**COURSE OUTCOME 1**

**PROBLEM: 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.

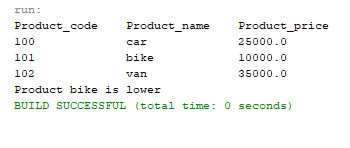
Step 4: Using if condition check which object has the lowest price and print it.

Step 5: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| product.java | public class product {  int pcode;  String pname;  double price;    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);  }  void small(double a, double b, double c)  {  if(a<b && a<c)  {  System.out.println("Product car is lower");  }  else if(b<c && b<a)  {  System.out.println("Product bike is lower");  }  else  {  System.out.println("Product van is lower");  }  }  public static void main(String[] args) {  product obj1 = new product();  product obj2 = new product();  product obj3 = new product();  obj1.data(100,"car",25000);  obj2.data(101,"bike",10000);  obj3.data(102,"van",35000);  System.out.println("Product\_code\tProduct\_name\tProduct\_price");  obj1.display();  obj2.display();  obj3.display();  obj1.small(obj1.price,obj2.price,obj3.price);    }  } |

**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 matrix\_add.

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] .

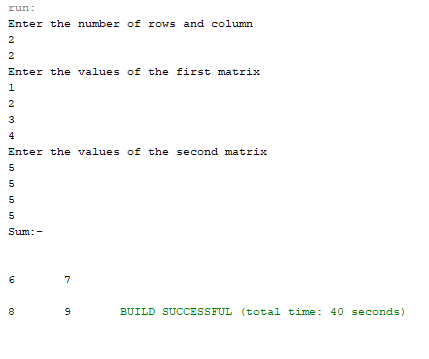
Step 6: Print sum of matrices res[i][j].

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;  int mat1[][]= new int[3][3];  int mat2[][]= new int[3][3];  int sum[][] = new int[3][3];  System.out.println("Enter the number of rows and column");  Scanner sc = new Scanner(System.in);  m = sc.nextInt();  n = sc.nextInt();  System.out.println("Enter the values of the first matrix");  for(int i=0;i<m;i++)  {  for(int j=0;j<n;j++)  {  mat1[i][j] = sc.nextInt();  }  }    System.out.println("Enter the values of the second matrix");  for(int i=0;i<m;i++)  {  for(int j=0;j<n;j++)  {  mat2[i][j] = sc.nextInt();  }  }    for(int i=0;i<m;i++)  {  for(int j=0;j<n;j++)  {  sum[i][j] = mat1[i][j]+mat2[i][j];  }  }  System.out.println("Sum:-");  for(int i=0;i<m;i++)  {  System.out.println("\n");  for(int j=0;j<n;j++)  {  System.out.print(sum[i][j]+"\t");    }  }  }  } |

**OUTPUT:**

****

**PROGRAM: 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 complex object and return the value.

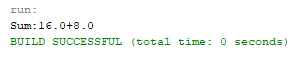
Step 5: Print the sum value.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| complex. java | public class complex {  double real,img;  complex(double a,double b)  {  real = a;  img = b;  }  public static complex sum(complex c1,complex c2)  {  complex t = new complex(0,0);  t.real = c1.real + c2.real;  t.img = c1.img + c2.img;  return t;  }  public static void main(String[] args) {  complex c1 = new complex(12,3);  complex c2 = new complex(4,5);  complex add = sum(c1,c2);  System.out.println("Sum:"+add.real+"+"+add.img);  }  } |

