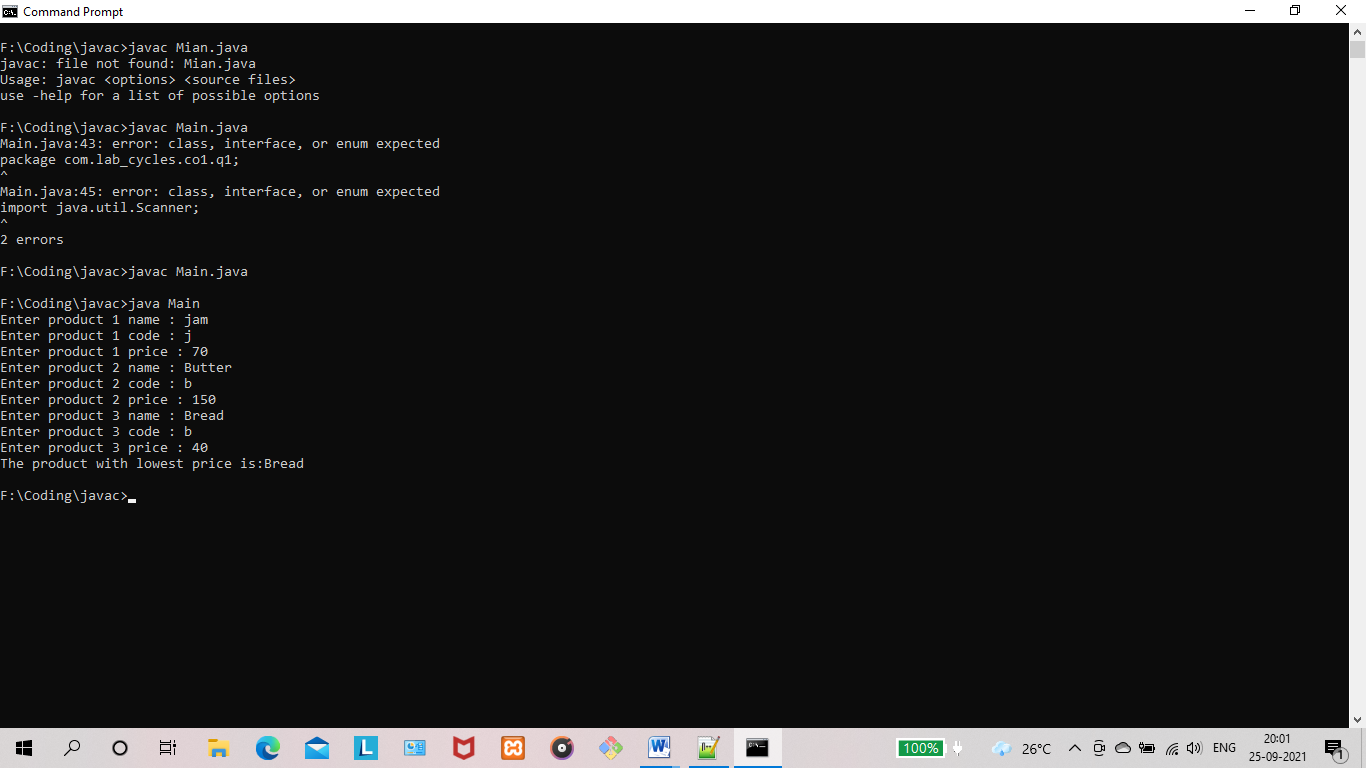
**PROGRAM NO : 1**

**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.

**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);  }  }  } |
| Product.java | public class Product {  public String pcode;  public String pname;  public int price;  } |

