**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: Define a class having name Product and members as pcode,pname and price.

Step 3: Declare three objects in the class and add the values of each data members into objects.

Step 4: Using if condition check which object has the lowest price and print it.

Step 5: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Product.java | 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 2 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();  }  }  } |

**OUTPUT:**

Graphical user interface, text, application, chat or text message

Description automatically generated

**PROGRAM NO : 2**

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

**ALGORITHM :**

Step 1: Start.

Step 2: Define a class having name matrixadd

Step 3: Read row number(m),column number (n) and initialize the  double dimensional arrays mat1[][],mat2[][],res[][] with same row number ,column number.

Step 4: Store the first matrix elements into the two-dimensional array matrix mat1[][] using two for loops. i indicates row number, j indicates column index. Similarly second matrix elements in to mat2[][].

Step 5: Add the two matrices using for loop.

for i=0 to i<m

for j=0 to j<n

mat1[i][j] + mat2[i][j] and store it in to the matrix res[i][j] .

Step 6: Print sum of matrices res[i][j].

Stop 7: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| matrix.java | package myproject;  import java.util.Scanner;      public class matrixadd {          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 a[][] = new int[row][col];          int b[][] = new int[row][col];          int tot[][] = new int[row][col];          System.out.println("Enter the elements of matrix 1:2");          for ( i= 0 ; i < row ; i++ ) {          for ( j= 0 ; j < col ;j++ )          a[i][j] = in.nextInt();          System.out.println();     }          System.out.println("Enter the elements of matrix 2:");          for ( i= 0 ; i < row ; i++ ) {          for ( j= 0 ; j < col ;j++ )          b[i][j] = in.nextInt();          System.out.println();   }          for ( i= 0 ; i < row ; i++ )          for ( j= 0 ; j < col ;j++ )          tot[i][j] = a[i][j] + b[i][j] ;          System.out.println("Sum of matrices:-");          for ( i= 0 ; i < row ; i++ )     {          for ( j= 0 ; j < col ;j++ )          System.out.print(tot[i][j]+"\t");          System.out.println();    }          }  } |

**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

**PROGRAM NO : 3**

**AIM:** Add complex numbers.

**ALGORITHM:**

Step 1: Start.

Step 2: Define a class having name complex and data members are real and imaginary number.

Step 3: Define a function complex and add values to variables.

Step 4 : Define a function complex sum to add complex number using 3rd ComplexNumber object and return the value.

Step 5: Print the sum value.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Add-complex.java | package java\_lab;  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(3.5, 6);  complex c2 = new complex(9.2, 2.5);  complex temp = sum(c1, c2);  System.out.printf("Sum is: "+ temp.real+" + "+ temp.img +"i");  }  } |

**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

**PROGRAM NO : 4**

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

**ALGORITHM:**

Step 1: Start.

Step 2 : Read row number,column number and initialize the  double dimensional array with same row number ,column number.

Step 3 : Store the first matrix elements into the two-dimensional array matrix using two for loops. i indicates row number, j indicates column index.

Step 4: Check whether the matrix is symmetric or not.

Step 5: Print the symmetric matrix or if not.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Read-matrix-console.java | package java\_lab;  import java.util.Scanner;  public class matrix {  public static void main(String[] args)  {  Scanner sc = new Scanner(System.in);  System.out.println("Enter the no. of rows : ");  int rows = sc.nextInt();  System.out.println("Enter the no. of columns : ");  int cols = sc.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] = sc.nextInt();  }  }  System.out.println("Printing the input matrix :");  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 symmetric = true;  for (int i = 0; i < rows; i++)  {  for (int j = 0; j < cols; j++)  {  if(matrix[i][j] != matrix[j][i])  {  symmetric = false;  break;  }  }  }  if(symmetric)  {  System.out.println("The given matrix is symmetric...");  }  else  {  System.out.println("The given matrix is not symmetric...");  }  }  sc.close();  }  } |

**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

**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: Define a class CPU with data member price and class processor.

Step 3: Class processor contain data members ncores, manufacturer and a nested class RAM.

Step 4: class RAM contain memory and manufacturer as data members.

Step 5: Create objects in corresponding classes and display its details.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Nested-Class.java | package JAVA;  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);  }  } |

**OUTPUT:**

Text

Description automatically generated with medium confidence

**PROGRAM NO : 6**

**AIM:** Program to Sort strings.

**ALGORITHM :**

Step 1: Start

Step 2: Select the first element of the list (i.e., Element at first position in the list).

