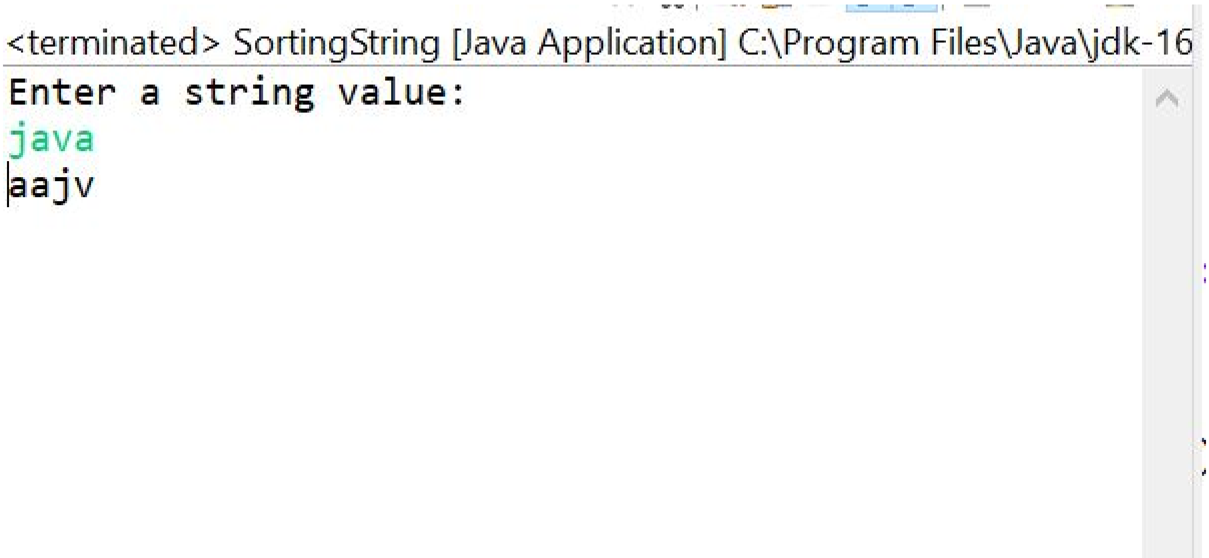
string.java

```
package myproject;

import java.util.Scanner;

public class string {

        public static void main(String[] args) {
                int count;
                String str;
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter the number of strings:
");
                count=sc.nextInt();

                String str_arr[]=new String[count];
                Scanner sc1=new Scanner(System.in);
                System.out.println("Enter the strings: ");
                for(int i=0;i<count;i++)
                {
                        str_arr[i]=sc1.nextLine();
                }
                sc.close();
                sc1.close();

                for(int i=0;i<count;i++)
                {
                        for(int j=i+1;j<count;j++)
                        {

if(str_arr[i].compareTo(str_arr[j])>0)
                                {
                                        str=str_arr[i];
                                        str_arr[i]=str_arr[j];
                                        str_arr[j]=str;
                                }
                        }
                }

                System.out.println("String after sorting: ");
                for(int i=0;i<count;i++)
                {
                        System.out.print(str_arr[i]+",");
                }


        }

}
```

RESULT

The above program is executed and obtains the output.

OUTPUT



```
<terminated> SortingString [Java Application] C:\Program Files\Java\jdk-16
Enter a string value:
java
aajv
```

PROGRAM -2

AIM: Search an element in an array.

