**OBJECT ORIENTED PROGRAMMING LAB**

**Submitted by,**

**Ananya B**

**S2-MCA**

**Roll NO:208**

**COURSE OUTCOME 1**

**PROGRAM NO 1**

**AIM:** 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.

**ALGORITHM**

Step 1: Start

Step 2: Created a class ‘Product’ with pcode,pname and price as data members

Step 3: Then created a function ‘printChanges’ to display details of product

Step 4: Then in main function created 3 objects,then accessed data members using objects

Step 5: Then compared which product have lowest price using If..else..condition

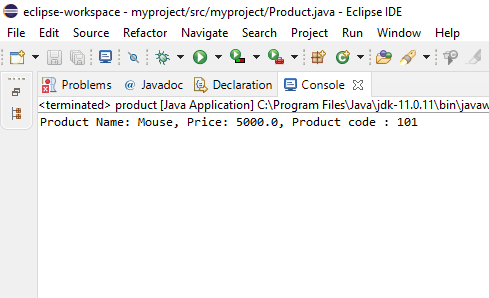
Step 6: Then displayed the details of lower priced product

Step 7: Stop

|  |  |
| --- | --- |
| **PROGRAM CODE:** | package myproject;  public class Product {  String pname;  double price;  int pcode;    void printChanges() {  System.out.println("Product Name: "+pname+", Price: "+price+", Product code : "+pcode);  }  public static void main(String[] args) {    // Create 3 product object  Product product1 = new Product();  Product product2 = new Product();  Product product3= new Product();  // Invoke method on each objects  product1.pname="Keyboard";  product1.price=85000;  product1.pcode=100;    product2.pname="Mouse";  product2.price=5000;  product2.pcode=101;  product3.pname="Monitor";  product3.price=100000;  product3.pcode=103;    if(product1.price<product2.price && product1.price<product3.price) {  product1.printChanges();    }else if (product2.price<product3.price && product2.price<product1.price)  {  product2.printChanges();    }else  {  product3.printChanges();      }      }    } |

**RESULT:** The program is executed successfully and obtained the output

**OUTPUT**



**PROGRAM NO 2**

**AIM:** Read 2 matrices from the console and perform matrix addition

**ALGORITHM**

Step 1: Start

Step 2: Take inputs like no. of rows and colums , and elements of two matrices from user

Step 3: Then the sum of two matrices is done using for loop

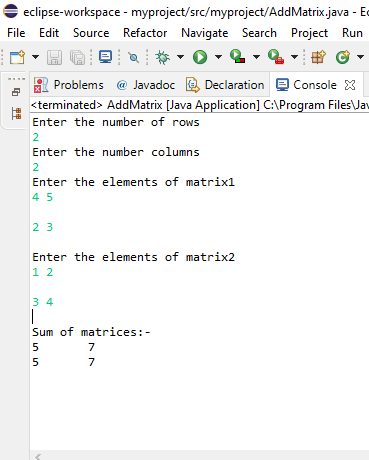
Step 4: Then displayed the resultant matrix as output

Step 5: Stop

|  |  |
| --- | --- |
| **PROGRAM CODE** | package myproject;  import java.util.Scanner;  class AddMatrix  {  public static void main(String args[])  {  int row, col,i,j;  Scanner in = new Scanner(System.in);  System.out.println("Enter the number of rows");  row = in.nextInt();  System.out.println("Enter the number columns");  col = in.nextInt();  int mat1[][] = new int[row][col];  int mat2[][] = new int[row][col];  int res[][] = new int[row][col];  System.out.println("Enter the elements of matrix1");  for ( i= 0 ; i < row ; i++ )  {  for ( j= 0 ; j < col ;j++ )  mat1[i][j] = in.nextInt();  System.out.println();  }  System.out.println("Enter the elements of matrix2");  for ( i= 0 ; i < row ; i++ )  {  for ( j= 0 ; j < col ;j++ )  mat2[i][j] = in.nextInt();  System.out.println();  }  for ( i= 0 ; i < row ; i++ )  for ( j= 0 ; j < col ;j++ )  res[i][j] = mat1[i][j] + mat2[i][j] ;  System.out.println("Sum of matrices:-");  for ( i= 0 ; i < row ; i++ )  {  for ( j= 0 ; j < col ;j++ )  System.out.print(res[i][j]+"\t");  System.out.println();  }  }  } |

**RESULT:** The program is executed successfully and obtained the output

**OUTPUT**



**PROGRAM NO 3**

**AIM:** Add complex numbers

**ALGORITHM**

Step 1: Start

Step 2: Created a constructor for the class ‘CompNum’

Step 3: Created a function to find the sum of complex numbers and

Step 4: Created a function to print the result

Step 5: In main function, created 2 objects c1 and c2 by giving values of real and img part