**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.

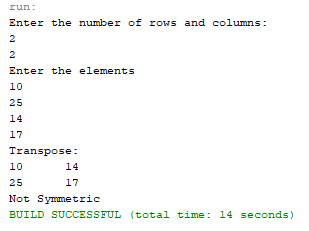
Step 5: Print the symmetric matrix or if not.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| symmetric.java | import java.util.Scanner;  public class symmetric {  public static void main(String[] args) {  int r,c;  int flag=0;  Scanner sc = new Scanner(System.in);  System.out.println("Enter the number of rows and columns:");  r = sc.nextInt();  c = sc.nextInt();  int a[][] = new int[r][c];  int b[][] = new int[r][c];  System.out.println("Enter the elements");  for(int i=0;i<r;i++)  {  for(int j=0;j<c;j++)  {  a[i][j] = sc.nextInt();  }  }  if(r==c)  {  for(int i=0;i<r;i++)  {  for(int j=0;j<c;j++)  {  b[i][j] = a[j][i];  }  }  }  System.out.println("Transpose:");  for(int i=0;i<r;i++)  {  for(int j=0;j<c;j++)  {  System.out.print(b[i][j]+"\t");  }  System.out.println("");  }  for(int i=0;i<r;i++)  {  for(int j=0;j<c;j++)  {  if(a[i][j]!=b[i][j])  {  flag = 1;  break;  }  }  }  if(flag==0)  {  System.out.println("Symmetric");  }  else  {  System.out.println("Not Symmetric");  }  }  } |

**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.

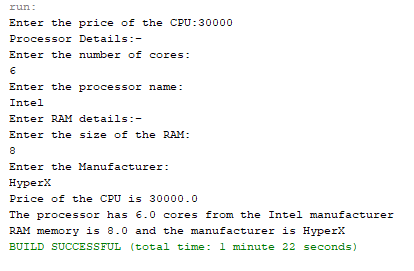
Step 5: Create objects in corresponding classes and display it’s details.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CPU.java | import java.util.Scanner;  public class CPU {  double price;  public class processor  {  float cores;  String mfg;  void display(float n, String pname){  cores = n;  mfg = pname;  System.out.println("The processor has "+cores+" cores from the "+mfg+" manufacturer");  }  }  static class RAM{  float mem;  String mfg;  void display(float n, String pname){  mem = n;  mfg = pname;  System.out.println("RAM memory is "+mem+" and the manufacturer is "+mfg);  }  }  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.print("Enter the price of the CPU:");  obj.price = sc.nextDouble();  System.out.println("Processor Details:-");  System.out.println("Enter the number of cores:");  float c = sc.nextFloat();  sc.nextLine();  System.out.println("Enter the processor name:");  String pname = sc.nextLine();    System.out.println("Enter RAM details:-");  System.out.println("Enter the size of the RAM:");  float mem = sc.nextFloat();  sc.nextLine();  System.out.println("Enter the Manufacturer:");  String mfg = sc.nextLine();    System.out.println("Price of the CPU is "+obj.price);  obj1.display(c, pname);  obj2.display(mem, mfg);  }} |

**OUTPUT:**

****

**COURSE OUTCOME 2**

**PROGRAM: 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 comparision, if any element is found smaller than the selected element (for Ascending order), then both are swapped.

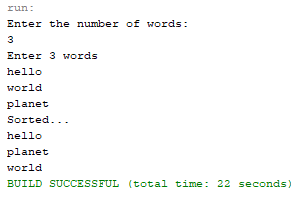
Step 5: Repeat the same procedure with element in the next position in the list till the entire list is sorted.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO2Q1.java | //Program to sort strings  import java.util.Scanner;  public class CO2Q1  {  public class sort  {  public void display(String s[],int n)  {  String temp;  String str[] = s;  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.println("Sorted...");  for(int i=0;i<n;i++)  System.out.println(str[i]);  }  }  public static void main(String[] args)  {  int count;  CO2Q1 obj = new CO2Q1();  CO2Q1.sort ob = obj.new sort();  Scanner sc = new Scanner(System.in);  System.out.println("Enter the number of words:");  count = sc.nextInt();  String str[] = new String[count];  System.out.println("Enter "+count+" words");    sc.nextLine();    for(int i=0;i<count;i++)  {  str[i] = sc.nextLine();  }    ob.display(str,count);  }  } |

**OUTPUT:**



**PROGRAM: 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.

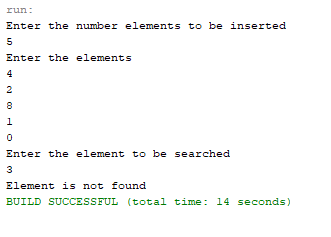
Step 3: If string is found, display the position of the string found, else display string not found.