**OUTPUT:**

****

**RESULT:** The program is successfully executed and the output is verified.

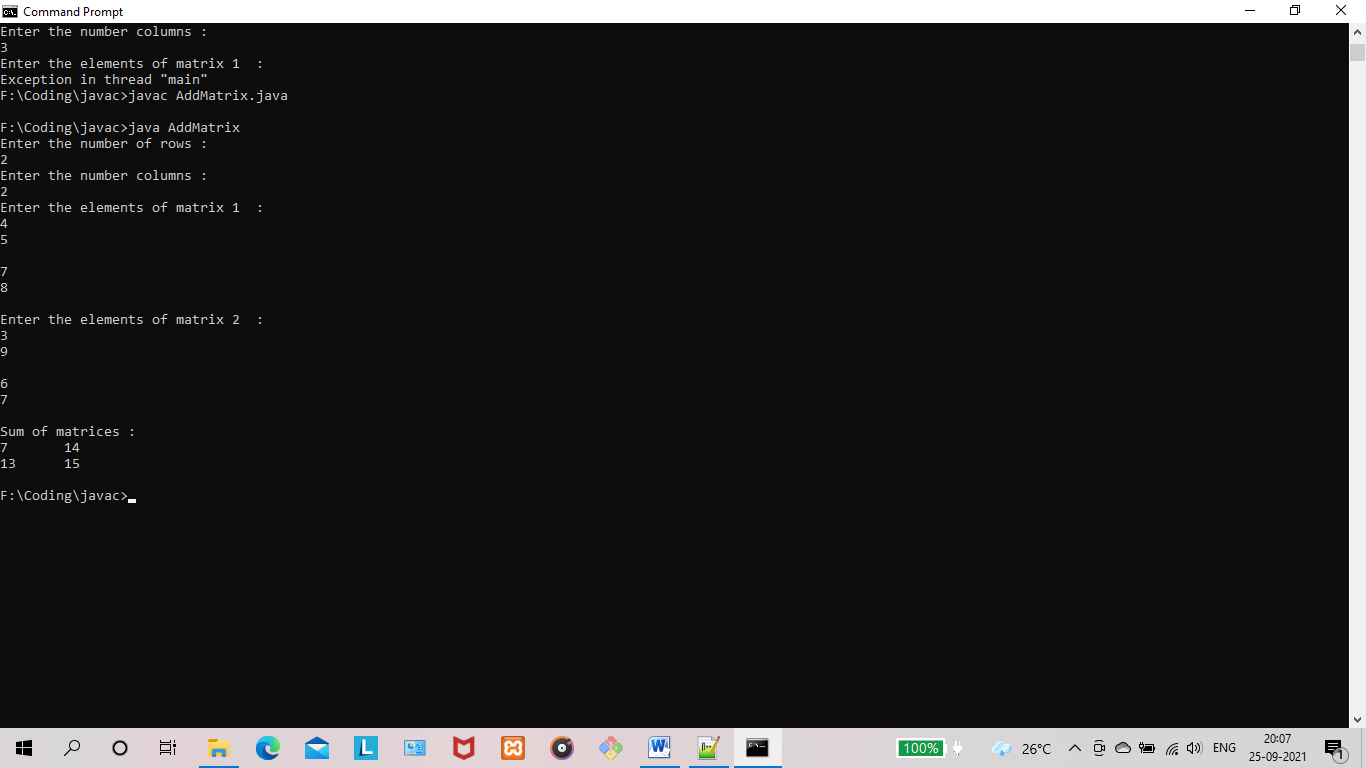
**PROGRAM NO : 2**

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

**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:**

****

**RESULT:** The program is successfully executed and the output is verified.

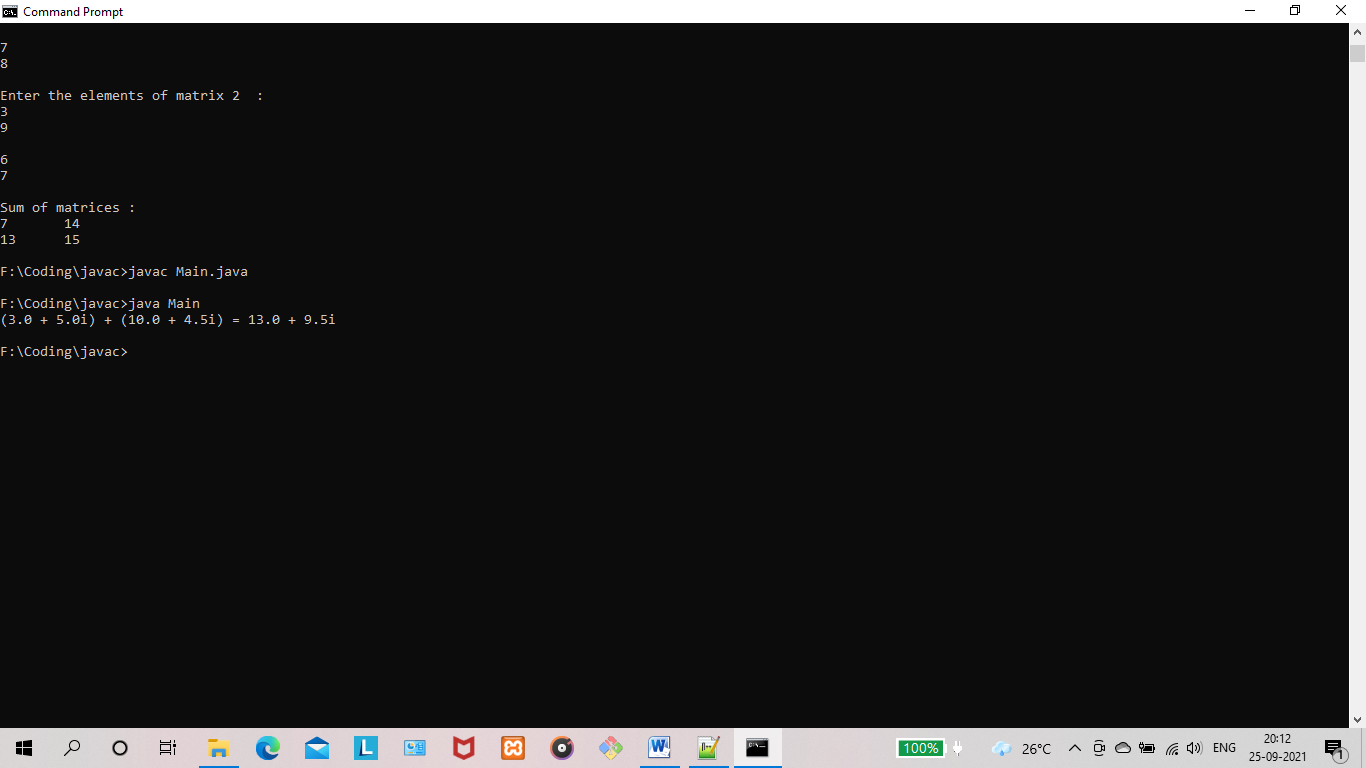
**PROGRAM NO : 3**

**AIM:** Add complex numbers.

**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:**



**RESULT:** The program is successfully executed and the output is verified.

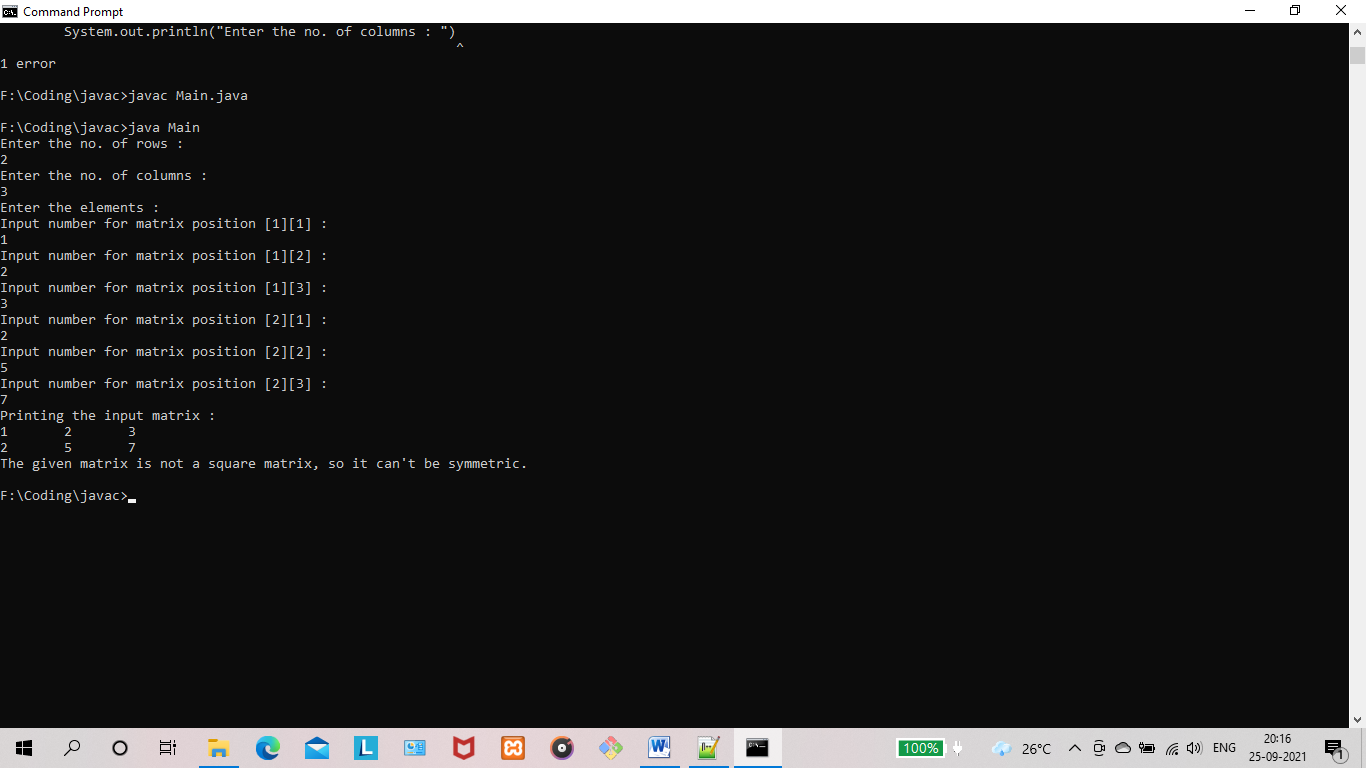
**PROGRAM NO : 4**

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

**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:**



**RESULT:** The program is successfully executed and the output is verified.

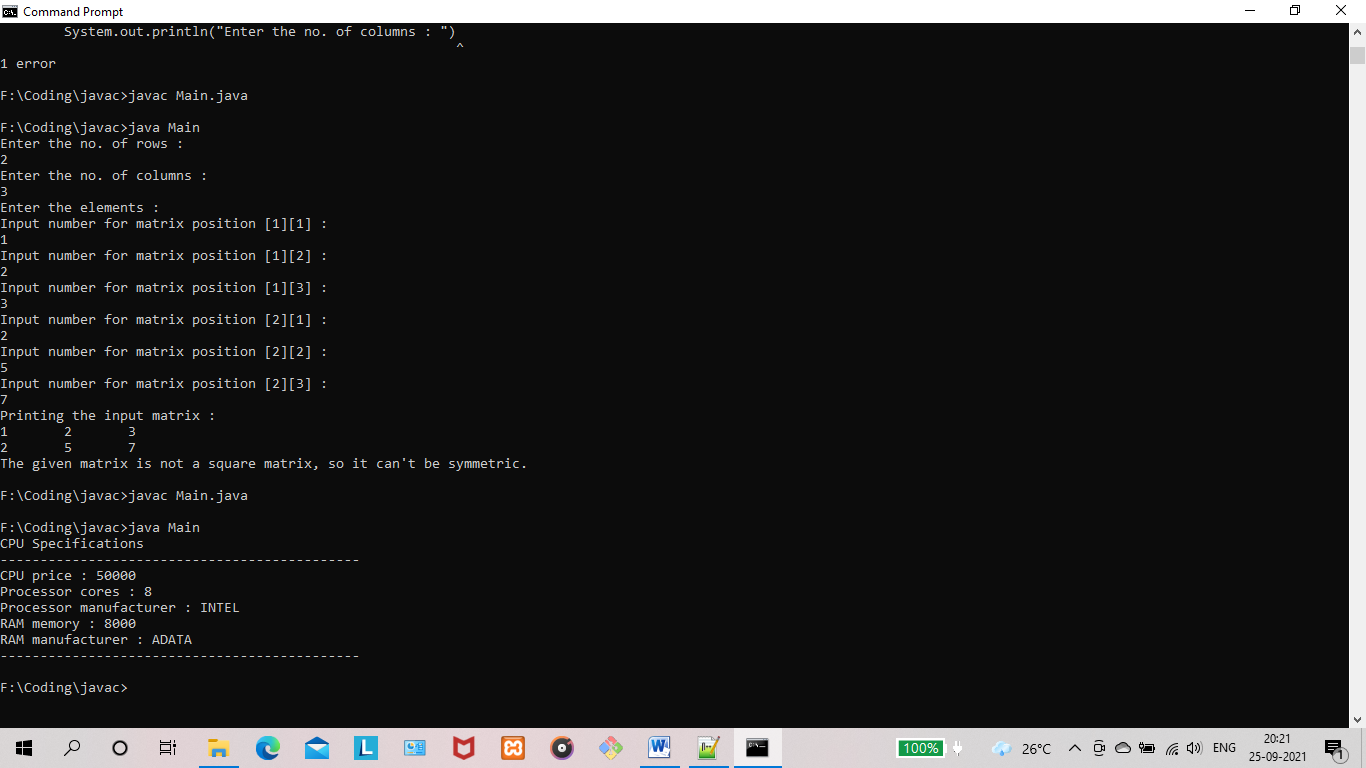
**PROGRAM NO : 5**

**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.

**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:**

****

**RESULT:** The program is successfully executed and the output is verified.

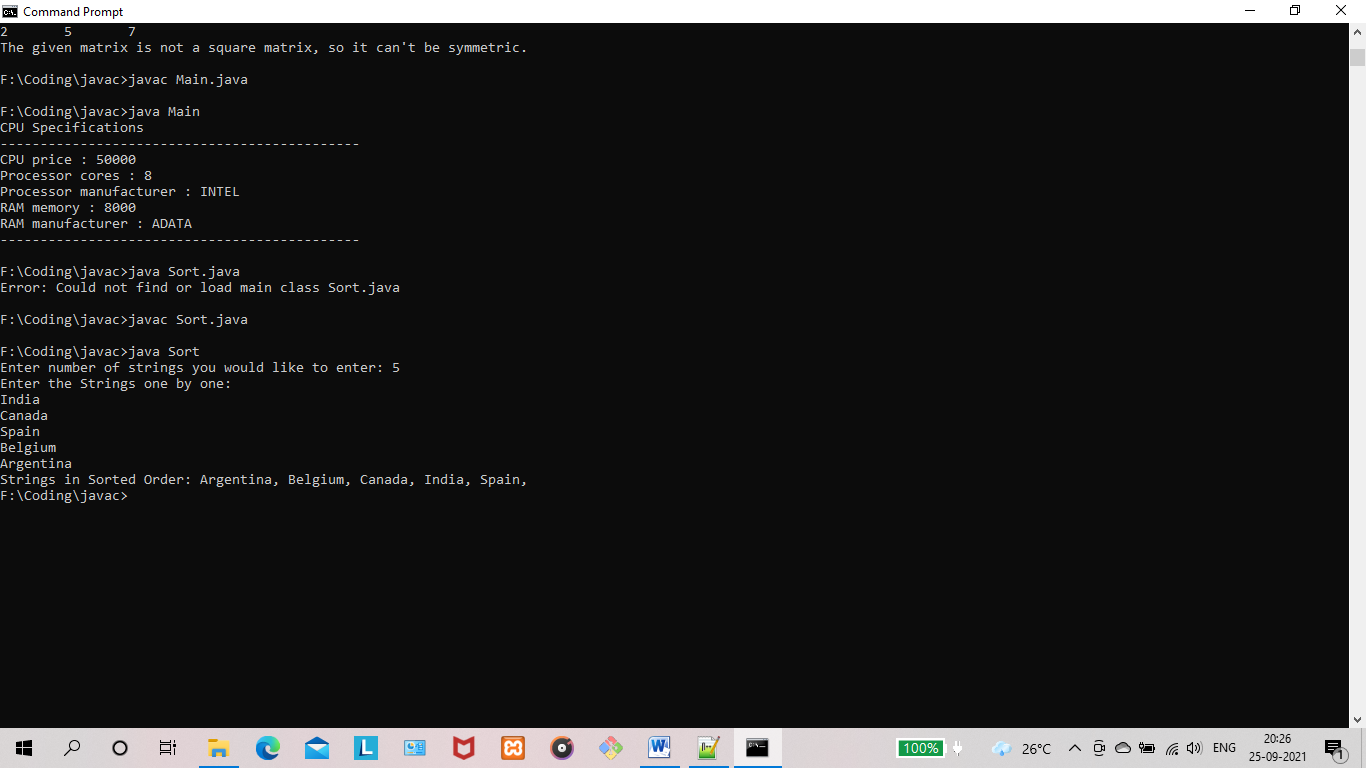
**PROGRAM NO : 6**

**AIM:** Program to Sort strings.

**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:**



**RESULT:** The program is successfully executed and the output is verified.

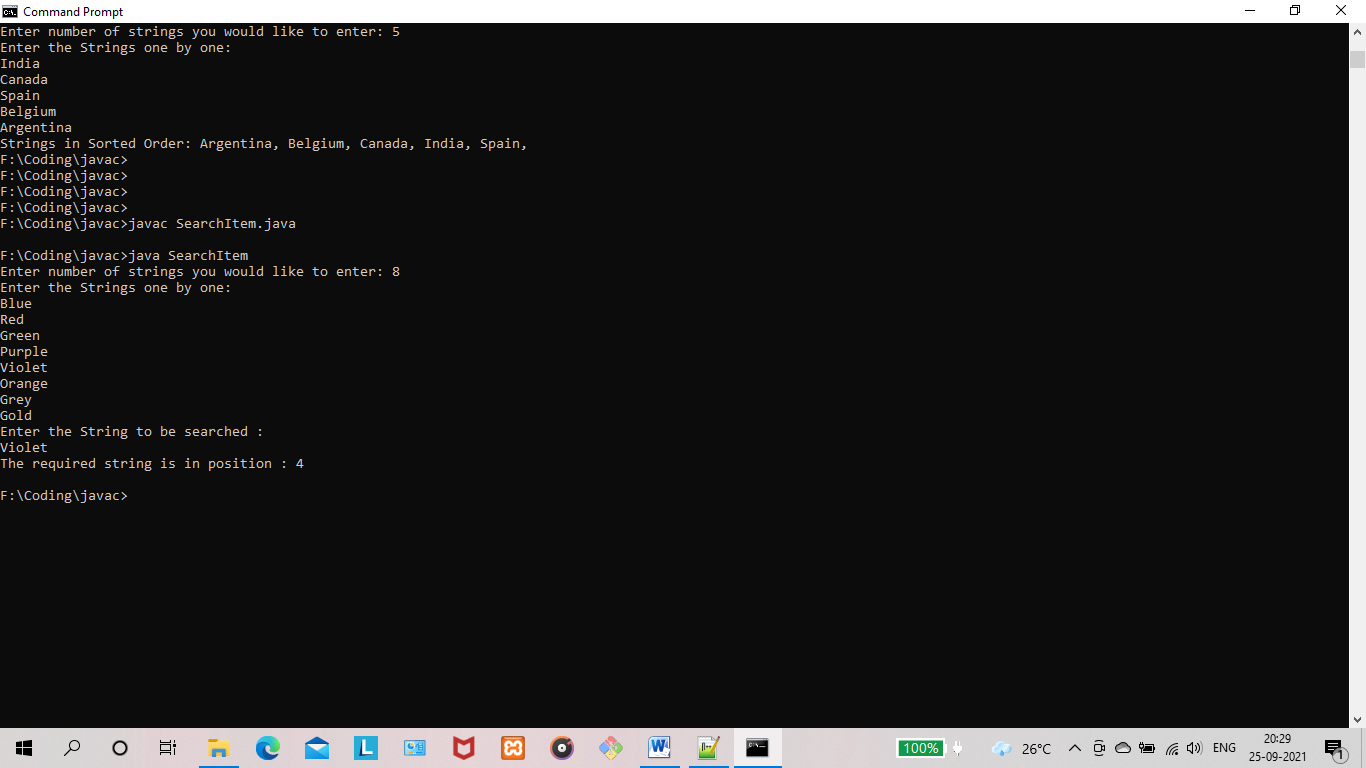
**PROGRAM NO : 7**

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

**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:**



**RESULT:** The program is successfully executed and the output is verified.

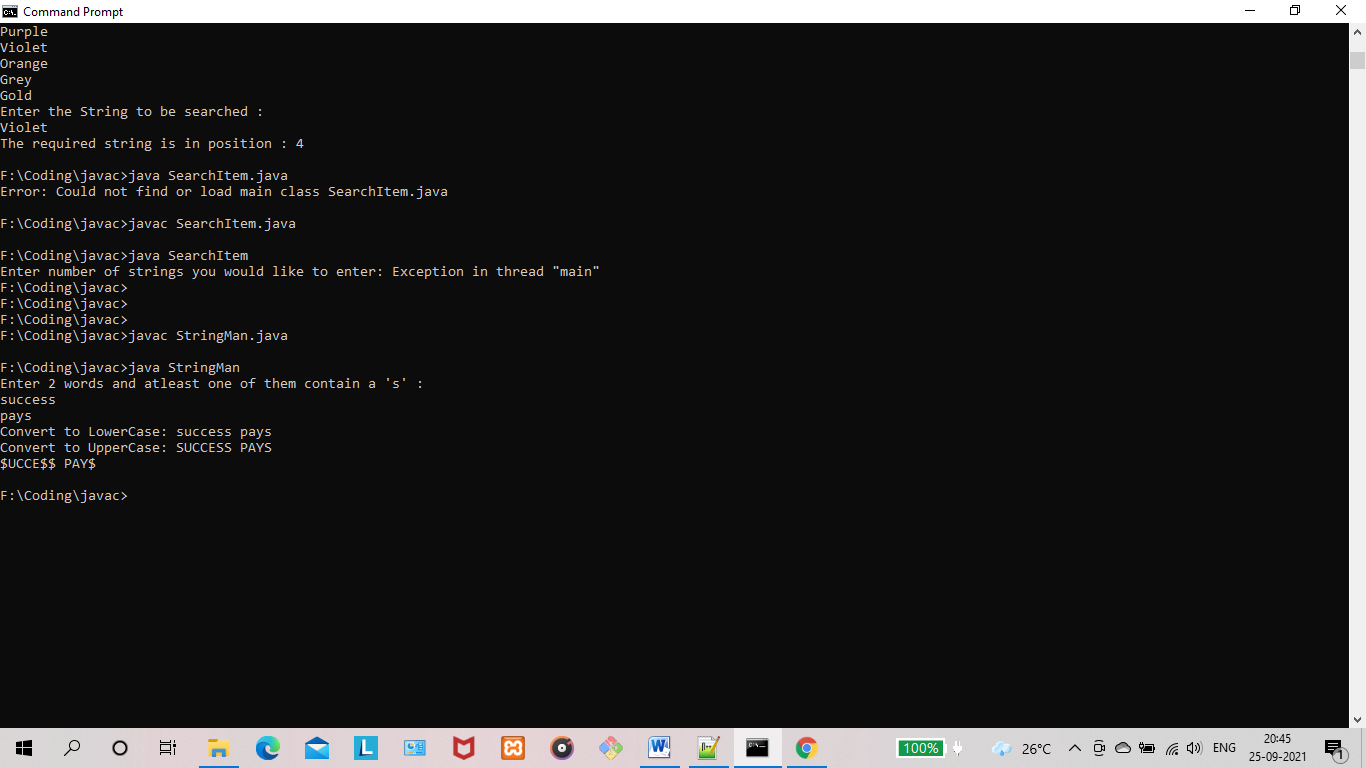
**PROGRAM NO : 8**

**AIM:** Perform string manipulations

**PROGRAM CODE:**

|  |  |
| --- | --- |
|  | import java.util.Scanner;  public class StringMan {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  System.out.println("Enter 2 words and atleast one of them contain a 's' : ");  String word1 = scanner.nextLine();  String word2 = scanner.nextLine();  String str\_Sample = word1+" "+word2;  System.out.println("Convert to LowerCase: " + str\_Sample.toLowerCase());  System.out.println("Convert to UpperCase: " + str\_Sample.toUpperCase());  System.out.println(str\_Sample.toUpperCase().replace("S", "$"));  }  } |