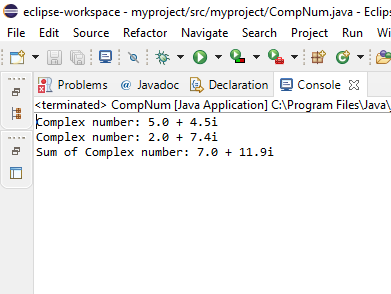
Step 6: Then by calling sum function to find sum and displayed result by calling ‘printComplexNum’ function

Step 7: Stop

|  |  |
| --- | --- |
| **PROGRAM CODE** | package myproject;  public class CompNum {  double real,img;    public CompNum(double real,double img) {  this.real=real;  this.img=img;    }  public static CompNum sum(CompNum c1, CompNum c2) {  CompNum temp = new CompNum(0,0);  temp.real=c1.real+c2.real;  temp.img=c1.img+c2.img;  return temp;  }  void printComplexNum()  {  System.out.println("Complex number: "  + real + " + "  + img + "i");  }  public static void main(String[] args) {  CompNum c1= new CompNum(5,4.5);  c1.printComplexNum();  CompNum c2= new CompNum(2,7.4);  c2.printComplexNum();  CompNum temp=sum(c1,c2);  System.out.print("Sum of ");  temp.printComplexNum();    }  } |

**RESULT:** The program is executed successfully and obtained the output

**OUTPUT**



**PROGRAM NO 4**

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

**ALGORITHM**

Step 1: Start

Step 2: Take a matrix as input from user using console

Step 3: Check whether the given matrix is a square matrix or not

Step 4: Then check whether every element at ith row and jth column is equal to element at jth row and ith column

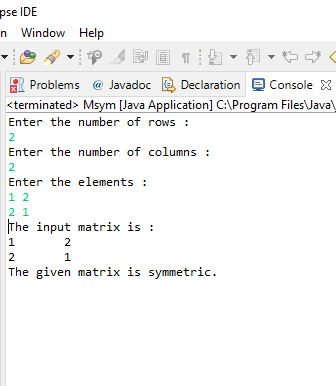
Step 5: If the given matrix satisfy these two conditions, then that matrix is a symmetric matrix, else not symmetric

Step 6: Stop

|  |  |
| --- | --- |
| **PROGRAM CODE** | package myproject;  import java.util.Scanner;  public class Msym {  public static void main(String[] args )  {  Scanner in = new Scanner( System.in);  System.out.println( "Enter the number of rows :");  int rows = in.nextInt();  System.out.println("Enter the number of columns :");  int cols = in.nextInt();  int matrix[][] = new int[rows][cols];  System.out.println("Enter the elements :");  for ( int i = 0; i < rows; i++ )  {  for ( int j = 0; j < cols; j++ )  {  matrix[i][j] = in.nextInt();  }  }  System.out.println("The input matrix is :");  for ( int i = 0; i < rows; i++ )  {  for ( int j = 0; j < cols; j++ )  {  System.out.print(matrix[i][j]+"\t");  }  System.out.println();  }      if ( rows != cols )  {  System.out.println("The given matrix is not a square matrix, so it can't be symmetric.");  }  else  {  boolean symm = true;  for ( int i = 0; i < rows; i++ )  {  for ( int j = 0; j < cols; j++ )  {  if ( matrix[i][j] != matrix[j][i] )  {  symm = false;  break;  }  }  }  if ( symm)  {  System.out.println("The given matrix is symmetric.");  }  else  {  System.out.println("The given matrix is not symmetric.");  }  }  }  } |

**RESULT:** The program is executed successfully and obtained the output

**OUTPUT**



**PROGRAM NO 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: Created a class ‘Cpu’ with price as data member

Step 3: Then created an inner class ‘Processor’ with ‘cores’ and ‘manufact’ as data members, then created a constructor for Processor class

Step 4: And created a display function to print the processor details like cores and manufact

Step 4: Then created a static nested class RAM with memory and manufact as data members, then created a constructor for the ‘RAM ’ class

Step 5: And created a display function to display RAM details and another display function to show price details

Step 6: Then created objects for each class like inter for ‘Cpu’; i\_processor for ‘Processor’;i\_ram for ‘RAM’ and by using these objects acces all details

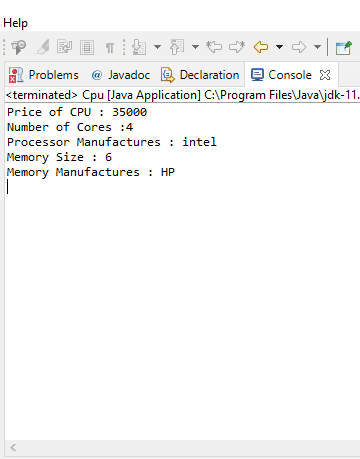
Step 7: And displayed all details using display each functions

