**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.

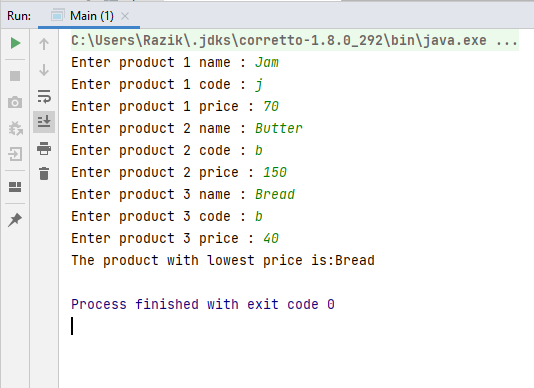
Step 4: Using if condition check which object has the lowest price and print it.

Step 5: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args) {  //Driver program  Scanner scan = new Scanner(System.in);  Product[] products = new Product[3];  for(int i=0;i<3;i++)  {  int count = i+1;  products[i] = new Product();  System.out.print("Enter product "+count+" name : ");  products[i].pname = scan.nextLine();  System.out.print("Enter product "+count+" code : ");  products[i].pcode = scan.nextLine();  System.out.print("Enter product "+count+" price : ");  products[i].price = Integer.parseInt(scan.nextLine());  }  scan.close();  if(products[0].price< products[1].price && products[0].price< products[2].price) {  System.out.println("The product with lowest price is:"+ products[0].pname);  }  if(products[1].price<products[0].price && products[1].price<products[2].price) {  System.out.println("The product with lowest price is:"+products[1].pname);  }  if(products[2].price<products[0].price && products[2].price<products[1].price) {  System.out.println("The product with lowest price is:"+products[2].pname);  }  }  }  package com.lab\_cycles.co1.q1;  import java.util.Scanner;  public class Main {  public static void main(String[] args) {  //Driver program  Scanner scan = new Scanner(System.in);  Product[] products = new Product[3];  for(int i=0;i<3;i++)  {  int count = i+1;  products[i] = new Product();  System.out.print("Enter product "+count+" name : ");  products[i].pname = scan.nextLine();  System.out.print("Enter product "+count+" code : ");  products[i].pcode = scan.nextLine();  System.out.print("Enter product "+count+" price : ");  products[i].price = Integer.parseInt(scan.nextLine());  }  scan.close();  if(products[0].price< products[1].price && products[0].price< products[2].price) {  System.out.println("The product with lowest price is:"+ products[0].pname);  }  if(products[1].price<products[0].price && products[1].price<products[2].price) {  System.out.println("The product with lowest price is:"+products[1].pname);  }  if(products[2].price<products[0].price && products[2].price<products[1].price) {  System.out.println("The product with lowest price is:"+products[2].pname);  }  }  } |
| Product.java | public class Product {  public String pcode;  public String pname;  public int price;  } |

**OUTPUT:**

****

**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 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] .

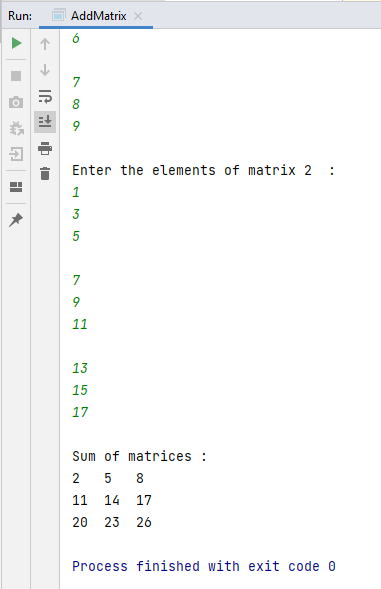
Step 6: Print sum of matrices res[i][j].

Stop 7: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| AddMatrix  .java | import java.util.Scanner;  public 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 matrix 1 : ");  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 matrix 2 : ");  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();  }  }  } |