**OUTPUT:**



**RESULT:** The program is successfully executed and the output is verified.

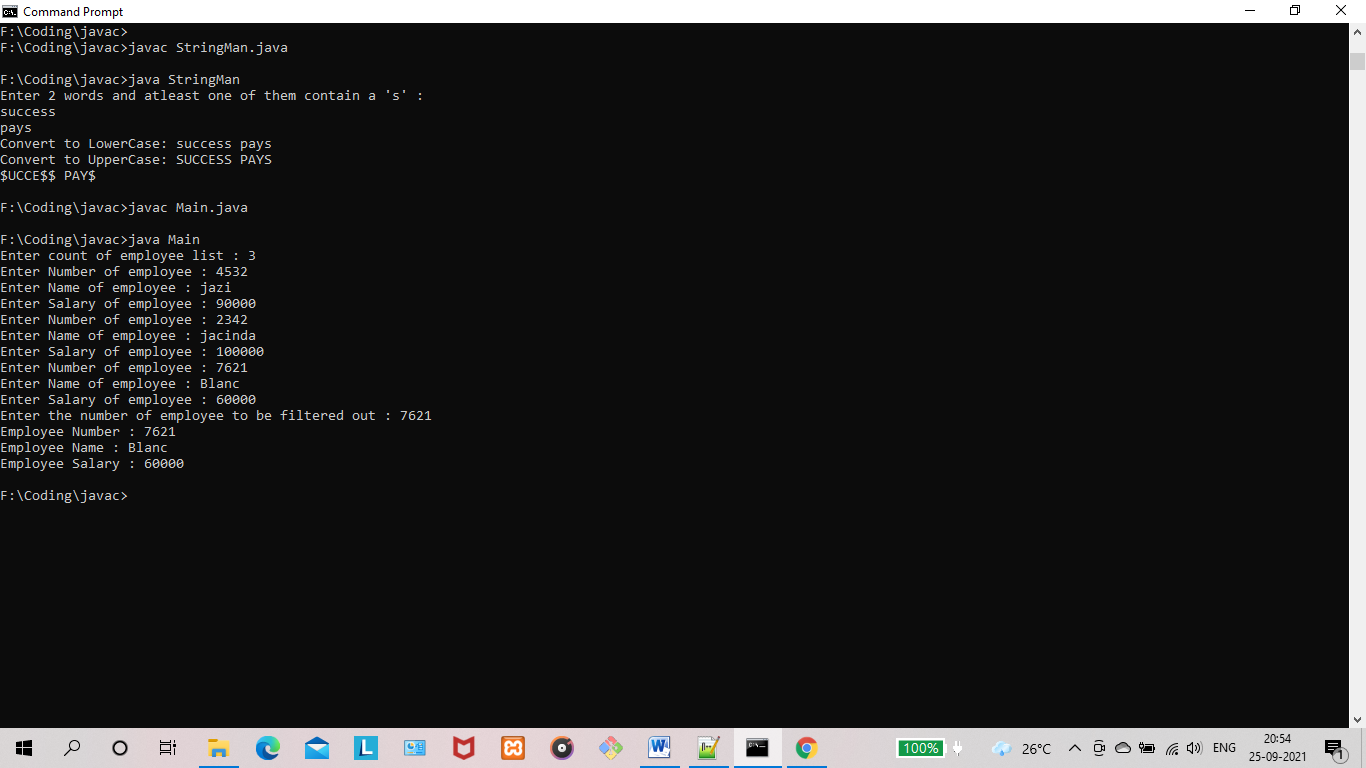
**PROGRAM NO : 9**

**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.

**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:**



**RESULT:** The program is successfully executed and the output is verified.

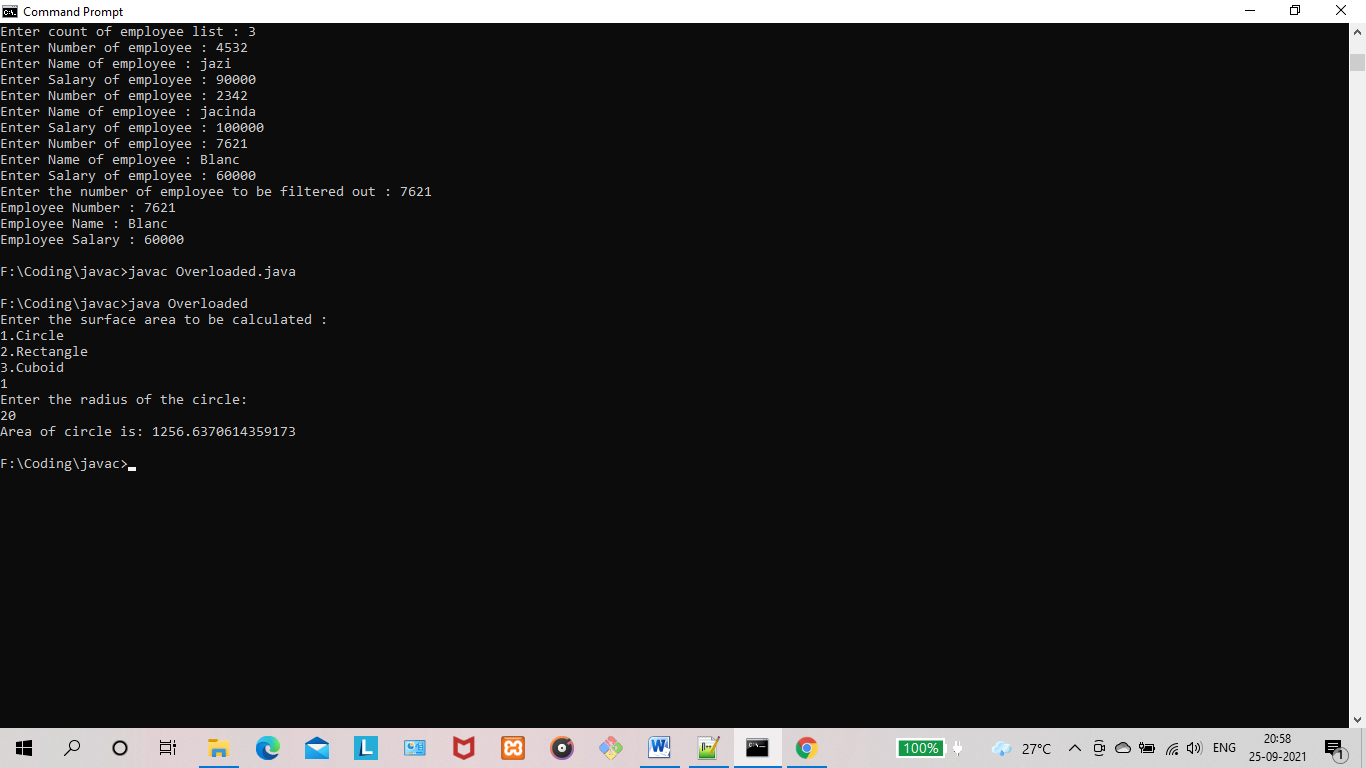
**PROGRAM NO : 10**

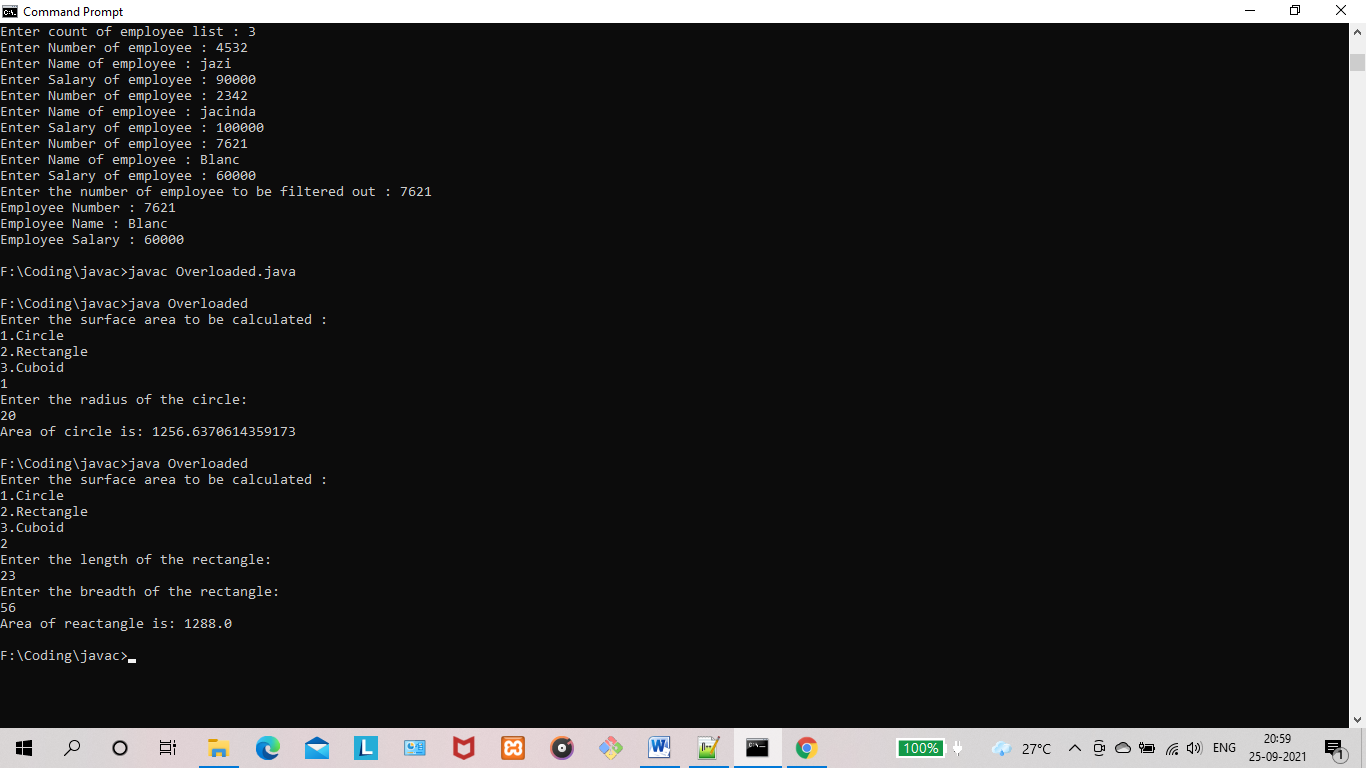
**AIM:** Area of different shapes using overloaded functions.

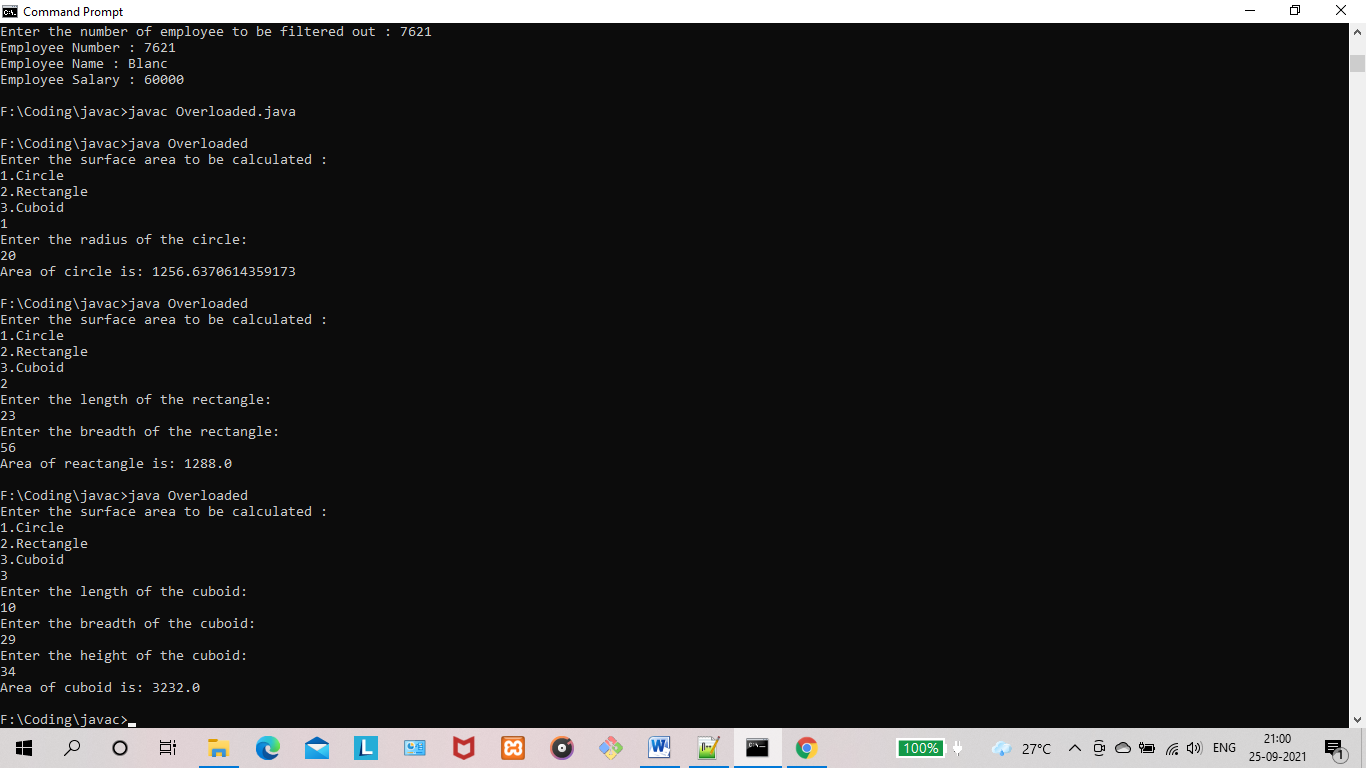
**PROGRAM CODE:**

|  |  |
| --- | --- |
| 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:**







**RESULT:** The program is successfully executed and the output is verified.

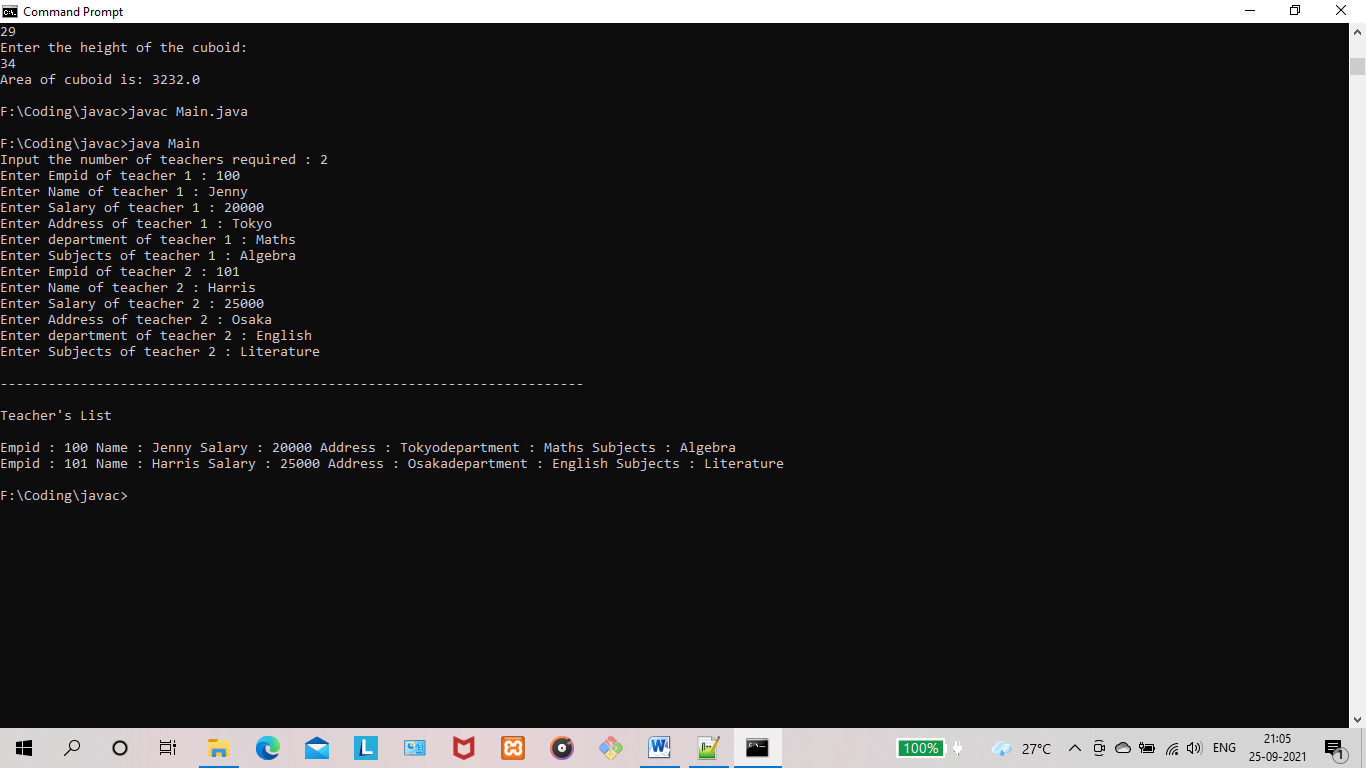
**PROGRAM NO : 11**

**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.

**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:**

****

**RESULT:** The program is successfully executed and the output is verified.

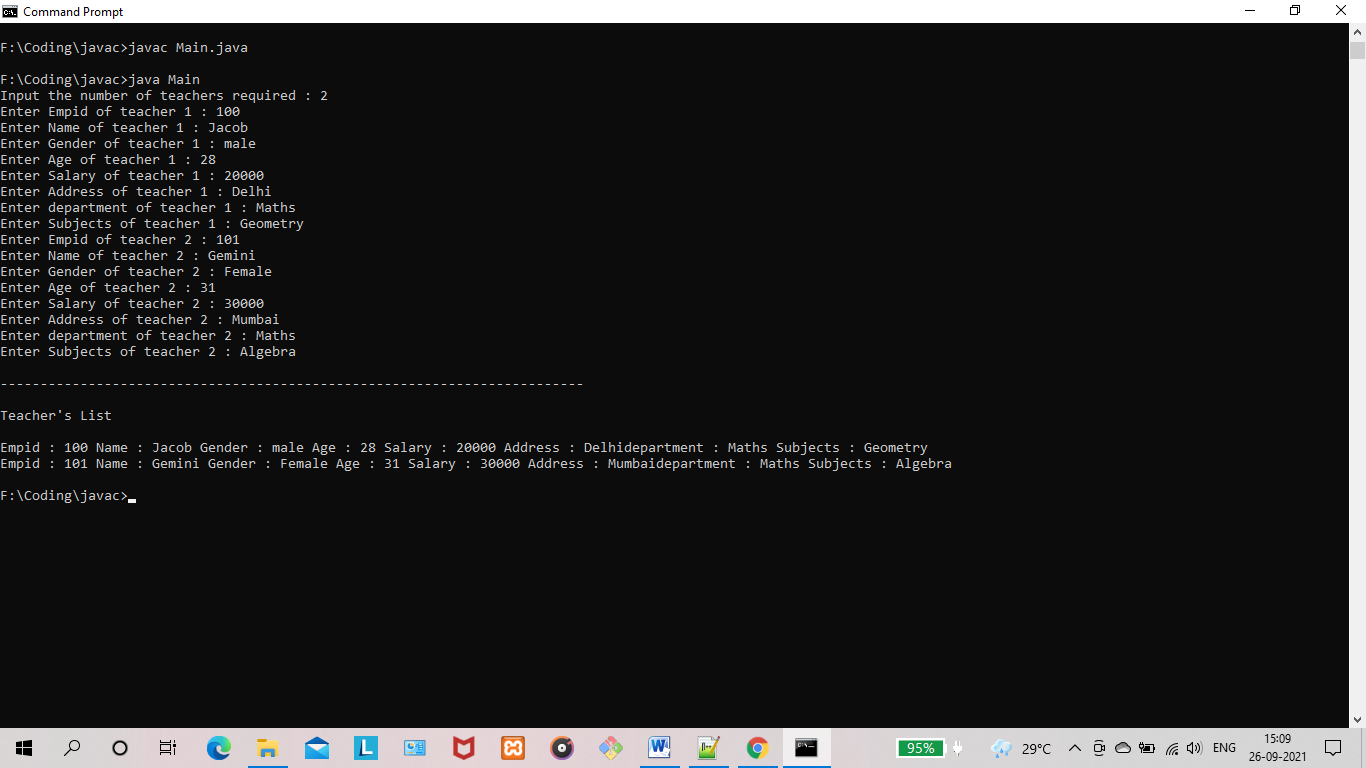
**PROGRAM NO : 12**

**AIM:** Create a class ‘Person’ with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class ‘Employee’ that inherits the properties of class Person and also contains its own data members like Empid, Company\_name, Qualification, Salary and its own constructor. Create another class ‘Teacher’ that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacherid and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.