ALGORITHM

STEP 1: Start
STEP 2: Traverse the array
STEP 3:Match the key element with array element
STEP 4:If key element is found, return the index position of the array element
STEP 5: If key element is not found, return -1
STEP 6: Stop

PROGRAM CODE

search.java

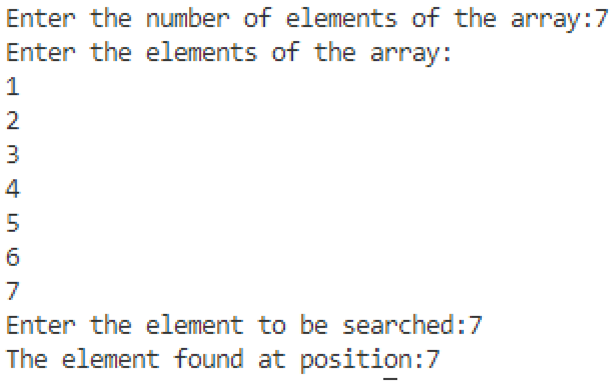
```
          package myproject;

          import java.util.Scanner;

          public class Search
          {
             public static void main(String[] args)
             {
                int n, m, 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:");
                m= s.nextInt();
                for(i = 0; i < n; i++)
                {
                   if(a[i] == m)
                   {
                      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 executed and obtains the output.

OUTPUT



```
Enter the number of elements of the array:7
Enter the elements of the array:
1
2
3
4
5
6
7
Enter the element to be searched:7
The element found at position:7
```

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

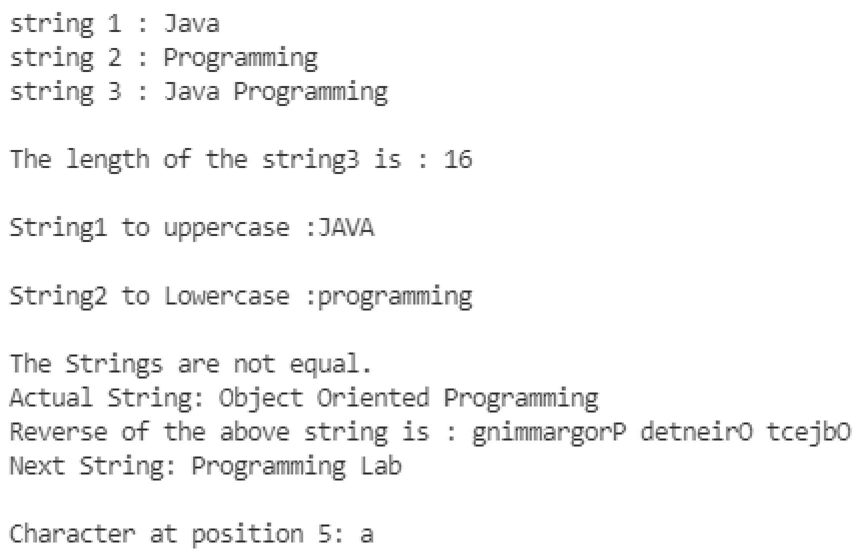
javastring.java

```
package myproject;
public class javastring
{
  public static void main(String[] args)
        {
                String s1="Java";
                String s2=" Programming";
                System.out.println("First string: "+s1+"\nSecond string: "+s2);
                int length1=s1.length();
                System.out.println("Length of First string: "+length1);
                int length2=s2.length();
                System.out.println("Length of First string: "+length2);
                String s3=s1.concat(s2);
                System.out.println(s3);
                System.out.println(s1.isEmpty());
                System.out.println(s1.toLowerCase());
                System.out.println(s2.toUpperCase());
                System.out.println(s3.replace("Java","Object Oriented"));
                System.out.println(s2.endsWith("ing"));
                System.out.println(s1.charAt(2));
                System.out.println(s2.indexOf('n'));

        }

}
```

RESULT

The above program is executed and obtains the output.

OUTPUT



```
string 1 : Java
string 2 : Programming
string 3 : Java Programming

The length of the string3 is : 16

String1 to uppercase :JAVA

String2 to Lowercase :programming

The Strings are not equal.
Actual String: Object Oriented Programming
Reverse of the above string is : gnimmargorP detneirO tcejbO
Next String: Programming Lab