**OUTPUT:**

****

**PROGRAM NO : 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.

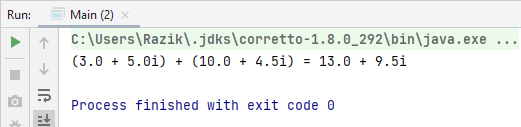
Step 5: Print the sum value.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| ComplexNumber. java | public class ComplexNumber {  double real,imaginary;  ComplexNumber(double real,double imaginary)  {  this.real = real;  this.imaginary = imaginary;  }  ComplexNumber addComplexNumber(ComplexNumber complexNumber1,ComplexNumber complexNumber2)  {  double real = complexNumber1.real+ complexNumber2.real;  double imaginary = complexNumber1.imaginary+complexNumber2.imaginary;  ComplexNumber complexNumber = new ComplexNumber(real,imaginary);  return complexNumber;  }  String displayComplexNumber()  {  return this.real+" + "+this.imaginary+"i";  }  } |
| Main.java | public class Main {  public static void main(String[] args) {  ComplexNumber complexNumber1 = new ComplexNumber(3,5);  ComplexNumber complexNumber2 = new ComplexNumber(10,4.5);  ComplexNumber complexNumber3 = new ComplexNumber(0,0);  complexNumber3 = complexNumber3.addComplexNumber(complexNumber1,complexNumber2);  System.out.println("("+complexNumber1.displayComplexNumber()+") + ("+complexNumber2.displayComplexNumber()+") = "+complexNumber3.displayComplexNumber());  }  } |

**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 : 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.

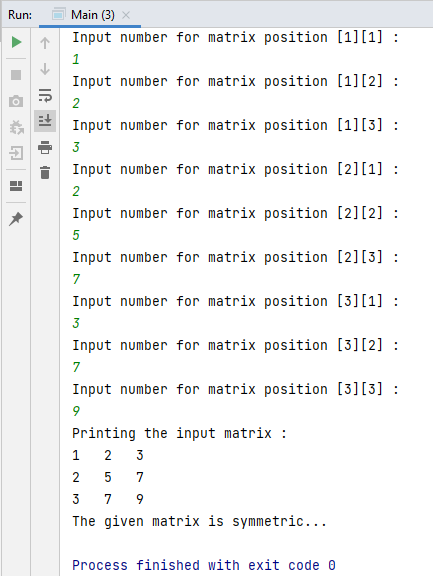
Step 5: Print the symmetric matrix or if not.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.Scanner;  public class Main  {  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++)  {  System.out.println("Input number for matrix position ["+(i+1)+"]["+(j+1)+"] : ");  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();  }  //Checking the input matrix for symmetric  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:**



**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 no\_cores,manufacturer and a nested class RAM.

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

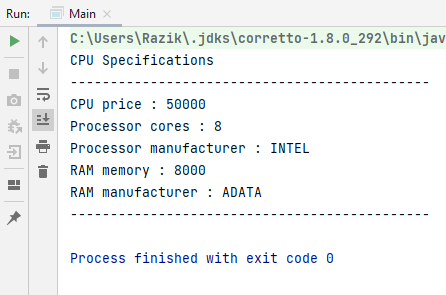
Step 5: Create objects in corresponding classes and display it’s details.

Step 6: Stop.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| CPU.java | public class CPU {  int price;  Processor processor = new Processor();  RAM ram = new RAM();  static class Processor  {  int cores;  String manufacturer;  }  static class RAM  {  int memory;  String manufacturer;  }  CPU(int cpuPrice,int cores,String cpuManufacturer,int memory,String ramManufacturer){  this.price = cpuPrice;  this.processor.cores = cores;  this.processor.manufacturer = cpuManufacturer;  this.ram.memory = memory;  this.ram.manufacturer = ramManufacturer;  }  void printCPUSpecifications()  {  System.out.println("CPU Specifications");  System.out.println("---------------------------------------------");  System.out.println("CPU price : "+this.price);  System.out.println("Processor cores : "+this.processor.cores);  System.out.println("Processor manufacturer : "+this.processor.manufacturer);  System.out.println("RAM memory : "+this.ram.memory);  System.out.println("RAM manufacturer : "+this.ram.manufacturer);  System.out.println("---------------------------------------------");  }  } |
| Main.java | public class Main {  public static void main(String[] args) {  CPU cpu1 = new CPU(50000, 8, "INTEL", 8000, "ADATA");  cpu1.printCPUSpecifications();  }  } |