**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);  }  } |
| 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:**



**RESULT:** The program is successfully executed and the output is verified.

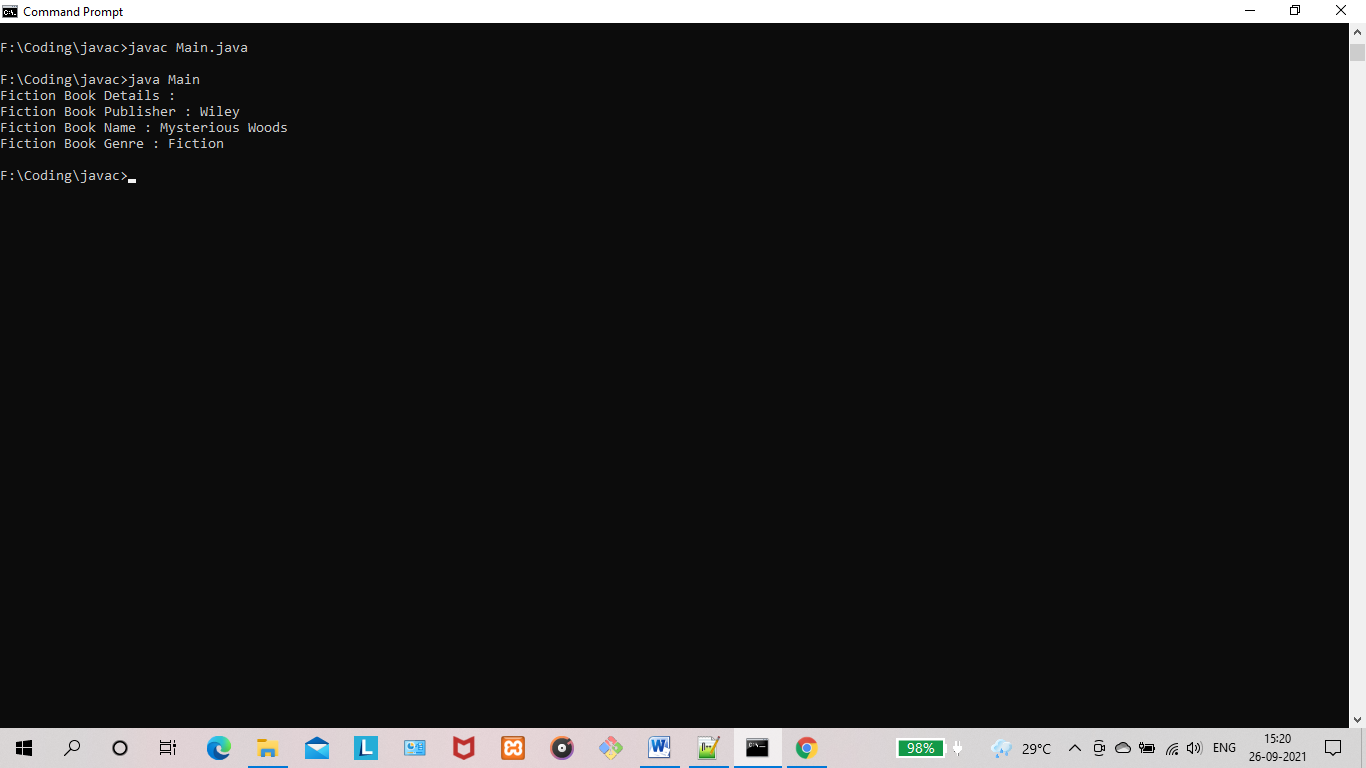
**PROGRAM NO : 13**

**AIM:** Write a program has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance.

**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:**



**RESULT:** The program is successfully executed and the output is verified.

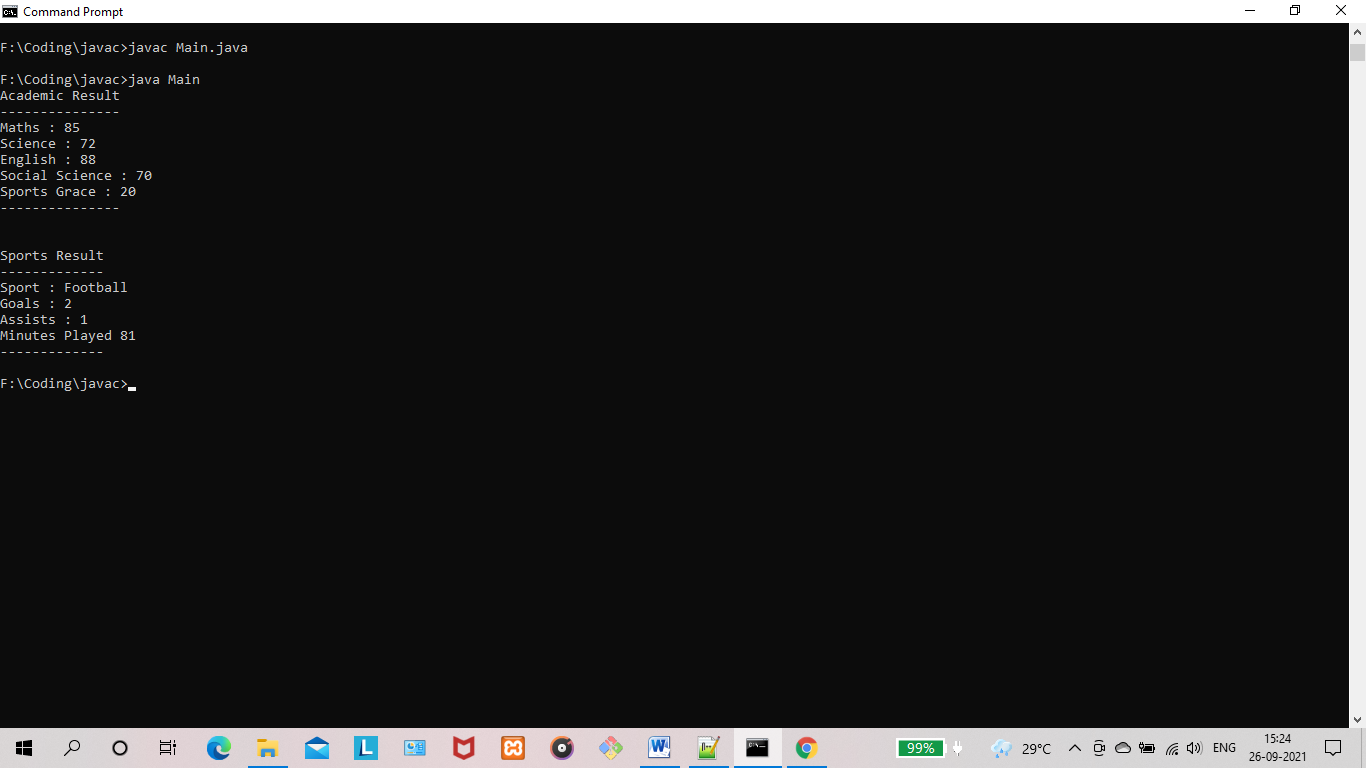
**PROGRAM NO : 14**

**AIM:** Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.

**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:**



**RESULT:** The program is successfully executed and the output is verified.

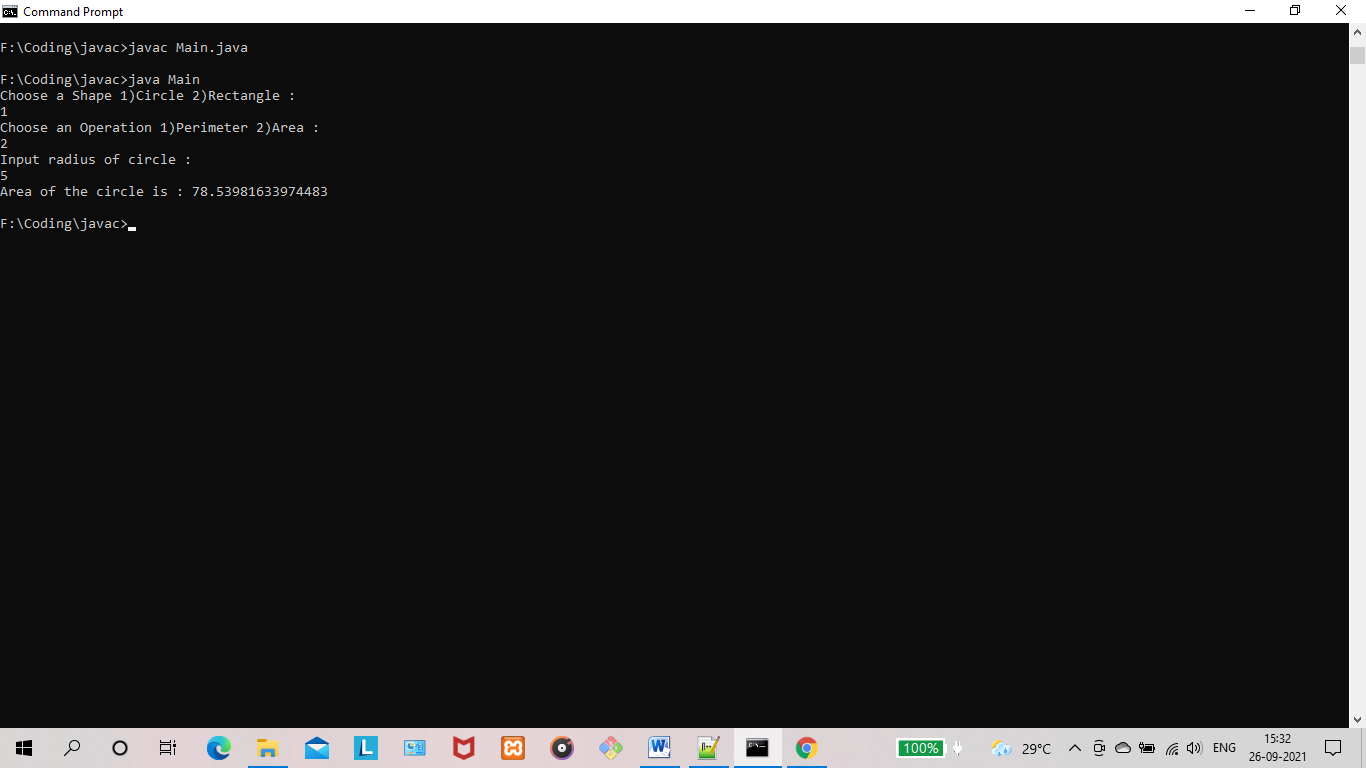
**PROGRAM NO : 15**

**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.

**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:**





**RESULT:** The program is successfully executed and the output is verified.

**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

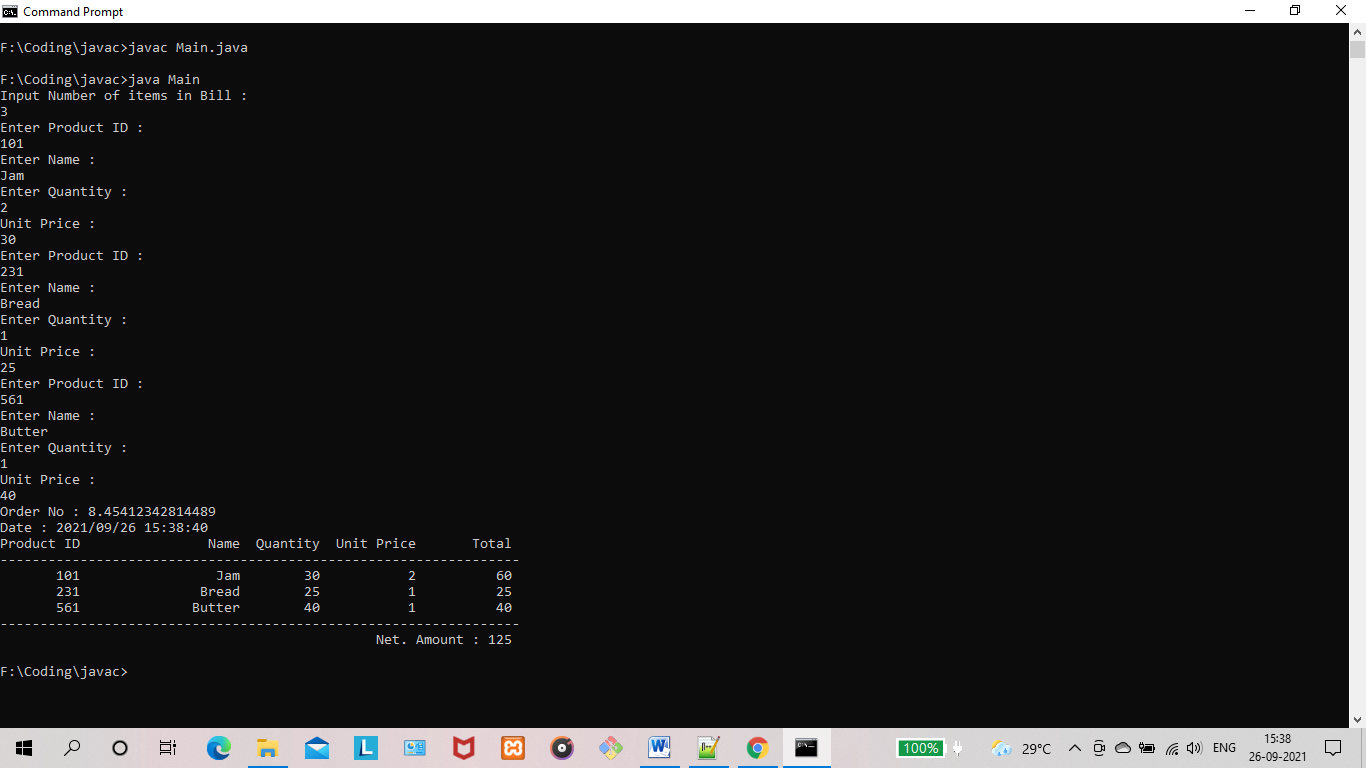
-----------------------------------------