Step 8:Stop

|  |  |
| --- | --- |
| **PROGRAM CODE** | package myproject;  public class Cpu {  int price;    Cpu(int p){  this.price=p;  }    class Processor{  int cores;  String manufact;    Processor(int c,String m){  this.cores=c;  this.manufact=m;  }    void display() {  System.out.println("Number of Cores :"+this.cores);  System.out.println("Processor Manufactures : " + this.manufact);  }  }    static class Ram {  int memory;  String manufact;  Ram(int n, String m) {  this.memory = n;  this.manufact = m;  }  void display() {  System.out.println("Memory Size : " + this.memory);  System.out.println("Memory Manufactures : " + this.manufact);  }  }  void display() {  System.out.println("Price of CPU : " + this.price);  }    public static void main(String[] args) {  Cpu intel = new Cpu(35000);  Cpu.Processor i\_processor = intel.new Processor(4, "intel");  Cpu.Ram i\_ram = new Ram(6, "HP");  intel.display();  i\_processor.display();  i\_ram.display();  // TODO Auto-generated method stub  }  } |

**RESULT:** The program is executed successfully and obtained the output

**OUTPUT**



**COURSE OUTCOME 2**

**PROGRAM NO 1**

**AIM:** Program to Sort strings

**ALGORITHM**

Step 1: Start

Step 2: Take a number of strings from user

Step 3: Perform sorting of strings

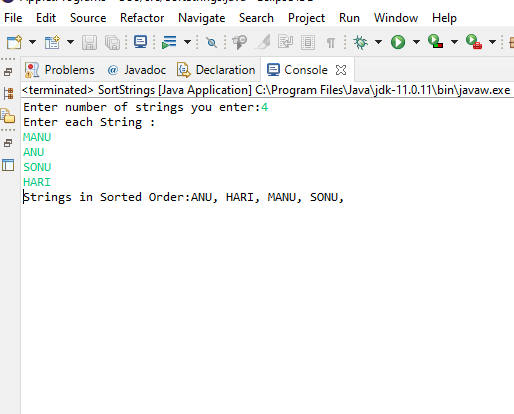
Step 4: Display sorted strings as output

Step 5: Stop

|  |  |
| --- | --- |
| **PROGRAM CODE** | package myproject;  import java.util.Scanner;  public class SortStrings {  public static void main(String[] args) {  int count;  String temp;  Scanner sc = new Scanner(System.in);      System.out.print("Enter number of strings you enter:");  count = sc.nextInt();      String str[] = new String[count];  Scanner sc2 = new Scanner(System.in);      System.out.println("Enter each String :");  for(int i = 0; i < count; i++)  {  str[i] = sc2.nextLine();  }  sc.close();  sc2.close();    //Sorting the strings  for (int i = 0; i < count; i++)  {  for (int j = i + 1; j < count; j++) {  if (str[i].compareTo(str[j])>0)  {  temp = str[i];  str[i] = str[j];  str[j] = temp;  }  }  }      System.out.print("Strings in Sorted Order:");  for (int i = 0; i <= count - 1; i++)  {  System.out.print(str[i] + ", ");  }  }      } |

**RESULT:** The program is executed successfully and obtained the output

**OUTPUT**



**PROGRAM NO 2**

**AIM:** Search an element in an array.

**ALGORITHM**

Step 1: Start

Step 2: Input array of elements and take an element to search from user

Step 3: Search for it in array

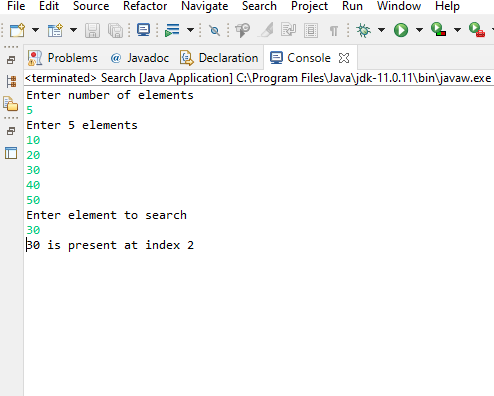
Step 4: If found, display its index and else, display not found message

Step 5: Stop

|  |  |
| --- | --- |
| **PROGRAM CODE** | package myproject;  import java.util.Scanner;  public class Search {  public static void main(String args[]) {  int count, num, i;  int[] inputArray = new int[500];    Scanner in = new Scanner(System.in);    System.out.println("Enter number of elements");  count = in.nextInt();  System.out.println("Enter " + count + " elements");  for(i = 0; i < count; i++) {  inputArray[i] = in.nextInt();  }    System.out.println("Enter element to search");  num = in.nextInt();  // Compare each element of array with num  for (i = 0; i < count ; i++) {  if(num == inputArray[i]){  System.out.println(num+" is present at index "+i);  break;  }  }    if(i == count)  System.out.println(num + " not present in input array");  }  } |