**OUTPUT:**

****

**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 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.

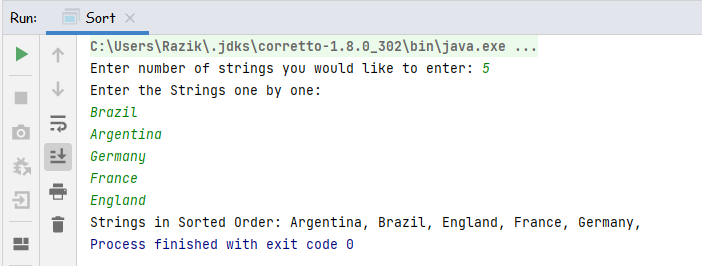
Step 6: Stop

.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Sort.java | import java.util.Scanner;  public class Sort {  public static void main(String[] args) {  int count;  String temp;  Scanner scan = new Scanner(System.in);  //User will be asked to enter the count of strings  System.out.print("Enter number of strings you would like to enter: ");  count = scan.nextInt();  String[] str = new String[count];  Scanner scan2 = new Scanner(System.in);  //User is entering the strings and they are stored in an array  System.out.println("Enter the Strings one by one: ");  for(int i = 0; i < count; i++)  {  str[i] = scan2.nextLine();  }  scan.close();  scan2.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;  }  }  }  //Displaying the strings after sorting them based on alphabetical order  System.out.print("Strings in Sorted Order: ");  for (int i = 0; i <= count - 1; i++)  {  System.out.print(str[i] + ", ");  }  }  } |

**OUTPUT:**



**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.

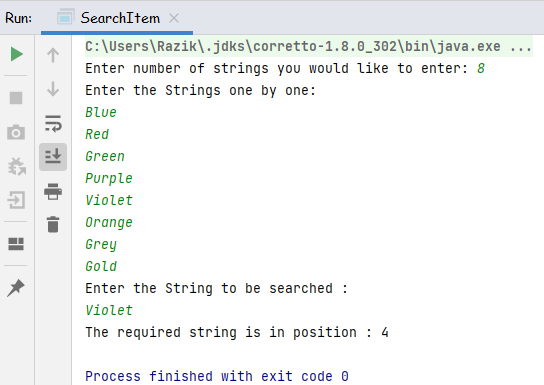
Step 3: If string is found, display the position of the string found, else display string not found.

Step : Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| SearchItem.java | import java.util.Scanner;  public class SearchItem {  public static void main(String[] args) {  int count;  String temp;  Scanner scan = new Scanner(System.in);  Scanner scan2 = new Scanner(System.in);  Scanner scan3 = new Scanner(System.in);  //User will be asked to enter the count of strings  System.out.print("Enter number of strings you would like to enter: ");  count = scan.nextInt();  String[] str = new String[count];  //User is entering the strings and they are stored in an array  System.out.println("Enter the Strings one by one: ");  for(int i = 0; i < count; i++)  {  str[i] = scan2.nextLine();  }  System.out.println("Enter the String to be searched : ");  String searchString = scan3.nextLine();  for (int i = 0; i < count; i++)  {  if(str[i].equals(searchString)){  System.out.println("The required string is in position : "+(i));  System.exit(0);  }  }  scan.close();  scan2.close();  scan3.close();  System.out.println("Could not find required string in Array.");  }  } |

**OUTPUT:**



**PROGRAM NO : 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.

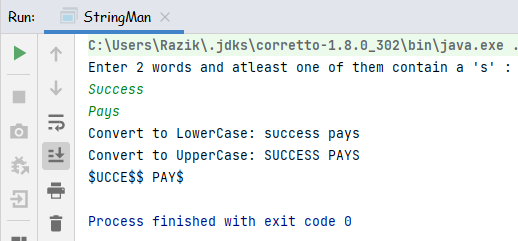
Step 4: Display the combined string after replacing all the ‘s’ & ‘S’ characters with ‘$’ character.

Step 5: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| StringMan.java | import java.util.Scanner;  public class SearchItem {  public static void main(String[] args) {  int count;  String temp;  Scanner scan = new Scanner(System.in);  Scanner scan2 = new Scanner(System.in);  Scanner scan3 = new Scanner(System.in);  //User will be asked to enter the count of strings  System.out.print("Enter number of strings you would like to enter: ");  count = scan.nextInt();  String[] str = new String[count];  //User is entering the strings and they are stored in an array  System.out.println("Enter the Strings one by one: ");  for(int i = 0; i < count; i++)  {  str[i] = scan2.nextLine();  }  System.out.println("Enter the String to be searched : ");  String searchString = scan3.nextLine();  for (int i = 0; i < count; i++)  {  if(str[i].equals(searchString)){  System.out.println("The required string is in position : "+(i));  System.exit(0);  }  }  scan.close();  scan2.close();  scan3.close();  System.out.println("Could not find required string in Array.");  }  } |

