**PROGRAM 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.

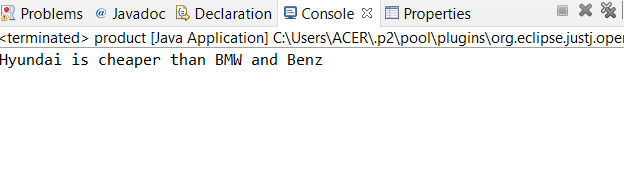
Step 5: Stop.

**PROGRAM CODE**

|  |  |
| --- | --- |
| product.java | **public** **class** product {  **int** pcode;  String pname;  **int** price;    **public** **static** **void** main(String[] args) {  product obj1=**new** product();  obj1.pcode=1001;  obj1.pname="BMW";  obj1.price=2500000;  product obj2=**new** product();  obj2.pcode=2012;  obj2.pname="Benz";  obj2.price=2000000;  product obj3=**new** product();  obj3.pcode=3211;  obj3.pname="Hyundai";  obj3.price=1500000;  **if**(obj1.price<=obj2.price && obj1.price<=obj3.price)  System.***out***.println(obj1.pname+" is cheaper than "+obj2.pname+" and "+obj3.pname);  **else** **if**(obj2.price<=obj1.price && obj2.price<=obj3.price)  System.***out***.println(obj2.pname+" is cheaper than "+obj1.pname+" and "+obj3.pname);  **else**  System.***out***.println(obj3.pname+" is cheaper than "+obj1.pname+" and "+obj2.pname);  }  } |

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

**OUTPUT**

****

**PROGRAM 2**

**AIM**

Read 2 matrices from the console and perform matrix addition.

**ALGORITHM**

Step 1: Start.

Step 2: Define a class having name AddMatrix.

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].

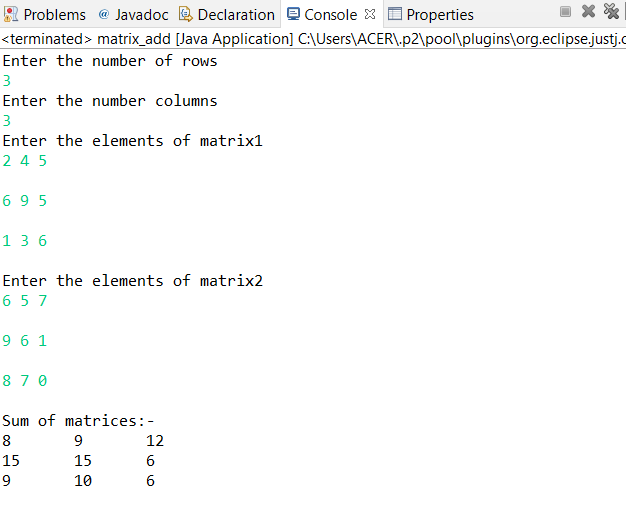
Stop 7: Stop.

**PROGRAM CODE**

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

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

**OUTPUT**

****

**PROGRAM 3**

**AIM**

Add complex numbers.

**ALGORITHM**

Step 1: Start.

Step 2: Define a class having name ComplexNumber and data members are real and

imaginary number.

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

Step 4: Define a function ComplexNumber sum to add complex number using 3rd

ComplexNumber object and return the value.

Step 5: Print the sum value.

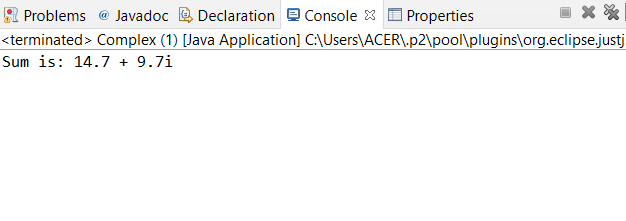
Step 6: Stop.

**PROGRAM CODE**

|  |  |
| --- | --- |
| Complex.java | **public** **class** Complex {  **double** real;  **double** 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(8.2, 6);  Complex c2 = **new** Complex(6.5, 3.7);  Complex temp = *sum*(c1, c2);  System.***out***.printf("Sum is: "+ temp.real+" + "+ temp.img +"i");  // **TODO** Auto-generated method stub  }  } |

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

**OUTPUT**

****

**PROGRAM 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.

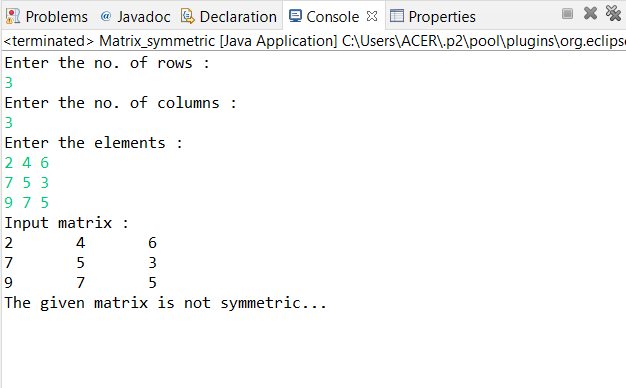
Step 6: Stop.