Net. Amount 150

**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:**



**RESULT:** The program is successfully executed and the output is verified.

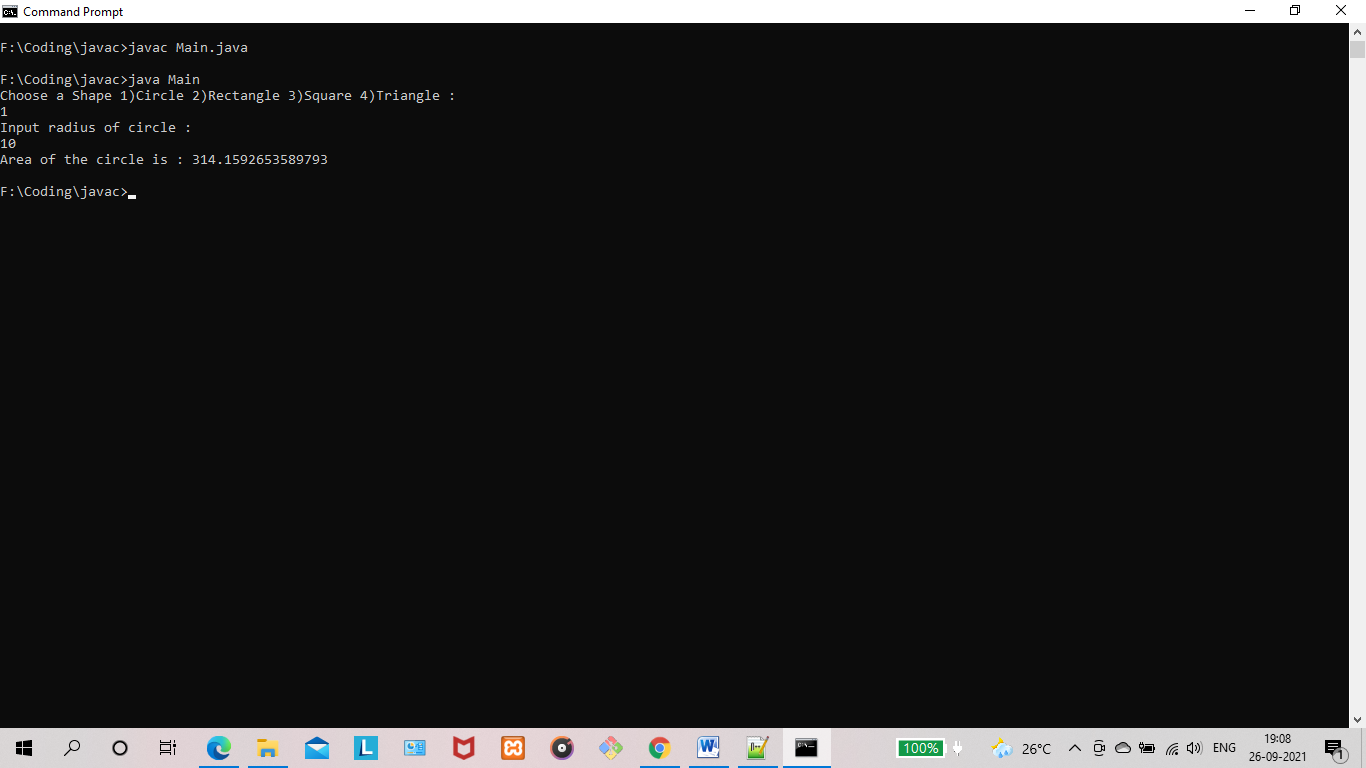
**PROGRAM NO : 17**

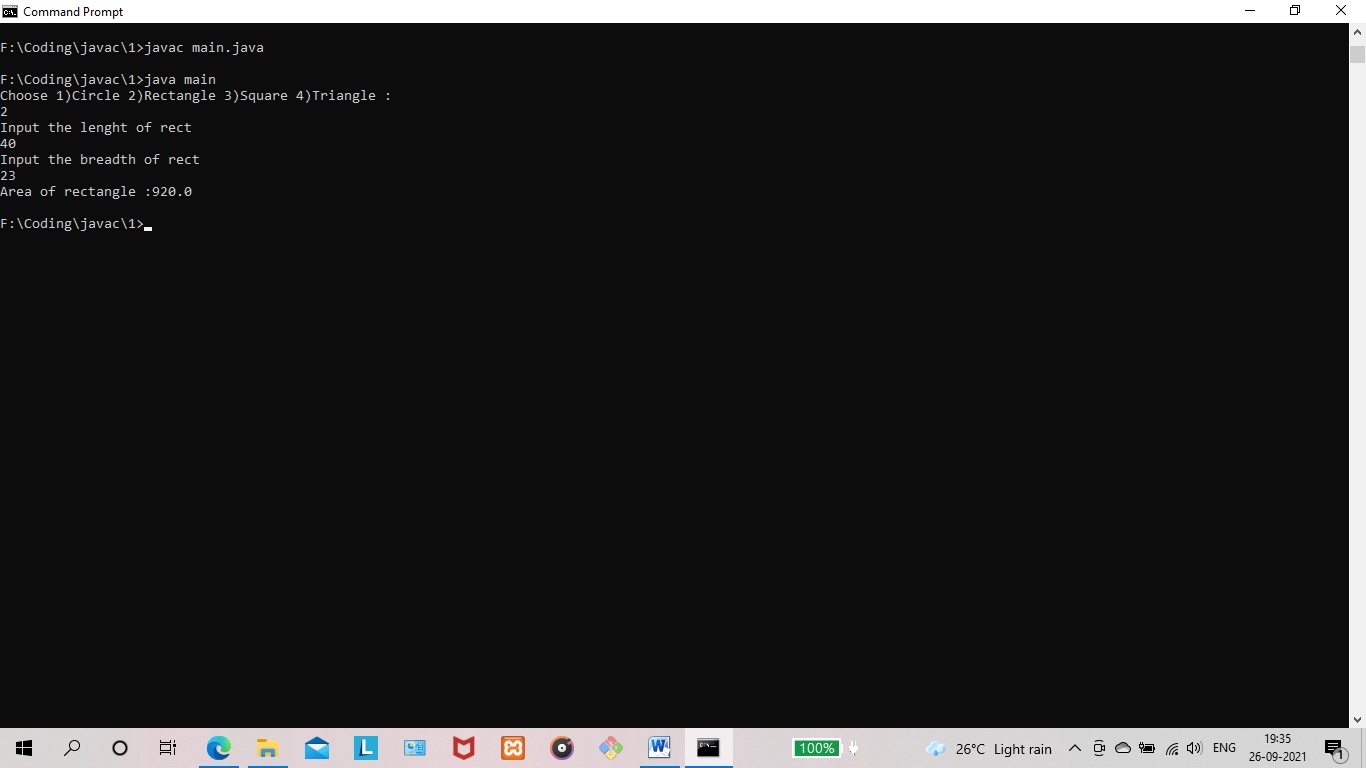
**AIM:** Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.

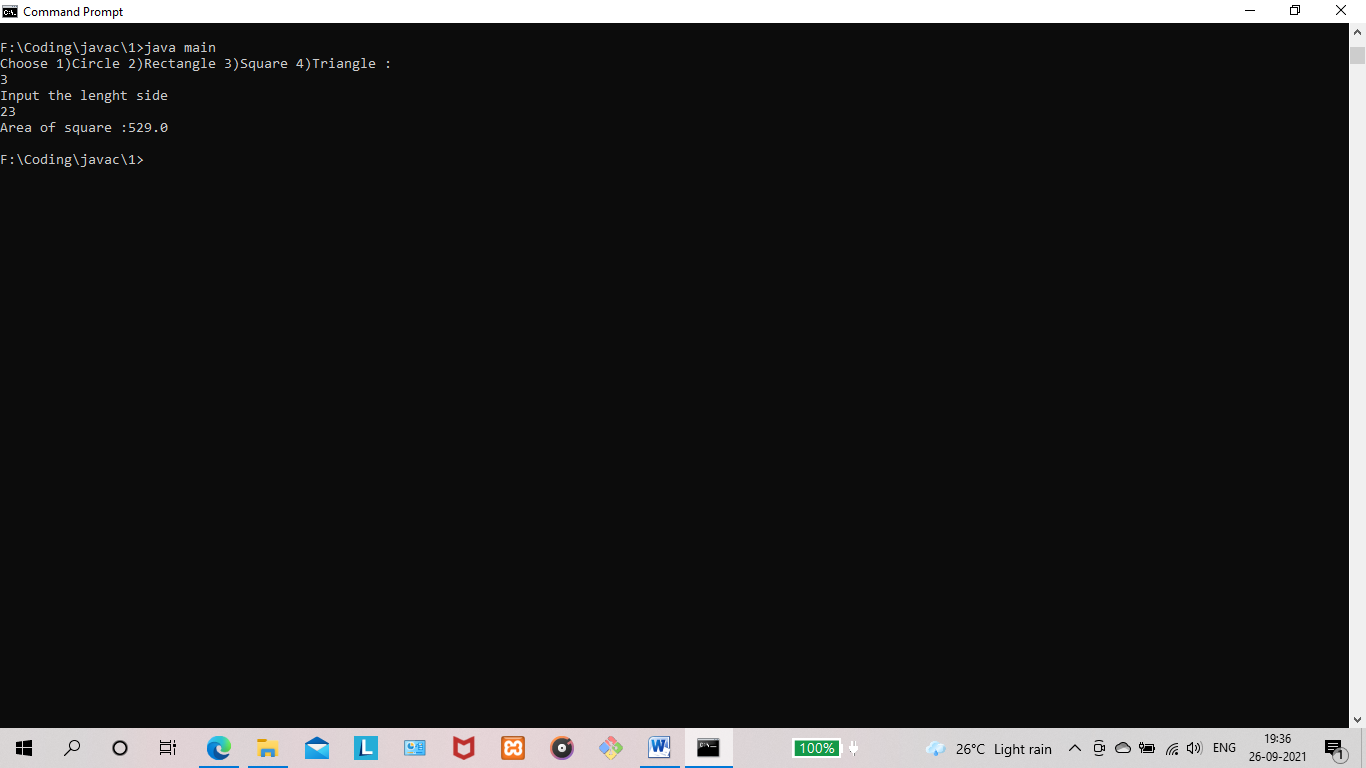
**PROGRAM CODE:**

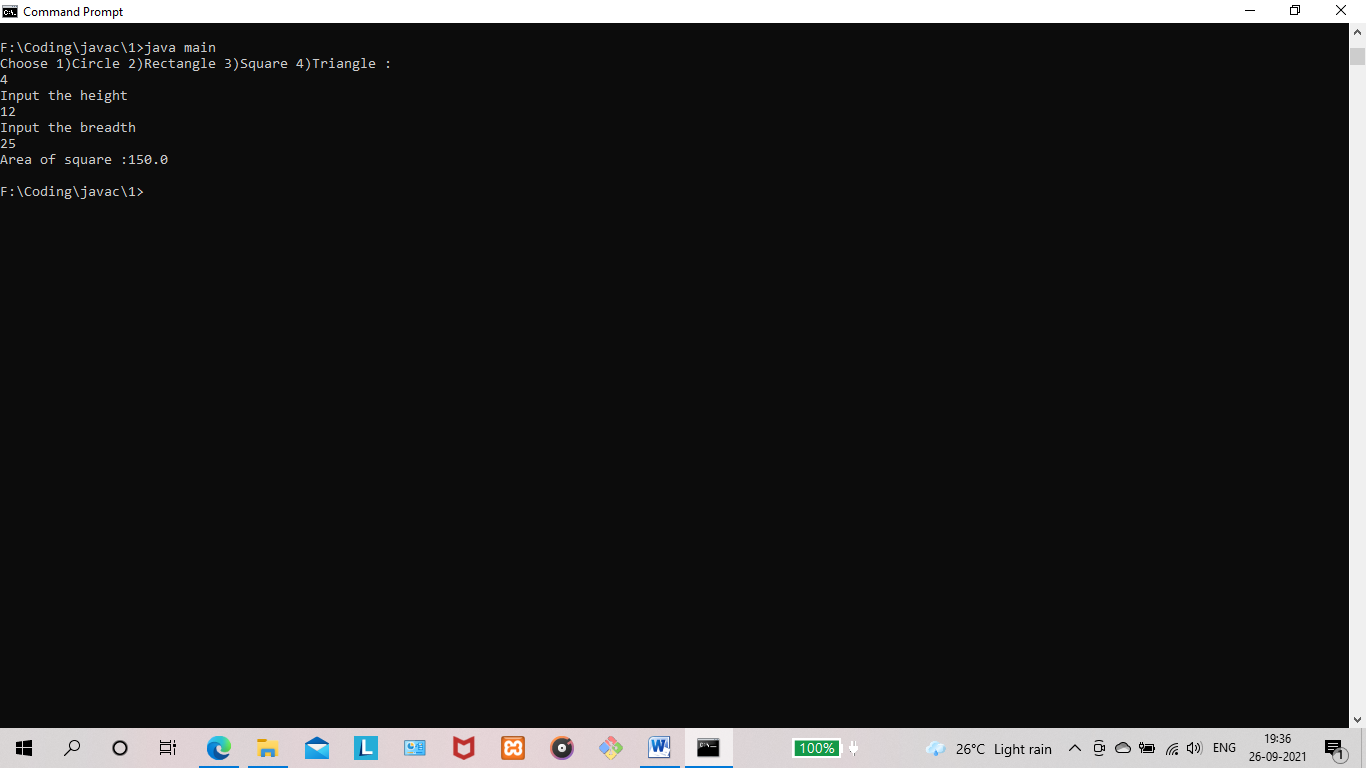
|  |  |
| --- | --- |
| Graphics/  Circle.java | import java.util.Scanner;  public class Circle implements Shape {  int radius;  Scanner scanner = new Scanner(System.in);  public void area() {  System.out.println("Input radius of circle : ");  radius = scanner.nextInt();  String area = Double.toString(Math.PI\*radius\*radius);  System.out.println("Area of the circle is : "+area);  }  } |
| Graphics/  Rectangle.java | import java.util.Scanner;  public class Rectangle implements Shape {  int length;  int breadth;  Scanner scanner = new Scanner(System.in);  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);  }  } |
| Graphics/  Shape.java | public interface Shape {  public void area();  } |
| Graphics/  Square.java | import java.util.Scanner;  public class Square {  int side;  Scanner scanner = new Scanner(System.in);  public void area() {  System.out.println("Input side length of square : ");  side = scanner.nextInt();  String area = Double.toString(side\*side);  System.out.println("Area of the square : "+area);  }  } |
| Graphics/  Triangle.java | import java.util.Scanner;  public class Triangle {  int height;  int breadth;  Scanner scanner = new Scanner(System.in);  public void area() {  System.out.println("Input height of the triangle : ");  height = scanner.nextInt();  System.out.println("Input breadth of triangle : ");  breadth = scanner.nextInt();  String area = Double.toString((height\*breadth)/2f);  System.out.println("Area of the triangle is : "+area);  }  } |
| Main.java | import com.lab\_cycles.co4.q1.Graphics.Circle;  import com.lab\_cycles.co4.q1.Graphics.Rectangle;  import com.lab\_cycles.co4.q1.Graphics.Square;  import com.lab\_cycles.co4.q1.Graphics.Triangle;  import java.util.Scanner;  public class Main {  public static void main(String[] args){  Scanner scanner = new Scanner(System.in);  int shape;  System.out.println("Choose a Shape 1)Circle 2)Rectangle 3)Square 4)Triangle : ");  shape = scanner.nextInt();  if(shape==1){  Circle circle = new Circle();  circle.area();  }  else if(shape==2)  {  Rectangle rectangle = new Rectangle();  rectangle.area();  }  else if(shape==3)  {  Square square = new Square();  square.area();  }  else if(shape==4)  {  Triangle triangle = new Triangle();  triangle.area();  }  else {  System.out.println("Incorrect Shape code.");  }  }  } |