**OUTPUT:**



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

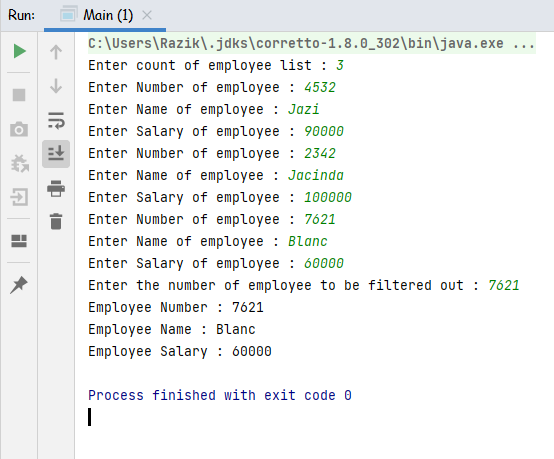
Step 3: If user provided ‘eNo’ is found inside the Employee object list, display the details of the corresponding employee.

Step 4: Stop

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Employee.java | public class Employee {  public String eNo;  public String eName;  public int eSalary;  public Employee(String eNo,String eName,int eSalary)  {  this.eNo = eNo;  this.eName = eName;  this.eSalary = eSalary;  }  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args) {  Scanner scan = new Scanner(System.in);  System.out.print("Enter count of employee list : ");  int count = Integer.parseInt(scan.nextLine());  Employee[] employees = new Employee[count];  for (int i=0;i<count;i++)  {  System.out.print("Enter Number of employee : ");  String eNo = scan.nextLine();  System.out.print("Enter Name of employee : ");  String eName = scan.nextLine();  System.out.print("Enter Salary of employee : ");  int eSalary = Integer.parseInt(scan.nextLine());  employees[i] = new Employee(eNo,eName,eSalary);  }  System.out.print("Enter the number of employee to be filtered out : ");  String search = scan.nextLine();  for (int i=0;i<count;i++){  if(employees[i].eNo.equals(search))  {  System.out.println("Employee Number : "+employees[i].eNo);  System.out.println("Employee Name : "+employees[i].eName);  System.out.println("Employee Salary : "+employees[i].eSalary);  }  }  }  } |

**OUTPUT:**



**PROGRAM NO : 10**

**AIM:** Area of different shapes using overloaded functions.

**ALGORITHM :**

Step 1: Start

Step 2: Define the main class

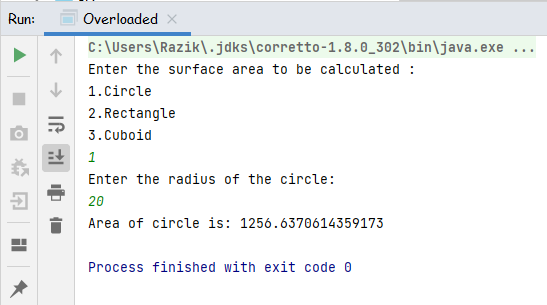
Step 3: Define methods with the same methodname that performs the area operation for each shape

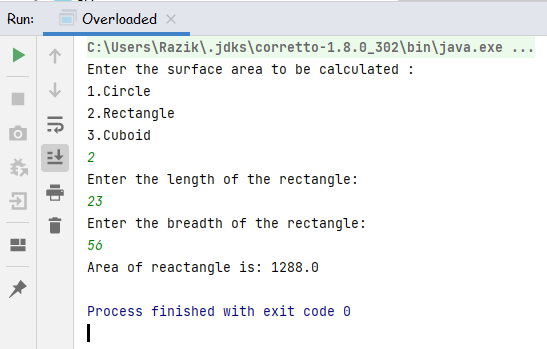
Step 4: Display the areas of each shapes.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Areas.java | public class Areas {  public void area(float length,float breadth)  {  System.out.println("Area of reactangle is: "+length\*breadth);  }  public void area(float radius)  {  System.out.println("Area of circle is: "+(radius\*radius\*Math.PI));  }  public void area(float length,float breadth, float height)  {  float area = 2\*((length\*breadth)+(breadth\*height)+(length\*height));  System.out.println("Area of cuboid is: "+area);  }  } |
| Overloaded.java | import java.util.Scanner;  public class Overloaded {  public static void main(String[] args) {  float a,b,c;  int choice;  Scanner scanner=new Scanner(System.in);  Areas areas = new Areas();  System.out.println("Enter the surface area to be calculated : ");  System.out.println("1.Circle");  System.out.println("2.Rectangle");  System.out.println("3.Cuboid");  choice = scanner.nextInt();  if(choice==1)  {  System.out.println("Enter the radius of the circle: ");  a=scanner.nextInt();  areas.area(a);  }  else if(choice==2)  {  System.out.println("Enter the length of the rectangle: ");  a=scanner.nextInt();  System.out.println("Enter the breadth of the rectangle: ");  b=scanner.nextInt();  areas.area(a,b);  }  else if(choice==3)  {  System.out.println("Enter the length of the cuboid: ");  a=scanner.nextInt();  System.out.println("Enter the breadth of the cuboid: ");  b=scanner.nextInt();  System.out.println("Enter the height of the cuboid: ");  c=scanner.nextInt();  areas.area(a,b,c);  }  else  {  System.out.println("Invalid Choice");  }  }  } |