Step : Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO2Q2.java | import java.util.Scanner;  public class CO2Q2  {  public static void main(String[] args)  {  int count,flag=0;  int pos = 0;  String t;  System.out.println("Enter the number elements to be inserted");  Scanner sc = new Scanner(System.in);  count = sc.nextInt();  String str[] = new String[count];  sc.nextLine();  System.out.println("Enter the elements");  for(int i=0;i<count;i++)  str[i] = sc.nextLine();  System.out.println("Enter the element to be searched");  t = sc.nextLine();  for(int i=0;i<count;i++)  {  if(str[i].equals(t))  {  flag = 1;  pos = i;  break;  }  }  if(flag==1)  {  System.out.println("Element is in position "+(pos+1));  }  else  {  System.out.println("Element is not found");  }  }  } |

**OUTPUT:**



**PROGRAM: 8**

**AIM:** Perform string manipulations

**ALGORITHM :**

Step 1: Start

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

Step 4: Display the length of the first string.

Step 5: Display the comparison of two strings.

Step 6: Display the strings were they are empty or not.

Step 7: Display the trimmed string form.

Step 8: Display the string with lower case.

Step 9: Display the string with upper case.

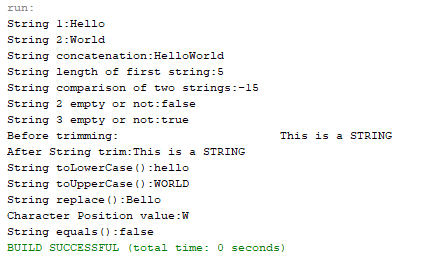
Step 10: Display the string after replacing all the ‘H’ characters with ‘B’ character.

Step 11: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO2Q3.java | //String Manipulation  public class CO2Q3  {  public static void main(String[] args)  {  String s1 = "Hello";  String s2 = "World";  String s3 = "";  String s4 = " This is a STRING ";  System.out.println("String 1:"+s1);  System.out.println("String 2:"+s2);  System.out.println("String concatenation:"+s1.concat(s2));  System.out.println("String length of first string:"+s1.length());  System.out.println("String comparison of two strings:"+s1.compareTo(s2));  System.out.println("String 2 empty or not:"+s2.isEmpty());  System.out.println("String 3 empty or not:"+s3.isEmpty());  System.out.println("Before trimming:"+s4);  System.out.println("After String trim:"+s4.trim());  System.out.println("String toLowerCase():"+s1.toLowerCase());  System.out.println("String toUpperCase():"+s2.toUpperCase());  System.out.println("String replace():"+(s1.replace("H", "B")));  System.out.println("Character Position value:"+s2.charAt(0));  System.out.println("String equals():"+s1.equals(s2));    }  } |

**OUTPUT:**



**PROGRAM: 9**

**AIM:** Program to create a class for Employee having attributes Empid, Name, Salary, Address. Read n employee information and Search for an employee given Empid, using the concept of Array of Objects.

**ALGORITHM :**

Step 1: Start

Step 2: Search the ‘Empid’ attribute of the list of Employee Objects for the ‘Empid’ provided by the user.

Step 3: If user provided ‘Empid’ is found inside the Employee object list, display the details of the corresponding employee.

Step 4: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CO2Q4.java | //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.  import java.util.\*;  public class CO2Q4{    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){  CO2Q4 obj = new CO2Q4();  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("\nTo check a specific record");  System.out.println("Enter the Emp\_No:");  int e = sc.nextInt();  obj.displayinfo(e,obj.count);  sc.close();  }  } |

**OUTPUT:**