**OUTPUT:**









**RESULT:** The program is successfully executed and the output is verified.

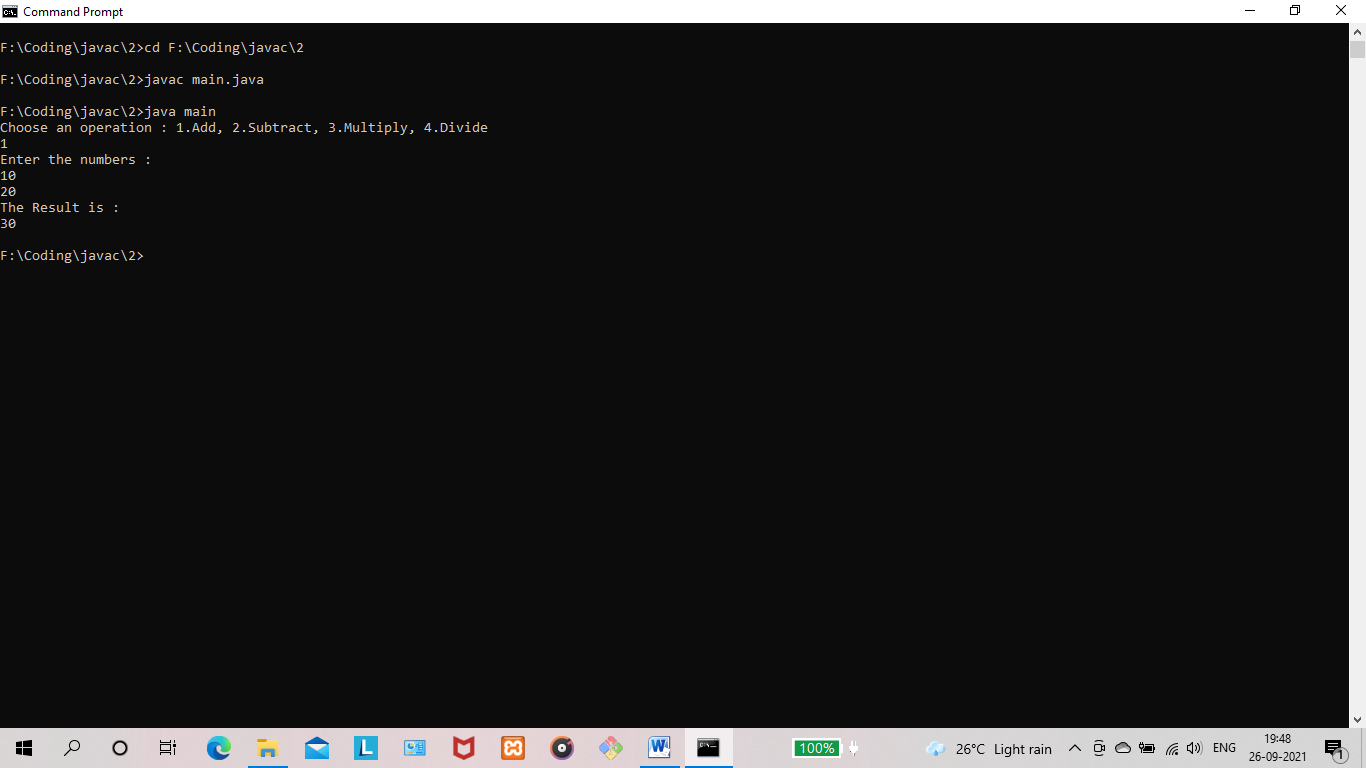
**PROGRAM NO : 18**

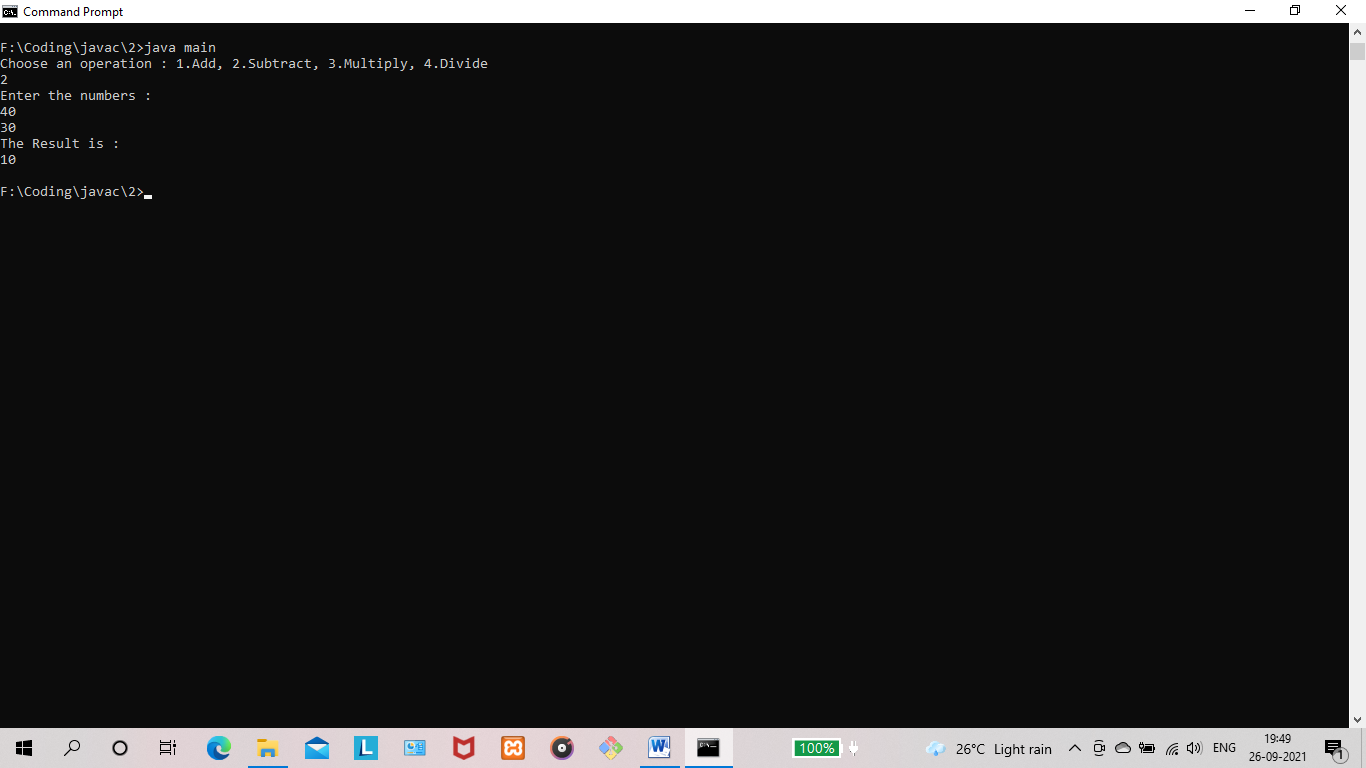
**AIM:** Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers.

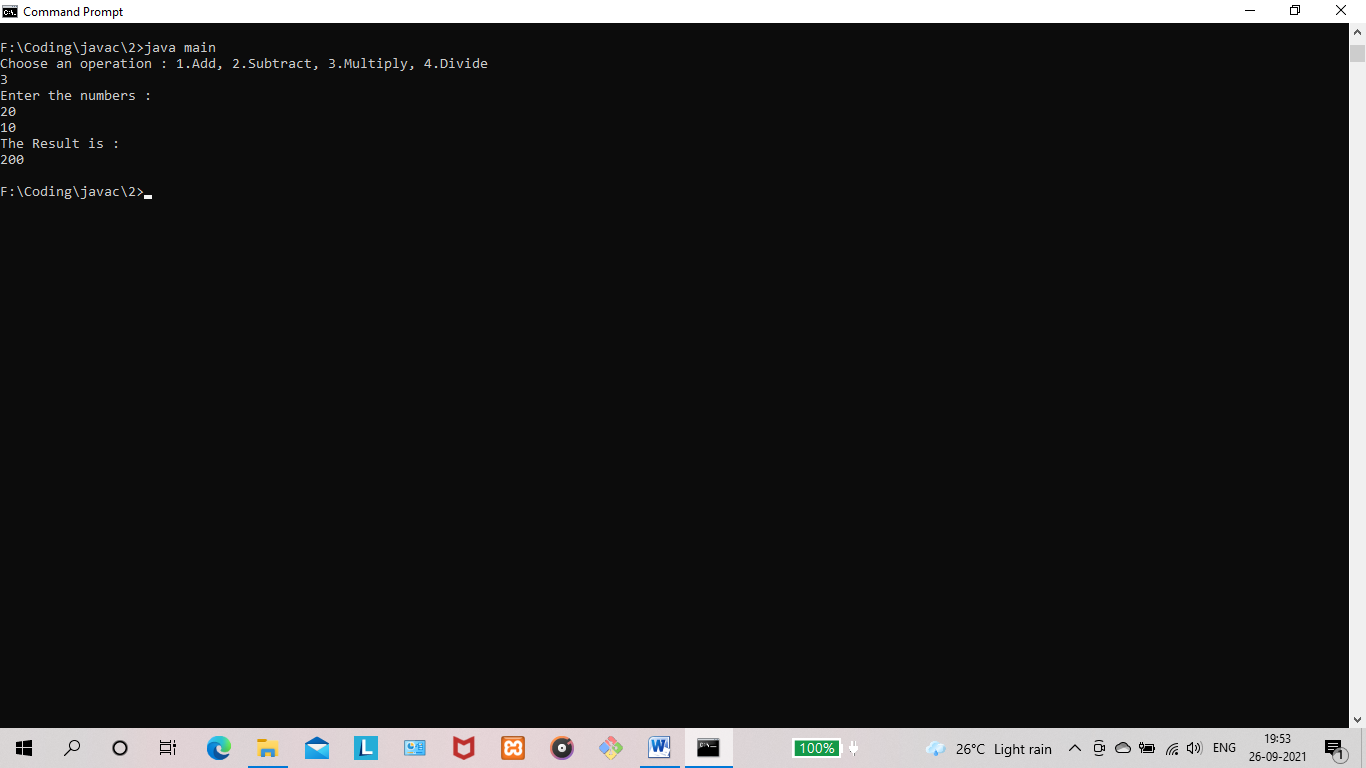
**PROGRAM CODE:**

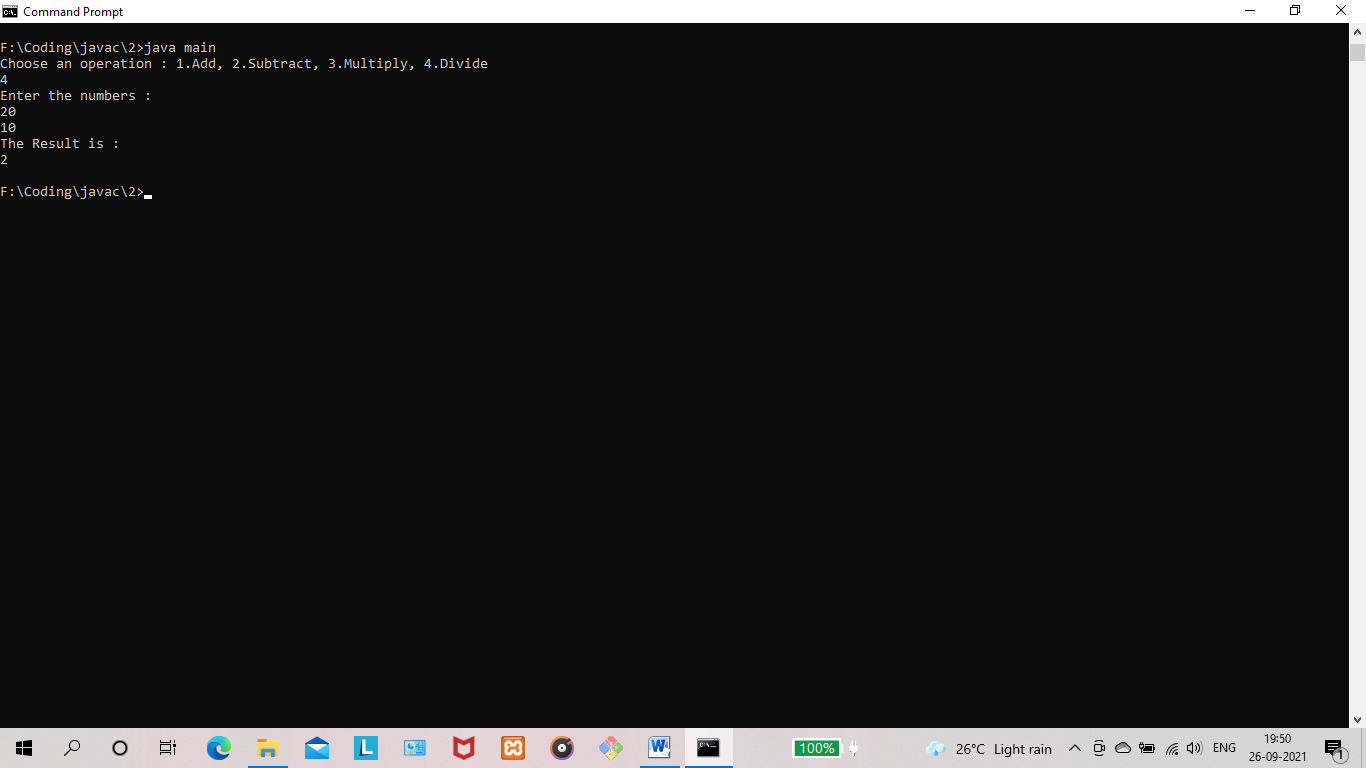
|  |  |
| --- | --- |
| Arithmetic/  AdditionOperation.java | public class AdditionOperation implements ArithmeticOperation {  public int operateNumbers(int number1,int number2)  {  return number1+number2;  }  } |
| Arithmetic/  ArithmeticOperation.java | public interface ArithmeticOperation {  public int operateNumbers(int number1,int number2);  } |
| Arithmetic/  DivisionOperator.java | public class DivisionOperator implements ArithmeticOperation{  public int operateNumbers(int number1,int number2)  {  return number1/number2;  }  } |
| Arithmetic/  MultiplicationOperator.java | public class MultiplicationOperator implements ArithmeticOperation{  public int operateNumbers(int number1,int number2)  {  return number1\*number2;  }  } |
| Arithmetic/  SubtractionOperation.java | public class SubtractionOperation implements ArithmeticOperation{  public int operateNumbers(int number1,int number2)  {  return number1-number2;  }  } |
| Main.java | import com.lab\_cycles.co4.q2.Arithmetic.AdditionOperation;  import com.lab\_cycles.co4.q2.Arithmetic.DivisionOperator;  import com.lab\_cycles.co4.q2.Arithmetic.MultiplicationOperator;  import com.lab\_cycles.co4.q2.Arithmetic.SubtractionOperation;  import java.util.Scanner;  public class Main {  public static void main(String[] args)  {  Scanner scanner = new Scanner(System.in);  System.out.println("Choose an operation : 1.Add, 2.Subtract, 3.Multiply, 4.Divide");  int choice = scanner.nextInt();  System.out.println("Enter the numbers : ");  int number1 = scanner.nextInt();  int number2= scanner.nextInt();  System.out.println("The Result is : 3");  switch (choice){  case 1:  AdditionOperation additionOperation = new AdditionOperation();  System.out.println(additionOperation.operateNumbers(number1,number2));  break;  case 2:  SubtractionOperation subtractionOperation = new SubtractionOperation();  System.out.println(subtractionOperation.operateNumbers(number1,number2));  break;  case 3:  MultiplicationOperator multiplicationOperator = new MultiplicationOperator();  System.out.println(multiplicationOperator.operateNumbers(number1,number2));  break;  case 4:  DivisionOperator divisionOperator = new DivisionOperator();  System.out.println(divisionOperator.operateNumbers(number1,number2));  break;  default:  System.out.println("Invalid Code");  }  }  } |

**OUTPUT:**

****







**RESULT:** The program is successfully executed and the output is verified.

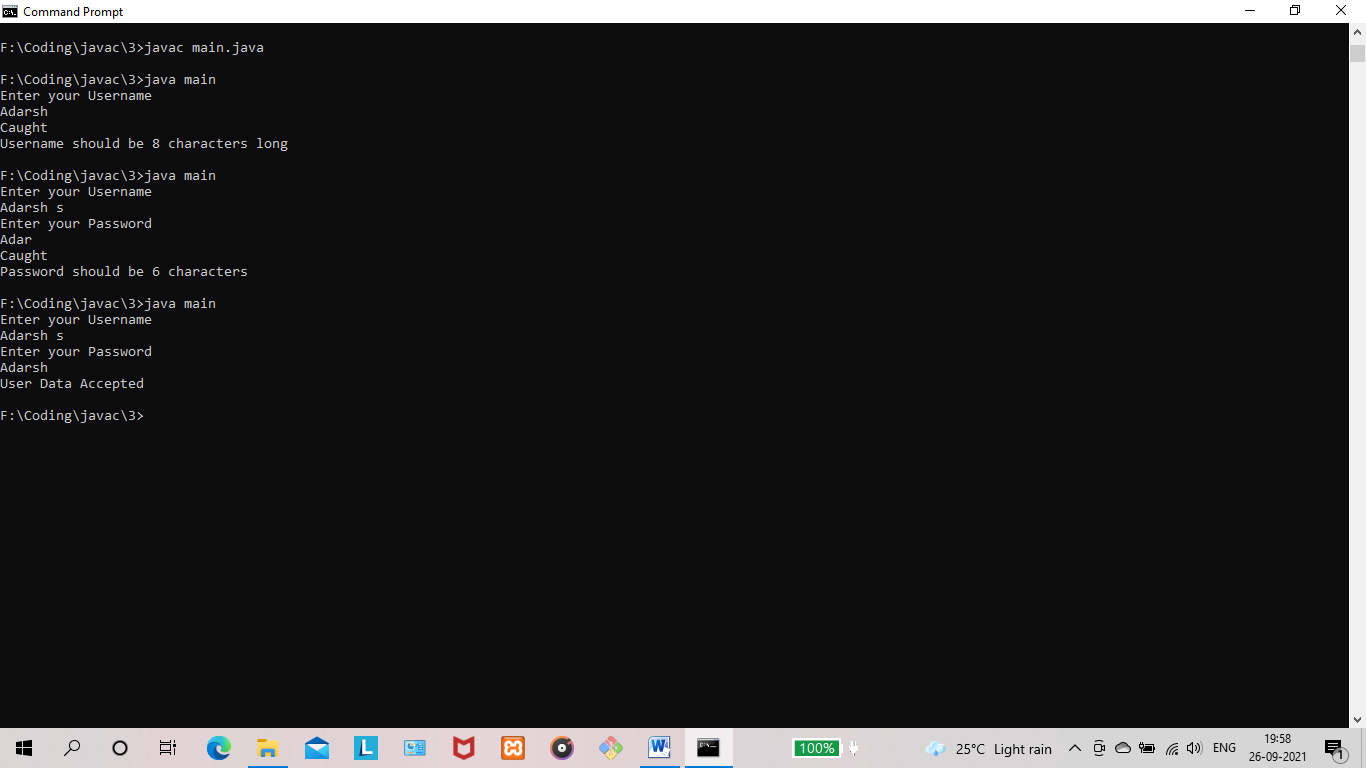
**PROGRAM NO : 19**

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

**PROGRAM CODE:**

|  |  |
| --- | --- |
| UserException.java | public class UserException extends Exception {  public UserException(String s)  {  // Call constructor of parent Exception  super(s);  }  } |
| Main.java | import java.util.Scanner;  public class Main  {  public static void main(String args[])  {  Scanner scanner = new Scanner(System.in);  String username,password;  try  {  System.out.println("Please enter a username : ");  username = scanner.nextLine();  if(username.equals("")){  throw new UserException("Username not provided !");  }  if(username.length()<8){  throw new UserException("The username should be atleast 8 characters long !");  }  System.out.println("Please enter a password : ");  password = scanner.nextLine();  if(password.equals("")){  throw new UserException("Password not provided !");  }  if(password.length()<6){  throw new UserException("The password should be atleast 6 characters long !");  }  System.out.println("User Data Accepted");  }  catch (UserException ex)  {  System.out.println("Caught");  System.out.println(ex.getMessage());  }  }  } |