**OUTPUT:**





**PROGRAM NO : 11**

**AIM:** Create a class ‘Employee’ with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class ‘Teacher’ that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

**ALGORITHM :**

Step 1: Start

Step 2: create class “employee” with the provided data members and define the constructors

Step 3: create another class “Teachers” that performs inheritance of employee class and define constructors for the same

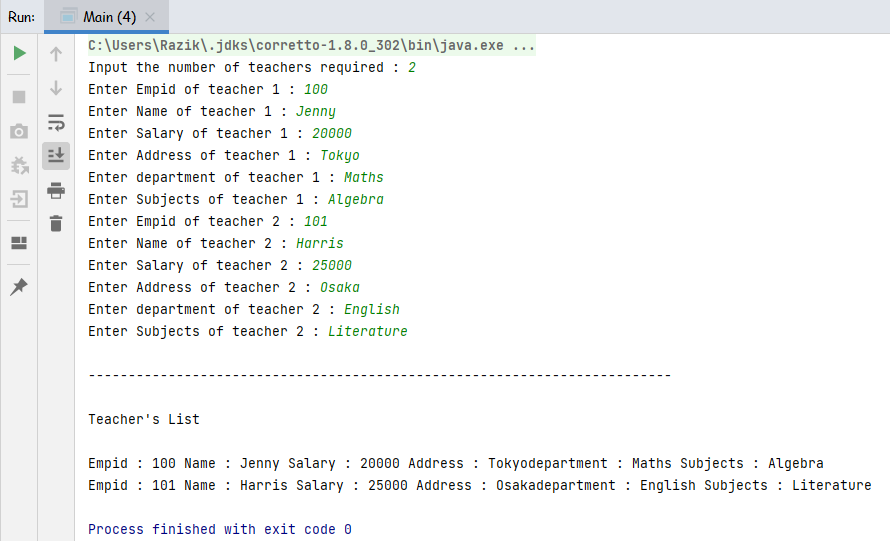
Step 4: create an array of objects in the corresponding class

Step 5: Display the details for the number of teachers provided.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Employee.java | public class Employee {  String Empid;  String Name;  String Salary;  String Address;  Employee(String Empid,String Name,String Salary,String Address)  {  this.Empid = Empid;  this.Name = Name;  this.Salary = Salary;  this.Address = Address;  }  } |
| Teacher.java | public class Teacher extends Employee {  String department;  String Subjects;  Teacher(String Empid, String Name, String Salary, String Address,String department,String Subjects) {  super(Empid, Name, Salary, Address);  this.department = department;  this.Subjects = Subjects;  }  void displayTeacherDetails()  {  System.out.println("Empid : "+this.Empid+" Name : "+this.Name+" Salary : "+this.Salary+" Address : "+this.Address+"department : "+this.department+" Subjects : "+this.Subjects);  }  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args)  {  int count;  Scanner scanner = new Scanner(System.in);  System.out.print("Input the number of teachers required : ");  count = scanner.nextInt();  Teacher[] teachers = new Teacher[count];  for(int i=0;i<count;i++) {  int j = i+1;  System.out.print("Enter Empid of teacher "+j+" : ");  String Empid = scanner.next();  System.out.print("Enter Name of teacher "+j+" : ");  String Name = scanner.next();  System.out.print("Enter Salary of teacher "+j+" : ");  String Salary = scanner.next();  System.out.print("Enter Address of teacher "+j+" : ");  String Address = scanner.next();  System.out.print("Enter department of teacher "+j+" : ");  String department = scanner.next();  System.out.print("Enter Subjects of teacher "+j+" : ");  String Subjects = scanner.next();  teachers[i] = new Teacher(Empid, Name, Salary, Address, department, Subjects);  }  System.out.println("\n-------------------------------------------------------------------------\n");  System.out.println("Teacher's List \n");  for(int i=0;i<count;i++) {  teachers[i].displayTeacherDetails();  }  }  } |