Character at position 5: a
```

PROGRAM -4

AIM: Program to create a class for Employees 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: Define a class Employee with member arrays eNo[], eName[] , eSalary[].
STEP 3: Define the following methods:

- getinfo() to store Employee information.
- printinfo() to display each Employee information according to the Employee No entered by the user.
- displayinfo() to display details of all the employees.

STEP 4:Define a main() to create object of the class and call the above member methods
STEP 5: Stop

PROGRAM CODE

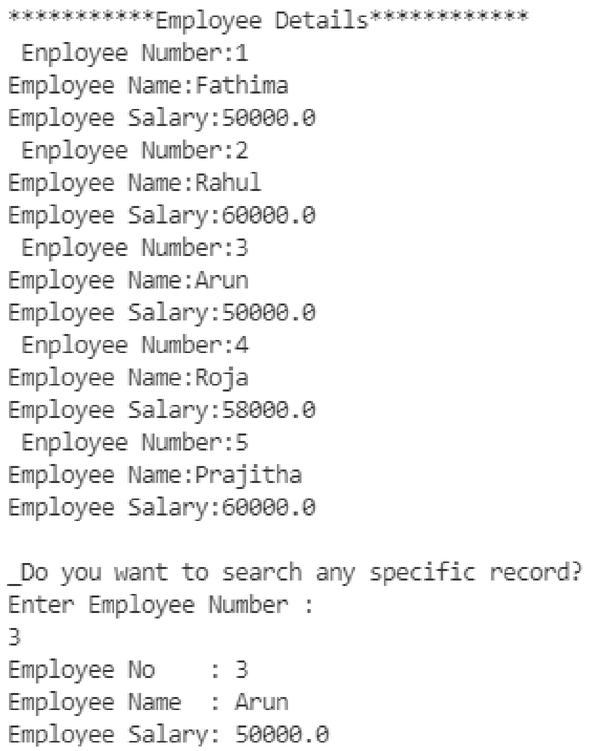
employee.java

```
package myproject;
import java.util.Scanner;
public class employee {
        int empno;
        String empname;
        double empsalary;

        void getinfo()
        {
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter employee number: ");
                empno=sc.nextInt();
                Scanner sc1=new Scanner(System.in);
                System.out.println("Enter employee name: ");
                empname=sc1.nextLine();
                Scanner sc2=new Scanner(System.in);
                System.out.println("Enter employee salary: ");
                empsalary=sc2.nextDouble();
        }
        void display()
        {
                System.out.println("Employee no: "+empno);
                System.out.println("Employee name:
"+empname);
                System.out.println("Salary: "+empsalary);

        }
        public static void main(String args[])
        {
                int n;
                Scanner sc3=new Scanner(System.in);
                System.out.println("Enter the no of employees: ");
                n=sc3.nextInt();
                employee e[]=new employee[n];

                for(int i=0;i<n;i++)
                {
                        e[i]=new employee();
                        e[i].getinfo();
                }
                System.out.println("The employee details are:");
                for(int i=0;i<n;i++)
                {
                        e[i].display();
                }

                int no,flag=0;
                Scanner sc4=new Scanner(System.in);
                System.out.println("Enter employee no to display
details: ");
                no=sc4.nextInt();

                for(int i=0;i<n;i++)
                {
                        if(no==e[i].empno)
                        {
                                e[i].display();
                                flag=1;
                                break;
                        }
                }
                if(flag==0)
                {
                        System.out.println("Not found");
                }
        }
}
```

RESULT

The above program is executed and obtains the output.

OUTPUT



```
***********Employee Details************
 Enployee Number:1
Employee Name:Fathima
Employee Salary:50000.0
 Enployee Number:2
Employee Name:Rahul
Employee Salary:60000.0
 Enployee Number:3
Employee Name:Arun
Employee Salary:50000.0
 Enployee Number:4
Employee Name:Roja
Employee Salary:58000.0
 Enployee Number:5
Employee Name:Prajitha
Employee Salary:60000.0

_Do you want to search any specific record?
Enter Employee Number :
3
Employee No    : 3
Employee Name  : Arun
Employee Salary: 50000.0
```