**OUTPUT:**



**RESULT:** The program is successfully executed and the output is verified.

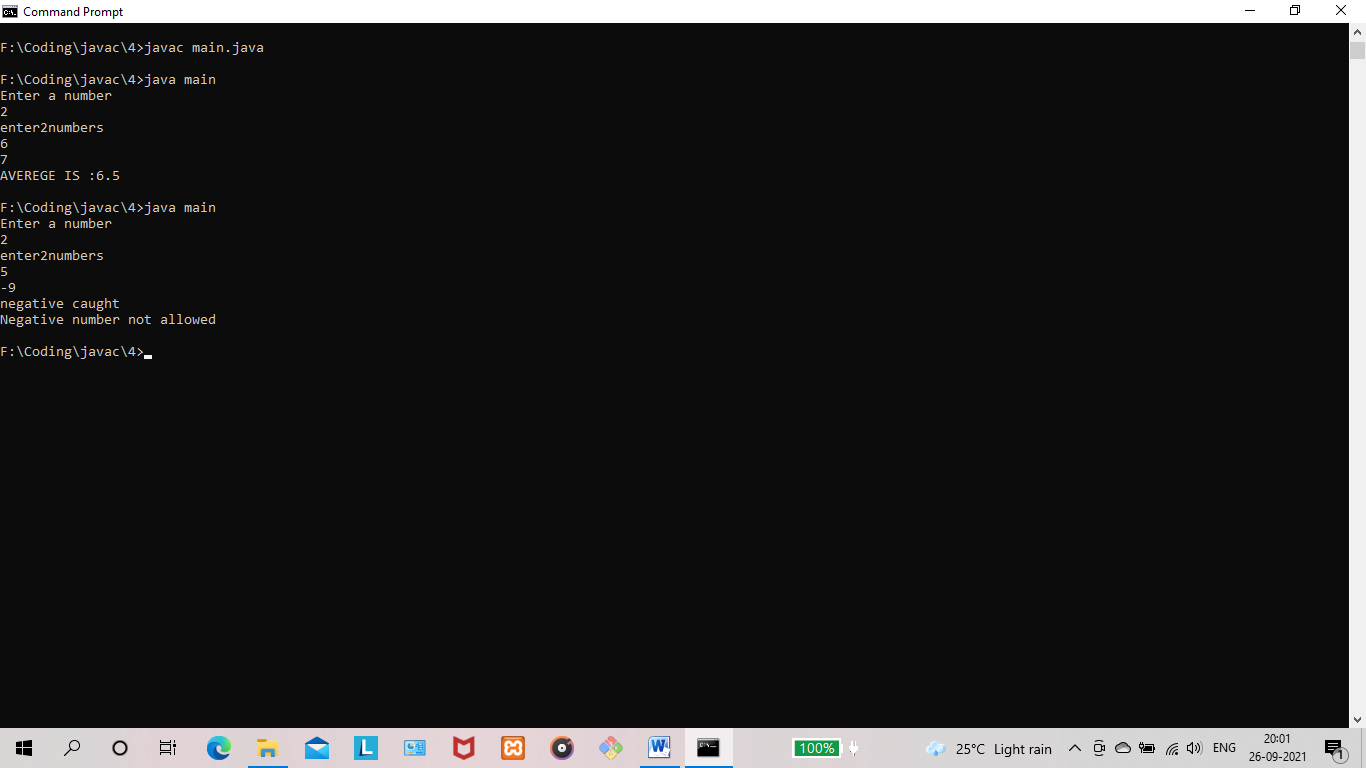
**PROGRAM NO : 20**

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

**PROGRAM CODE:**

|  |  |
| --- | --- |
| NegativeNumberException.java | public class NegativeNumberException extends Exception {  public NegativeNumberException(String s)  {  // Call constructor of parent Exception  super(s);  }  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String args[]) {  Scanner scanner = new Scanner(System.in);  int count;  System.out.println("Please enter the number of numbers to be calculated : ");  count = scanner.nextInt();  int[] numbers = new int[count];  try {  System.out.println("Please enter " + count + " numbers : ");  for (int i = 0; i < count; i++) {  int num = scanner.nextInt();  numbers[i] = num;  if (num < 0) {  throw new NegativeNumberException("Negative numbers are not allowed.");  }  }  int sum =0;  for (int i = 0; i < count; i++) {  sum += numbers[i];  }  float average = (float)sum/count;  System.out.println("Average of given numbers is : "+average);  } catch (NegativeNumberException ex) {  System.out.println("Exception Caught !!");  System.out.println(ex.getMessage());  }  }  } |

**OUTPUT:**



**RESULT:** The program is successfully executed and the output is verified.

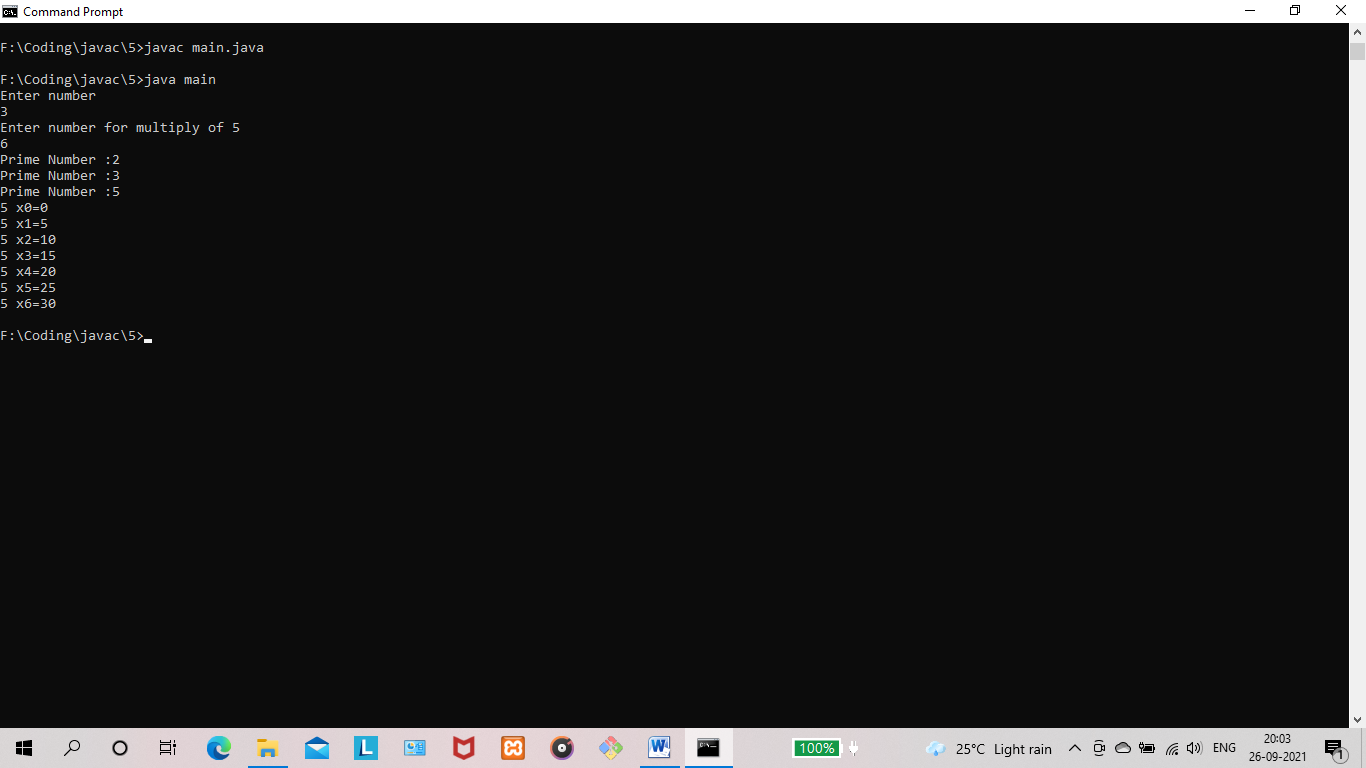
**PROGRAM NO : 21**

**AIM:** Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)

**PROGRAM CODE:**

|  |  |
| --- | --- |
| PrimeNumbers.java | import java.util.Scanner;  public class PrimeNumbers extends Thread {  int count;  int primeCount = 0;  PrimeNumbers(int count){  this.count=count;  }  public void run() {  for(int i=1;primeCount<count;i++){  boolean isPrime=true;  if(i==1)  {  i++;  }  for (int j=2;j<i;j++){  if(i%j==0){  isPrime = false;  break;  }  }  if (isPrime){  System.out.println("Prime Number : "+i);  primeCount++;  }  }  }  } |
| MultiTable5.java | public class MultiTable5 extends Thread {  int count;  MultiTable5(int count){  this.count=count;  }  public void run() {  for (int i=0;i<=count;i++){  System.out.println("5 x "+i+" = "+(5\*i));  }  }  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args)  {  Scanner scanner = new Scanner(System.in);  int primeCount,multipleCount;  System.out.println("Enter number of Prime Numbers to be listed : ");  primeCount = scanner.nextInt();  System.out.println("Enter number of Multiples of 5 to be listed : ");  multipleCount = scanner.nextInt();  PrimeNumbers thread1 = new PrimeNumbers(primeCount);  thread1.start();  MultiTable5 thread2 = new MultiTable5(multipleCount);  thread2.start();  }  } |

**OUTPUT:**



**RESULT:** The program is successfully executed and the output is verified.

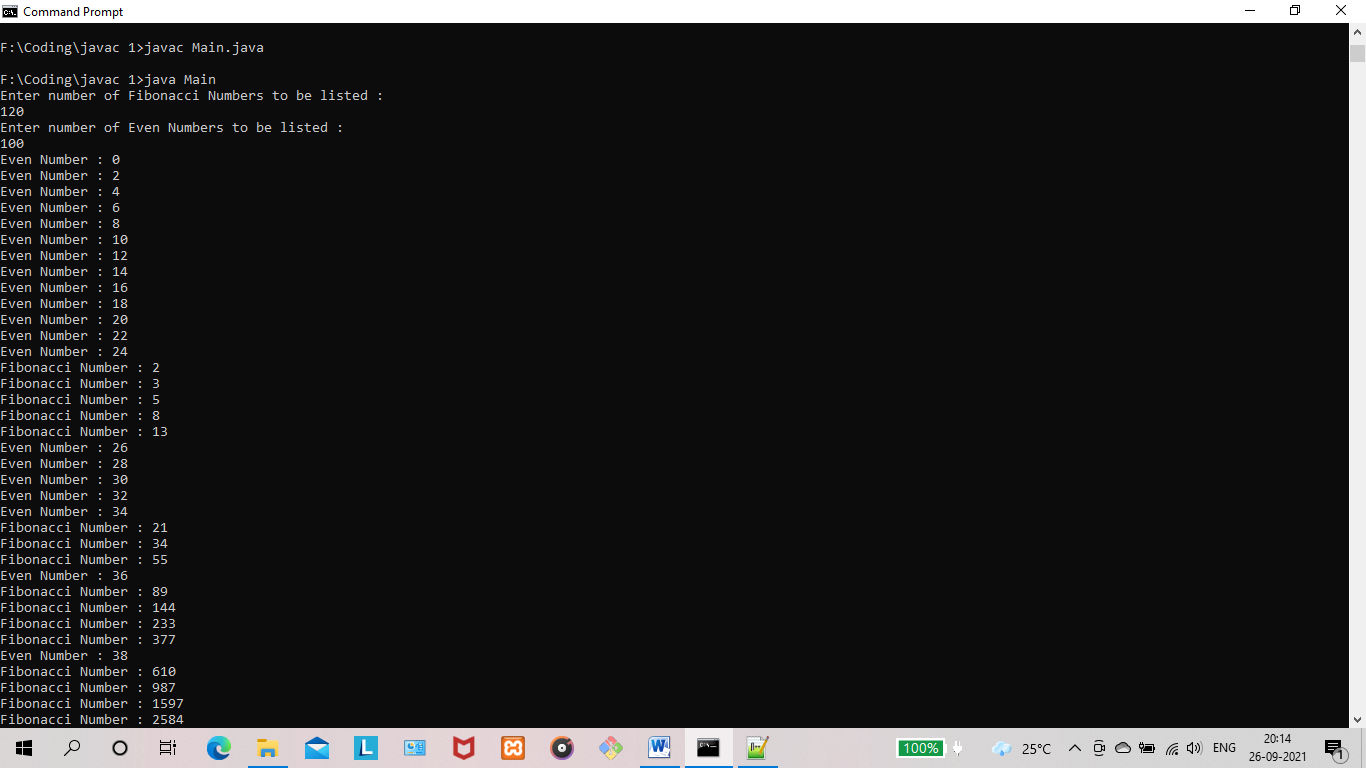
**PROGRAM NO : 22**

**AIM:** Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface).

**PROGRAM CODE:**

|  |  |
| --- | --- |
| EvenSeries.java | public class EvenSeries implements Runnable {  int count;  int[] fibonacciSeries;  EvenSeries(int count){this.count=count;}  public void run() {  for (int i=0;i<count;i++)  {  if(i%2==0)  {  System.out.println("Even Number : "+i);  }  }  }    } |
| FibonacciSeries.java | public class FibonacciSeries implements Runnable{  int count;  long[] fibonacciSeries;  FibonacciSeries(int count){this.count=count;this.fibonacciSeries=new long[count];}  public void run() {  this.fibonacciSeries[0]=0;  this.fibonacciSeries[1]=1;  this.fibonacciSeries[2]=1;  for (int i=3;i<count;i++)  {  this.fibonacciSeries[i]= this.fibonacciSeries[(i-1)]+ this.fibonacciSeries[(i-2)];  if(i>91) {  break;  }  System.out.println("Fibonacci Number : " + this.fibonacciSeries[i]);  }  }  } |
| Main.java | import java.util.Scanner;  public class Main {  public static void main(String[] args)  {  Scanner scanner = new Scanner(System.in);  int fiboCount,evenCount;  System.out.println("Enter number of Fibonacci Numbers to be listed : ");  fiboCount = scanner.nextInt();  System.out.println("Enter number of Even Numbers to be listed : ");  evenCount = scanner.nextInt();  FibonacciSeries fibonacciSeries = new FibonacciSeries(fiboCount);  EvenSeries evenSeries = new EvenSeries(evenCount);  Thread thread1 = new Thread(evenSeries);  Thread thread2 = new Thread(fibonacciSeries);  thread1.start();  thread2.start();  }  } |

**OUTPUT:**



**RESULT:** The program is successfully executed and the output is verified.

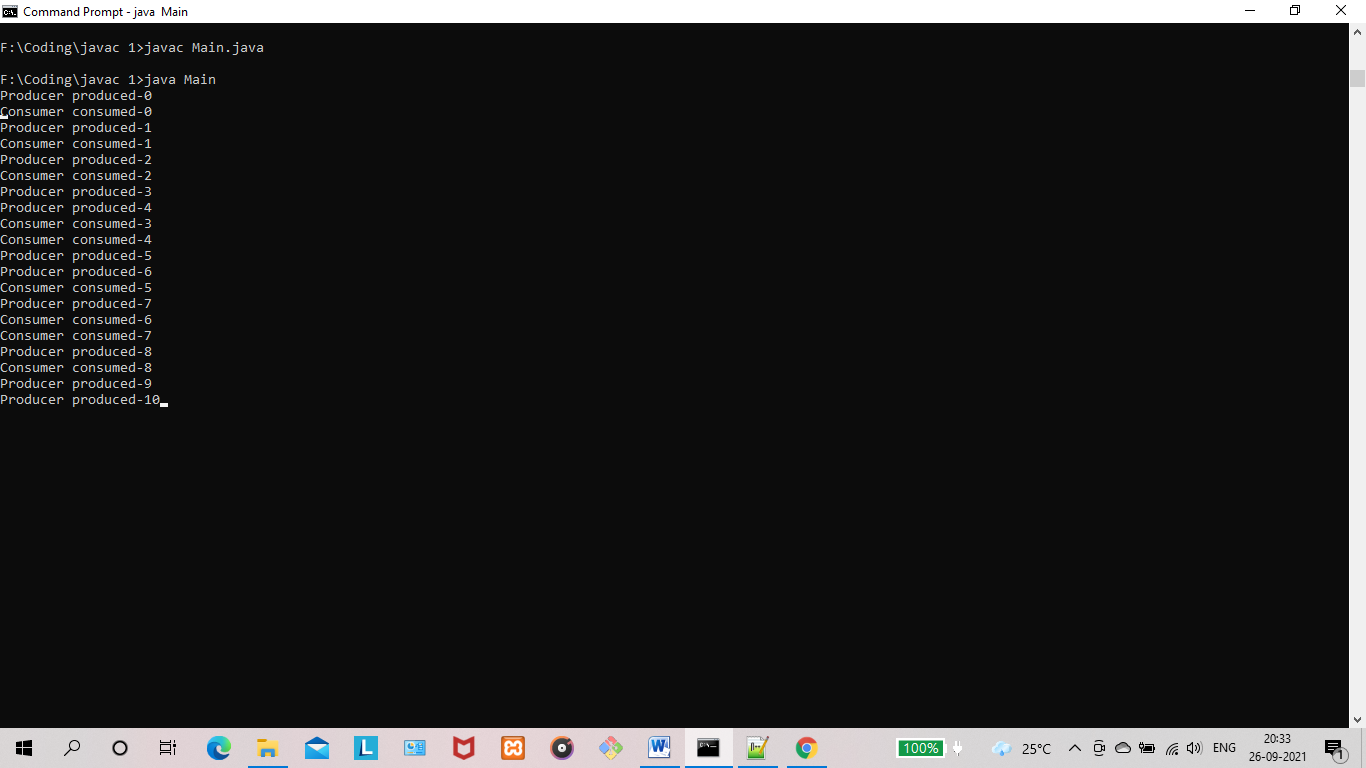
**PROGRAM NO : 23**

**AIM:** Producer/Consumer using ITC

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.LinkedList;  public class Main {  public static void main(String[] args) throws InterruptedException {  final PC pc = new PC();  Thread t1 = new Thread(new Runnable() {  @Override  public void run() {  try {  pc.produce();  } catch (InterruptedException e) {  e.printStackTrace();  }  }  });  Thread t2 = new Thread(new Runnable() {  @Override  public void run() {  try {  pc.consume();  } catch (InterruptedException e) {  e.printStackTrace();  }  }  });  t1.start();  t2.start();  t1.join();  t2.join();  }  public static class PC {  LinkedList<Integer> list = new LinkedList<>();  int capacity = 2;  public void produce() throws InterruptedException {  int value = 0;  while (true) {  synchronized (this) {  while (list.size() == capacity)  wait();  System.out.println("Producer produced-"  + value);  list.add(value++);  notify();  Thread.sleep(1000);  }  }  }  public void consume() throws InterruptedException {  while (true) {  synchronized (this) {  while (list.size() == 0)  wait();  int val = list.removeFirst();  System.out.println("Consumer consumed-"  + val);  notify();  Thread.sleep(1000);  }  }  }  }  } |