**OUTPUT:**



**PROGRAM NO : 12**

**AIM:** Write a user defined exception class to authenticate the user name and password.

**ALGORITHM :**

Step 1: Start

Step 2: create class “person” with the provided data members and define the constructors

Step 3: create another class “employee” that performs inheritance of person class and another class “teacher” that further inherits the properties of its former class

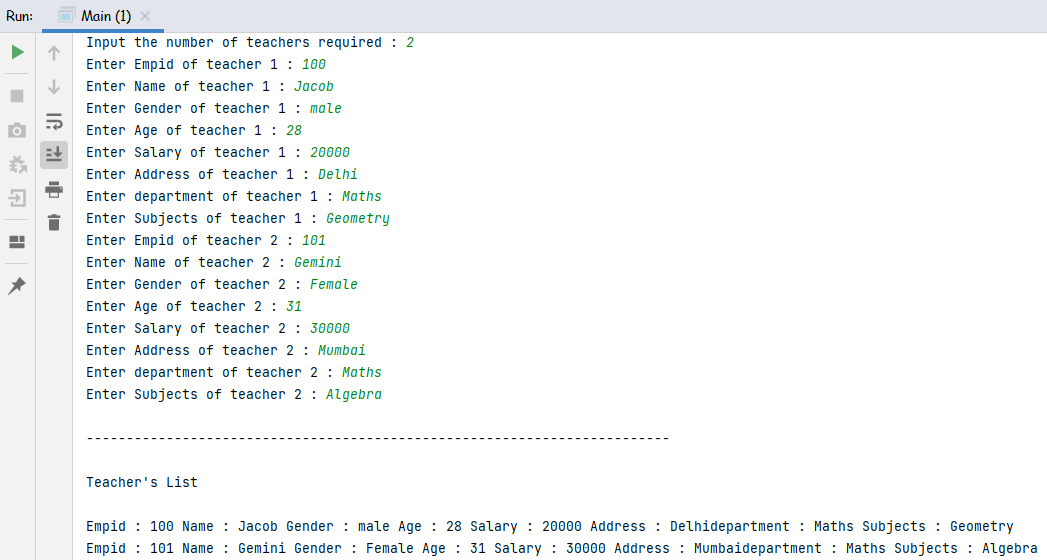
Step 4: create an array of objects in the corresponding class

Step 5: Display the details for the number of teachers provided.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Employee.java | public class Employee extends Person{  String Empid;  String Salary;  Employee(String Name,String Gender,String Address,String Age,String Empid,String Salary)  {  super(Name,Gender,Address,Age);  this.Empid = Empid;  this.Salary = Salary;  }  } |
| Teacher.java | public class Teacher extends Employee {  String department;  String Subjects;  Teacher(String Name,String Gender,String Address,String Age,String Empid,String Salary,String department,String Subjects) {  super(Name,Gender,Address,Age,Empid,Salary);  this.department = department;  this.Subjects = Subjects;  }  void displayTeacherDetails()  {  System.out.println("Empid : "+this.Empid+" Name : "+this.Name+" Gender : "+this.Gender+" Age : "+this.Age+" Salary : "+this.Salary+" Address : "+this.Address+"department : "+this.department+" Subjects : "+this.Subjects);  }  } System.out.println("Empid : "+this.Empid+" Name : "+this.Name+" Salary : "+this.Salary+" Address : "+this.Address+"department : "+this.department+" Subjects : "+this.Subjects);  }  } |
| Person.java | public class Person {  String Name;  String Gender;  String Address;  String Age;  Person(String Name,String Gender,String Address,String Age){  this.Name = Name;  this.Gender = Gender;  this.Address = Address;  this.Age = Age;  }  } |
| Main.java | public class Main {  public static void main(String[] args)  {  int count;  Scanner scanner = new Scanner(System.in);  System.out.print("Input the number of teachers required : ");  count = scanner.nextInt();  Teacher[] teachers = new Teacher[count];  for(int i=0;i<count;i++) {  int j = i+1;  System.out.print("Enter Empid of teacher "+j+" : ");  String Empid = scanner.next();  System.out.print("Enter Name of teacher "+j+" : ");  String Name = scanner.next();  System.out.print("Enter Gender of teacher "+j+" : ");  String Gender = scanner.next();  System.out.print("Enter Age of teacher "+j+" : ");  String Age = scanner.next();  System.out.print("Enter Salary of teacher "+j+" : ");  String Salary = scanner.next();  System.out.print("Enter Address of teacher "+j+" : ");  String Address = scanner.next();  System.out.print("Enter department of teacher "+j+" : ");  String department = scanner.next();  System.out.print("Enter Subjects of teacher "+j+" : ");  String Subjects = scanner.next();  teachers[i] = new Teacher(Name,Gender,Address,Age,Empid,Salary,department,Subjects);  }  System.out.println("\n-------------------------------------------------------------------------\n");  System.out.println("Teacher's List \n");  for(int i=0;i<count;i++) {  teachers[i].displayTeacherDetails();  }  }  } |