Step 3: Compare the selected element with all the other elements in the list.

Step 4: In every comparison, if any element is found smaller than the selected element (for Ascending order), then both are swapped.

Step 5: Repeat the same procedure with element in the next position in the list till the entire list is sorted.

Step 6: Stop

.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Sort-strings.java | package java\_lab;  import java.util.Arrays;  import java.util.Scanner;  public class Sort {  public static void main(String args[]) {  Scanner sc = new Scanner(System.in);  System.out.println("Enter a string : ");  String str = sc.nextLine();  char charArray[] = str.toCharArray();  Arrays.sort(charArray);  System.out.println(new String(charArray));  }  } |

**OUTPUT:**

**Graphical user interface

Description automatically generated**

**PROGRAM NO : 7**

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

**ALGORITHM :**

Step 1: Start

Step 2: Check each element in the given list with the string provided by the user.

Step 3: If string is found, display the position of the string found, else display string not found.

Step : Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Element-Search.java | package java\_lab;  import java.util.Scanner;  public class SearchElement {  public static void main(String[] args)  {  int num, s, flag = 0, i = 0;  Scanner sc = new Scanner(System.in);  System.out.print("Enter no. of elements in Array:");  num = sc.nextInt();  int arr[] = new int[num];  System.out.println("Enter the elements in to the Array:");  for(i = 0; i < num; i++)  {  arr[i] = sc.nextInt();  }  System.out.print("Enter the element you want to Search:");  s = sc.nextInt();  for(i = 0; i < num; i++)  {  if(arr[i] == s)  {  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");  }  } |

**OUTPUT:**

Graphical user interface, text, application

Description automatically generated

**PROGRAM NO : 8**

**AIM:** Perform string manipulations

**ALGORITHM :**

Step 1: Start

Step 2: Character at position 2 of string1

Step 3: Take the strings provided by the user and concatenate them.

Step 3: Display length of string2

Step 4: Display Unicode of first character in string1

Step 5: Compare string1 and string2

Step 6: Display String2 in uppercase

Step 7: Display String1 in lowercase

Step 8: Check whether string1 is empty?

Step 9: Display String1 after replacing “H” with ”$”

Step 10: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Str-Manipulation.java | package str;  import java.util.Scanner;  public class str\_manipulation {  public static void main(String[] args) {  Scanner sc=new Scanner(System.in);  String str1,str2;  System.out.println("Enter string1 : ");  str1=sc.nextLine();  System.out.println("Enter string2 : ");  str2=sc.nextLine();  System.out.println("Character at position 2 of string1 :"+str1.charAt(2));  System.out.println("Concating string1 and string2 : "+str1.concat(str2));  System.out.println("Length of string2: "+str2.length());  System.out.println("Unicode of first character in string1: "+str1.codePointAt(0));  System.out.println("Compare string1 and string2: "+str2.compareTo(str1));  System.out.println("String2 in uppercase: "+str2.toUpperCase());  System.out.println("String1 in lowercase: "+str1.toLowerCase());  System.out.println("Is string1 empty?: "+str1.isEmpty());  System.out.println("String1 after replacing : "+str1.replace('H', '$'));  }  } |

**OUTPUT:**

Text, letter

Description automatically generated

**PROGRAM NO : 9**

**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: Store the information of employee name, number and salary in eNo, eName and salary respectively

Step 3: Search the ‘eNo’ attribute of the list of Employee Objects for the ‘eNo’ provided by the user.

Step 4: If user provided ‘eNo’ is found inside the Employee object list, display the details of the corresponding employee.

Step 5: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| employee.java | package employee;  import java.util.Scanner;  public class employee {  int eNo;  String eName;  double eSalary;  void getinfo()  {  Scanner sc=new Scanner(System.in);  System.out.println("Enter employee number: ");  eNo=sc.nextInt();  Scanner sc1=new Scanner(System.in);  System.out.println("Enter employee name: ");  eName=sc1.nextLine();  Scanner sc2=new Scanner(System.in);  System.out.println("Enter employee salary: ");  eSalary=sc2.nextDouble();  }  void display()  {  System.out.println("Employee no: "+eNo);  System.out.println("Employee name: "+eName);  System.out.println("Salary: "+eSalary);  }  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].eNo)  {  e[i].display();  flag=1;  break;  }  }  if(flag==0)  {  System.out.println("No such employee");  }  }  } |

**OUTPUT:**