**RESULT:** The program is executed successfully and obtained the output

**OUTPUT**



**PROGRAM NO 3**

**AIM:** Perform string manipulations

**ALGORITHM**

Step 1: Start

Step 2: Take two strings as inputs

Step 3: Perform various string operations on it

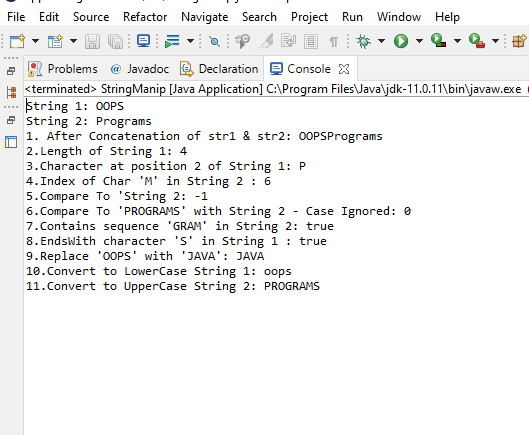
Step 4: Displays each operation’s result as output

Step 5: Stop

|  |  |
| --- | --- |
| **PROGRAM CODE** | package myproject;  public class StringManip {    public static void main(String[] args) {    String str1="OOPS";  String str2="Programs";  System.out.println("String 1: " + str1);  System.out.println("String 2: " +str2);  //String Manipulations  //1.CONCATENATION  String str3=str1.concat(str2);  System.out.println("1. After Concatenation of str1 & str2: "+str3);  //Length of a String  System.out.println("2.Length of String 1: " + str1.length());  //3.CHARACTER AT  System.out.println("3.Character at position 2 of String 1: " + str1.charAt(2));  //4.INDEX OF  System.out.println("4.Index of Char 'M' in String 2 : " + str2.indexOf('m'));  //5.COMPARE TO  System.out.println("5.Compare To 'String 2: " + str1.compareTo(str2));  //6.COMPARE TO IGNORE CASE  System.out.println("6.Compare To 'PROGRAMS' with String 2 - Case Ignored: " + str2.compareToIgnoreCase("Programs"));  //7.CONTAIN  System.out.println("7.Contains sequence 'GRAM' in String 2: " + str2.contains("ram"));  //8.ENDS WITH  System.out.println("8.EndsWith character 'S' in String 1 : " + str1.endsWith("S"));  //9.REPLACE ALL  System.out.println("9.Replace 'OOPS' with 'JAVA': " + str1.replaceAll("OOPS","JAVA"));  //10.TO LOWERCASE  System.out.println("10.Convert to LowerCase String 1: " + str1.toLowerCase());  //11.TO UPPERCASE  System.out.println("11.Convert to UpperCase String 2: " + str2.toUpperCase());  // TODO Auto-generated method stub  }  } |

**RESULT:** The program is executed successfully and obtained the output

**OUTPUT**



**PROGRAM NO 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: Created a class Employee with attributes eNo,eName,eSalary

Step 3: Take n employee information as input

Step 4: And search an employee details using eNo and display it as output

Step 5: Stop

|  |  |
| --- | --- |
| **PROGRAM CODE** | package myproject;  import java.util.Scanner;  public class Employee {  int eNO;  String eName;  int eSalary;  public void GetEmployeeData()  {  Scanner in = new Scanner(System.in);  System.out.print("Enter Employee id:");  eNO=in.nextInt();  System.out.print("Enter name of Employee:");  eName=in.next();  System.out.print("Enter salary of Employee:");  eSalary=in.nextInt();  System.out.println("\n");  }  void display() {  System.out.println("Employee id = " + eNO);  System.out.println("Employee name = " + eName);  System.out.println("Employee salary = " + eSalary);  System.out.println("\n");  }  public static void main(String[] args) {  int num;  Scanner sc= new Scanner(System.in);  System.out.print("Enter No of employees :");  num=sc.nextInt();  Employee e[]= new Employee[num];  for( int i=0;i<num;i++) {  e[i]= new Employee();  e[i].GetEmployeeData();  }  System.out.println("\*\*\*\*Details of Employees\*\*\*\*");  for(int i =0;i<num;i++) {  e[i].display();  }  System.out.print("Enter the Employee ID to search :");  int id = sc.nextInt();  int i;  for(i =0;i<num;i++)  {  if(id == e[i].eNO)  {  e[i].display();  }  }  if(i == 0)  {  System.out.println("\nEmployee Details are not available, Please enter a valid ID!!");  }  sc.close();  // TODO Auto-generated method stub  }  } |

**RESULT:** The program is executed successfully and obtained the output

**OUTPUT**