**OUTPUT:**



**PROGRAM NO : 13**

**AIM:** Find the average of N positive integers, raising a user defined exception for each negative input.

**ALGORITHM :**

Step 1: Start

Step 2: create class “publisher” and initialize its data members

Step 3: create classes book, literature, fiction. Each class inherit from their subsequent previous class and have its own data members

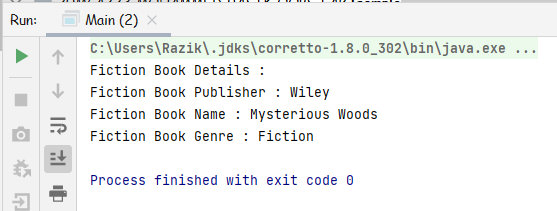
Step 4: create an array of objects in the corresponding class

Step 5: Display the details of the books required.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Book.java | public class Book extends Publisher{  String book = "Mysterious Woods";  } |
| Fiction.java | public class Fiction extends Book{  String genre = "Fiction";  } |
| Literature.java | public class Literature extends Book{  String genre = "Literature";  } |
| Publisher.java | public class Publisher {  String publisher = "Wiley";  } |
| Main.java | public class Main {  public static void main(String[] args) {  Literature literatureBook = new Literature();  Fiction fictionBook = new Fiction();  System.out.println("Fiction Book Details : ");  System.out.println("Fiction Book Publisher : "+fictionBook.publisher);  System.out.println("Fiction Book Name : "+fictionBook.book);  System.out.println("Fiction Book Genre : "+fictionBook.genre);  }  } |

**OUTPUT:**



**PROGRAM NO : 14**

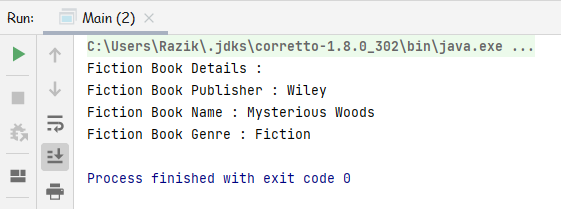
**AIM:** Find the average of N positive integers, raising a user defined exception for each negative input.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Student.java | public class Student {  int maths = 85;  int science = 72;  int english = 88;  int socialScience = 70;  } |
| Sports.java | public class Sports extends Student {  String sport = "Football";  int goals = 2;  int assists = 1;  int minutesPlayed = 81;  int grace = 20;  } |
| Result.java | public class Result extends Sports {  public void displayInfo()  {  System.out.println("Academic Result");  System.out.println("---------------");  System.out.println("Maths : "+this.maths);  System.out.println("Science : "+this.science);  System.out.println("English : "+this.english);  System.out.println("Social Science : "+this.socialScience);  System.out.println("Sports Grace : "+this.grace);  System.out.println("---------------");  System.out.println("\n");  System.out.println("Sports Result");  System.out.println("-------------");  System.out.println("Sport : "+this.sport);  System.out.println("Goals : "+this.goals);  System.out.println("Assists : "+this.assists);  System.out.println("Minutes Played "+this.minutesPlayed);  System.out.println("-------------");  }  } |
| Main.java | public class Main {  public static void main(String[] args) {  Result result = new Result();  result.displayInfo();  }  } |