**PROGRAM CODE**

|  |  |
| --- | --- |
| Matrix\_symmetric.java | **import** java.util.Scanner;  **public** **class** Matrix\_symmetric {  **public** **static** **void** main(String[] args) {  Scanner mat = **new** Scanner(System.***in***);    System.***out***.println("Enter the no. of rows : ");    **int** rows = mat.nextInt();    System.***out***.println("Enter the no. of columns : ");    **int** cols = mat.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] = mat.nextInt();  }  }    System.***out***.println("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("Matrix is not a square matrix, It is not 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("Entered matrix is symmetric...");  }  **else**  {  System.***out***.println("The given matrix is not symmetric...");  }  }    mat.close();    // **TODO** Auto-generated method stub  }  } |

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

**OUTPUT**

****

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

Step 3: Class processor contain data members no\_cores,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 it’s details.

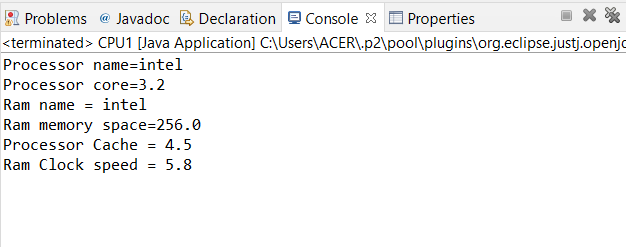
Step 6: Stop.

**PROGRAM CODE**

|  |  |
| --- | --- |
| CPU1.java | **package** CPU;  **class** CPU {  **double** price;  **class** Processor{  **double** cores=3.2;  String manufacturer="intel";  **double** getCache(){  **return** 4.5;  }  }  **protected** **class** RAM{    **double** memory=256;  String manufacturer="intel";  **double** getClockSpeed(){  **return** 5.8;  }  }  }  **public** **class** CPU1 {  **public** **static** **void** main(String[] args) {  CPU cpu = **new** CPU();  CPU.Processor processor = cpu.**new** Processor();  CPU.RAM ram = cpu.**new** RAM();    System.***out***.println("Processor name="+processor.manufacturer);  System.***out***.println("Processor core="+processor.cores);  System.***out***.println("Ram name = " + ram.manufacturer);  System.***out***.println("Ram memory space="+ram.memory);  System.***out***.println("Processor Cache = " + processor.getCache());  System.***out***.println("Ram Clock speed = " + ram.getClockSpeed());  // **TODO** Auto-generated method stub  }  } |

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

**OUTPUT**

****

**PROGRAM 6**

**AIM**

Program to Sort strings.

**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.

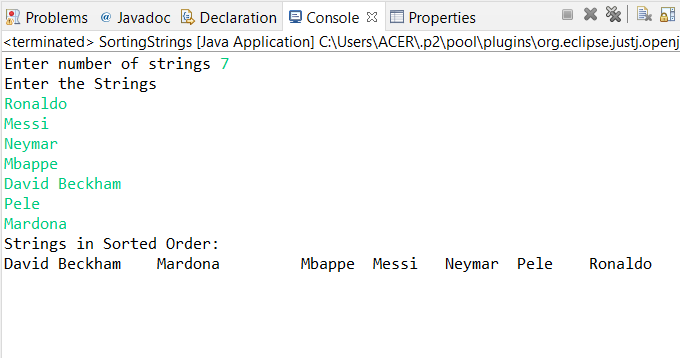
Step 4: Stop

**PROGRAM CODE**

|  |  |
| --- | --- |
| SortingStrings.java | **import** java.util.Scanner;  **public** **class** SortingStrings {  **public** **static** **void** main(String[] args) {  **int** n;  String temp;  Scanner sc = **new** Scanner(System.***in***);  System.***out***.print("Enter number of strings ");  n=sc.nextInt();      String str[] = **new** String[n];  Scanner sc1 = **new** Scanner(System.***in***);    System.***out***.println("Enter the Strings");  **for**(**int** i=0;i<n;i++)  {  str[i]= sc1.nextLine();  }  sc.close();  sc1.close();    **for**(**int** i=0;i<n; i++)  {  **for** (**int** j=i+1;j<n; 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:\n");  **for**(**int** i=0;i<n;i++)  {  System.***out***.print(str[i]+ "\t ");  }  }  } |

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

**OUTPUT**



**PROGRAM 7**

**AIM**

Search an element in an array.