**OUTPUT:**



**PROGRAM NO : 15**

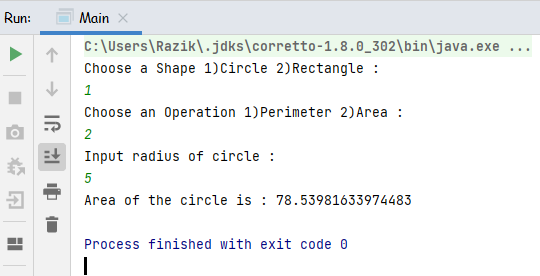
**AIM:** Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Circle.java | public class Circle implements Shape{  int radius;  Scanner scanner = new Scanner(System.in);  public void perimeter() {  System.out.println("Input radius of circle : ");  radius = scanner.nextInt();  String perimeter = Double.toString(Math.PI\*radius\*2);  System.out.println("Circumference of the circle is : "+perimeter);  }  public void area() {  System.out.println("Input radius of circle : ");  radius = scanner.nextInt();  String area = Double.toString(Math.PI\*radius\*radius);  System.out.println("Area of the circle is : "+area);  }  } |
| Rectangle.java | import java.util.Scanner;  public class Rectangle implements Shape{  int length;  int breadth;  Scanner scanner = new Scanner(System.in);  public void perimeter() {  System.out.println("Input length of rectangle : ");  length = scanner.nextInt();  System.out.println("Input breadth of rectangle : ");  length = scanner.nextInt();  String perimeter = Double.toString(2\*(length+breadth));  System.out.println("Perimeter of the rectangle is : "+perimeter);  }  public void area() {  System.out.println("Input length of rectangle : ");  length = scanner.nextInt();  System.out.println("Input breadth of rectangle : ");  length = scanner.nextInt();  String area = Double.toString(length\*breadth);  System.out.println("Area of the rectangle is : "+area);  }  } |
| Shape.java | public interface Shape {  public void perimeter();  public void area();  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args){  Scanner scanner = new Scanner(System.in);  int shape,operation;  System.out.println("Choose a Shape 1)Circle 2)Rectangle : ");  shape = scanner.nextInt();  System.out.println("Choose an Operation 1)Perimeter 2)Area : ");  operation = scanner.nextInt();  if(shape==1){  Circle circle = new Circle();  if(operation==1){  circle.perimeter();  }  else if(operation==2)  {  circle.area();  }  else {  System.out.println("Operation code.");  }  }  else if(shape==2)  {  Rectangle rectangle = new Rectangle();  if(operation==1){  rectangle.perimeter();  }  else if(operation==2)  {  rectangle.area();  }  else {  System.out.println("Operation code :");  System.exit(0);  }  }  else {  System.out.println("Incorrect Shape code.");  }  }  } |

**OUTPUT:**



**PROGRAM NO : 16**

**AIM:** Prepare bill with the given format using calculate method from interface :

Order No.

Date :

Product Id Name Quantity unit price Total

-----------------------------------------

101 A 2 25 50

102 B 1 100 100

-----------------------------------------

Net. Amount 150

**ALGORITHM :**

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Bill.java | public interface Bill {  String productId = "";  String productName="";  int unitPrice = 0;  int quantity = 0;  int total = 0;  public void printBillItem();  public void printBillHeader();  public void printBillFooter(int billTotal);  } |
| ProductBill.java | import java.time.format.DateTimeFormatter;  import java.time.LocalDateTime;  public class ProductBill implements Bill {  String productId = "";  String productName="";  int unitPrice = 0;  int quantity = 0;  int total = 0;  ProductBill(String productId,String productName,int unitPrice,int quantity){  this.productId = productId;  this.productName = productName;  this.unitPrice = unitPrice;  this.quantity = quantity;  this.total = unitPrice\*quantity;  }  public void printBillHeader() {  System.out.println("Order No : " + Math.random() \* 1000);  DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");  LocalDateTime now = LocalDateTime.now();  System.out.println("Date : " + dtf.format(now));  System.out.println("Product ID Name Quantity Unit Price Total ");  System.out.println("-----------------------------------------------------------------");  }  public void printBillItem()  {  System.out.format("%10s%20s%10d%12d%12d \n",this.productId,this.productName,this.unitPrice,this.quantity,this.total);  }  public void printBillFooter(int billTotal)  {  System.out.println("-----------------------------------------------------------------");  System.out.format("%64s \n","Net. Amount : "+billTotal);  }  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args)  {  Scanner scanner = new Scanner(System.in);  System.out.println("Input Number of items in Bill : ");  int count = scanner.nextInt();  ProductBill[] productBill=new ProductBill[count];  int billTotal=0;  for(int i=0;i<count;i++) {  System.out.println("Enter Product ID : ");  String productId = scanner.next();  System.out.println("Enter Name : ");  String name = scanner.next();  System.out.println("Enter Quantity : ");  int qty = scanner.nextInt();  System.out.println("Unit Price : ");  int up = scanner.nextInt();  productBill[i]=new ProductBill(productId,name,up,qty);  }  if(count>0){  productBill[0].printBillHeader();  for(int i=0;i<count;i++) {  productBill[i].printBillItem();  billTotal += productBill[i].total;  }  productBill[0].printBillFooter(billTotal);  }  }  } |

**OUTPUT:**