**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.

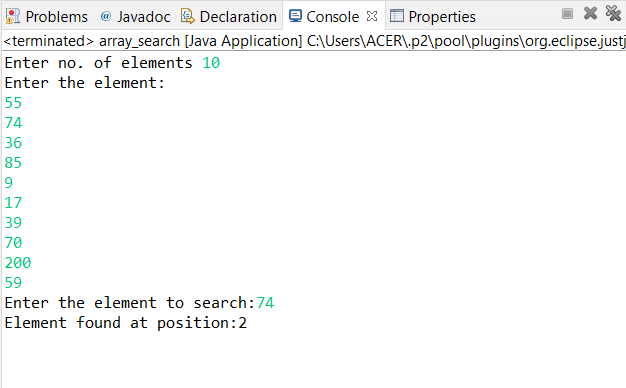
Step 6: Stop

**PROGRAM CODE**

|  |  |
| --- | --- |
| Array\_search.java | **import** java.util.Scanner;  **public** **class** array\_search {  **public** **static** **void** main(String[] args) {  **int** n, x, flag = 0, i = 0;  Scanner sc = **new** Scanner(System.***in***);  System.***out***.print("Enter no. of elements ");  n = sc.nextInt();  **int** a[] = **new** **int**[n];  System.***out***.println("Enter the element:");  **for**(i = 0; i < n; i++)  {  a[i] = sc.nextInt();  }  System.***out***.print("Enter the element to search:");  x = sc.nextInt();  sc.close();  **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");  }  }  // **TODO** Auto-generated method stub  } |

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

**OUTPUT**



**PROGRAM 8**

**AIM**

Perform string manipulations

**ALGORITHM**

Step 1: Start

Step 2: Take the strings provided by the user and concatenate them.

Step 3: Display the combined string with lower case.

Step 3: Display the combined string with upper case.

Step 4: Display the combined string after replacing all the ‘s’ & ‘S’ characters with ‘$’

character.

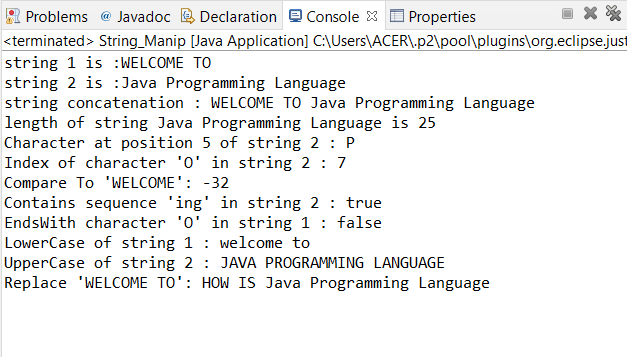
Step 5: Stop

**PROGRAM CODE**

|  |  |
| --- | --- |
| String\_Manip.java | **public** **class** String\_Manip {  **public** **static** **void** main(String[] args) {  String st1="WELCOME TO";  System.***out***.println("string 1 is :"+ st1);  String st2="Java Programming Language";  System.***out***.println("string 2 is :"+ st2);  String st=st1.concat(st2);  System.***out***.println("string concatenation : " + st);  System.***out***.println("length of string "+ st2 +" is " +st2.length());  System.***out***.println("Character at position 5 of string 2 : " + st2.charAt(5));  System.***out***.println("Index of character 'O' in string 2 : " + st2.indexOf('o'));  System.***out***.println("Compare To 'WELCOME': " + st1.compareTo("Welecome"));  System.***out***.println("Contains sequence 'ing' in string 2 : " + st2.contains("ing"));  System.***out***.println("EndsWith character 'O' in string 1 : " + st1.endsWith("O"));  System.***out***.println("LowerCase of string 1 : " + st1.toLowerCase());  System.***out***.println("UpperCase of string 2 : " + st2.toUpperCase());  System.***out***.println("Replace 'WELCOME TO': " + st.replace("WELCOME TO", "HOW IS"));  // **TODO** Auto-generated method stub  }  } |

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

**OUTPUT**



**PROGRAM 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: Search the ‘eNo’ attribute of the list of Employee Objects for the ‘eNo’ provided by

the user.

Step 3: If user provided ‘eNo’ is found inside the Employee object list, display the details of

the corresponding employee.

Step 4: Stop

**PROGRAM CODE**

|  |  |
| --- | --- |
| Employee.java | **import** java.util.Scanner;  **public** **class** Employee {  **int** eNo;  String eName;  **double** eSalary;  **void** getdata()  {  Scanner sc=**new** Scanner(System.***in***);    System.***out***.println("Enter Employee Id ");  eNo=sc.nextInt();  System.***out***.println("Enter Employee Name");  eName=sc.next();  System.***out***.println("Enter Employee Salary");  eSalary=sc.nextDouble();  }  **void** display()  {  System.***out***.println("Employee id is : "+ eNo);  System.***out***.println("Employee name is : "+ eName);  System.***out***.println("Employee salary is : "+ eSalary);  }  **public** **static** **void** main(String[] args) {  Scanner sc1=**new** Scanner(System.***in***);  **int** i,n,c,f=0;  System.***out***.println("Enter the number of Employees");  n=sc1.nextInt();  Employee e[]=**new** Employee[n];  **for**(i=0;i<n;i++) {  e[i]=**new** Employee();  e[i].getdata();  }  System.***out***.println("Employee details are :\n");  **for**(i=0;i<n;i++) {  e[i].display();  }  System.***out***.println("\nEnter id of Employee to search: ");  c=sc1.nextInt();  **for**(i=0;i<n;i++) {  **if**(c==e[i].eNo) {  f=1;  **break**;  }  }  **if**(f==1) {  System.***out***.println("Details of employee is ");  e[i].display();  }  **else**  System.***out***.println("Employee Id is Invalid");  }    } |

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

**OUTPUT**