**OUTPUT :**



**RESULT:** The program is successfully executed and the output is verified.

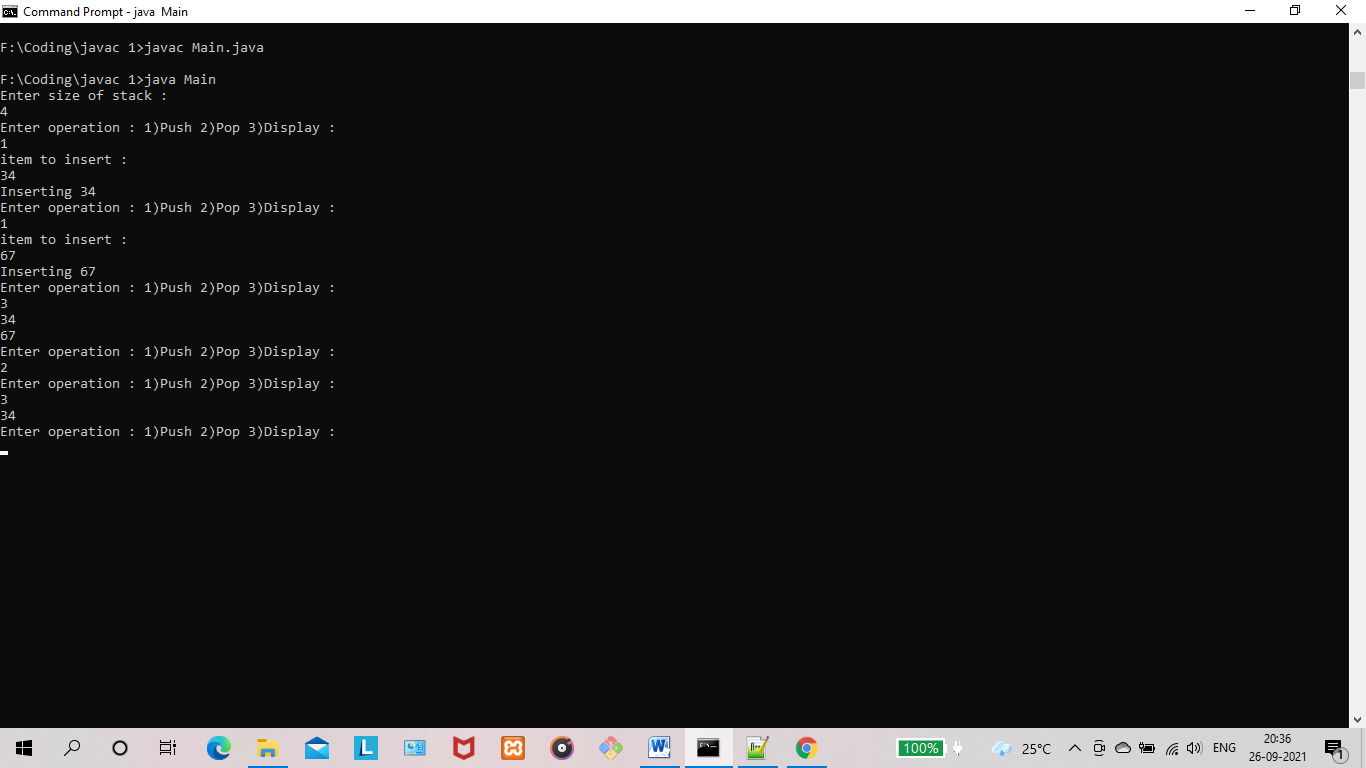
**PROGRAM NO : 24**

**AIM:** Program to create a generic stack and do the Push and Pop operations.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.Scanner;  class Main{  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  System.out.println("Enter size of stack : ");  int count = scanner.nextInt();  GenericStack stack = new GenericStack(count);  while (true) {  System.out.println("Enter operation : 1)Push 2)Pop 3)Display : ");  int choice = scanner.nextInt();  switch (choice) {  case 1:  System.out.println("item to insert :");  int item = scanner.nextInt();  stack.push(item);  break;  case 2:  stack.pop();  break;  case 3:  stack.printStack();  }  }  }  } |
| GenericStack.java | class GenericStack {  private int[] arr;  private int top;  private int capacity;  // Creating a stack  GenericStack(int size) {  arr = new int[size];  capacity = size;  top = -1;  }  // Add elements into stack  public void push(int x) {  if (isFull()) {  System.out.println("OverFlow");  }  else{  System.out.println("Inserting " + x);  arr[++top] = x;  }  }  // Remove element from stack  public int pop() {  if (isEmpty()) {  System.out.println("STACK EMPTY");  return -1;  }  else {  return arr[top--];  }  }  // Utility function to return the size of the stack  public int size() {  return top + 1;  }  // Check if the stack is empty  public Boolean isEmpty() {  return top == -1;  }  // Check if the stack is full  public Boolean isFull() {  return top == capacity - 1;  }  public void printStack() {  for (int i = 0; i <= top; i++) {  System.out.println(arr[i]);  }  }  } |

**OUTPUT :**



**RESULT:** The program is successfully executed and the output is verified.

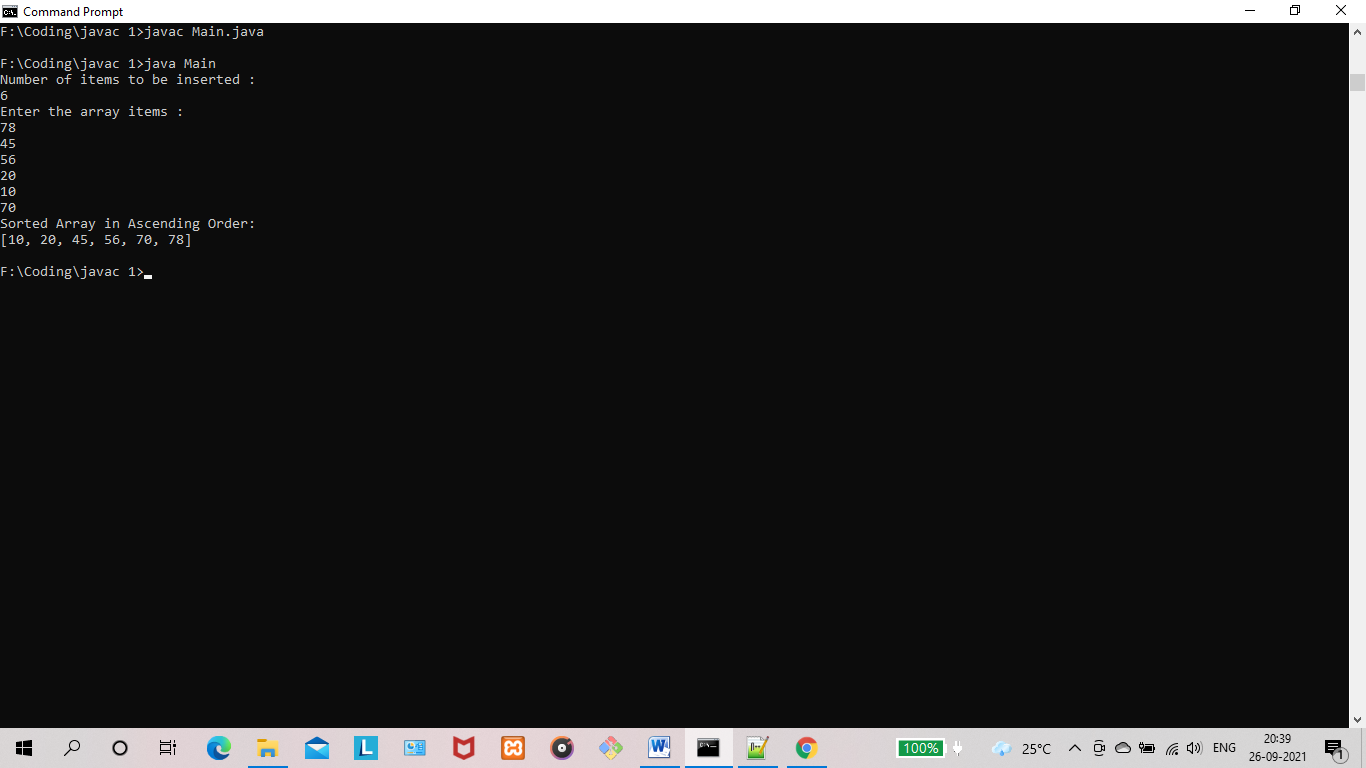
**PROGRAM NO : 25**

**AIM:** Using generic method perform Bubble sort.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.Arrays;  import java.util.Scanner;  public class Main {  static void bubbleSort(int array[]) {  int size = array.length;  for (int i = 0; i < size - 1; i++)  for (int j = 0; j < size - i - 1; j++)  if (array[j] > array[j + 1]) {  int temp = array[j];  array[j] = array[j + 1];  array[j + 1] = temp;  }  }  public static void main(String args[]) {  Scanner scanner = new Scanner(System.in);  System.out.println("Number of items to be inserted : ");  int count = scanner.nextInt();  int[] data = new int[count];  System.out.println("Enter the array items : ");  for(int i=0;i<count;i++)  {  data[i] = scanner.nextInt();  }  Main.bubbleSort(data);  System.out.println("Sorted Array in Ascending Order:");  System.out.println(Arrays.toString(data));  }  } |

**OUTPUT :**



**RESULT:** The program is successfully executed and the output is verified.

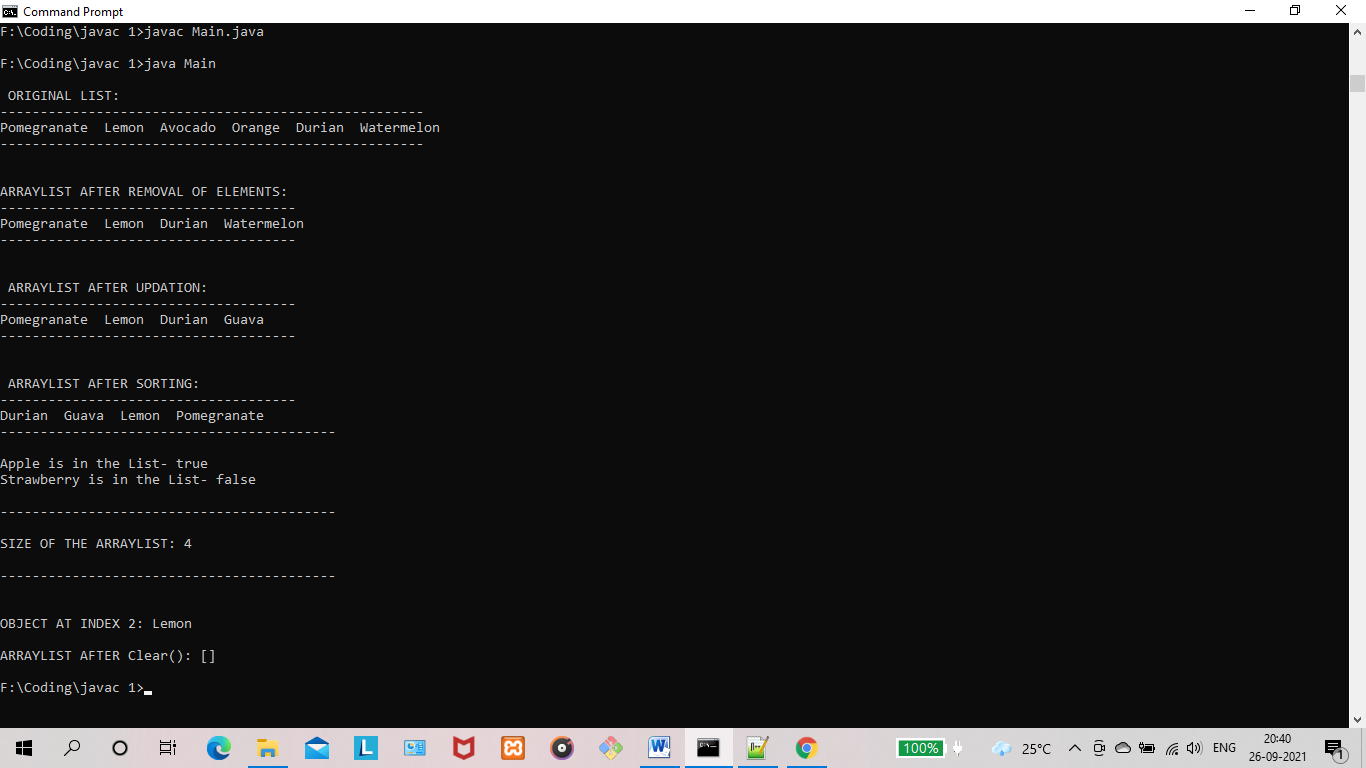
**PROGRAM NO : 26**

**AIM:** Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.\*;  public class Main {  public static void main(String[] args) {  // Creating ArrayList of type "String" which means we can only add "String" elements  ArrayList<String> fruits = new ArrayList<String>();  //adding elements to an ArrayList  fruits.add("Pomegranate");  fruits.add("Lemon");  fruits.add("Avocado");  fruits.add("Durian");  fruits.add("Watermelon");  fruits.add(3, "Orange");  // Displaying elements  System.out.println("\n ORIGINAL LIST:");  System.out.println("-----------------------------------------------------");  for(String str : fruits)  System.out.printf(str+" ");  //Remove elements from ArrayList  fruits.remove("Avocado");  fruits.remove(2);  // Displaying elements  System.out.println("\n-----------------------------------------------------");  System.out.println("\n\nARRAYLIST AFTER REMOVAL OF ELEMENTS:");  System.out.println("-------------------------------------");  for(String str : fruits )  System.out.printf(str+" ");  //Updating the ArrayList  fruits.set(3,"Guava");  System.out.println("\n-------------------------------------");  System.out.println("\n\n ARRAYLIST AFTER UPDATION:");  System.out.println("-------------------------------------");  for(String str : fruits )  System.out.printf(str+" ");  System.out.println("\n-------------------------------------");  //Sorting the ArrayList  Collections.sort(fruits);  System.out.println("\n\n ARRAYLIST AFTER SORTING:");  System.out.println("-------------------------------------");  for (String str : fruits)  System.out.printf(str+" ");  // Checks whether the object is in the ArrayList  System.out.println("\n------------------------------------------");  System.out.println("\nApple is in the List- "+ fruits.contains("Durian"));  System.out.println("Strawberry is in the List- "+fruits.contains("Strawberry"));  //Size of the ArrayList  System.out.println("\n------------------------------------------");  System.out.println("\nSIZE OF THE ARRAYLIST: "+ fruits.size());  //returns the object of list which is present at the specified index  System.out.println("\n------------------------------------------");  System.out.println("\n\nOBJECT AT INDEX 2: "+ fruits.get(2));  // removing all the elements of the ArrayList  fruits.clear();  System.out.println("\nARRAYLIST AFTER Clear(): "+ fruits);  }  } |

**OUTPUT :**



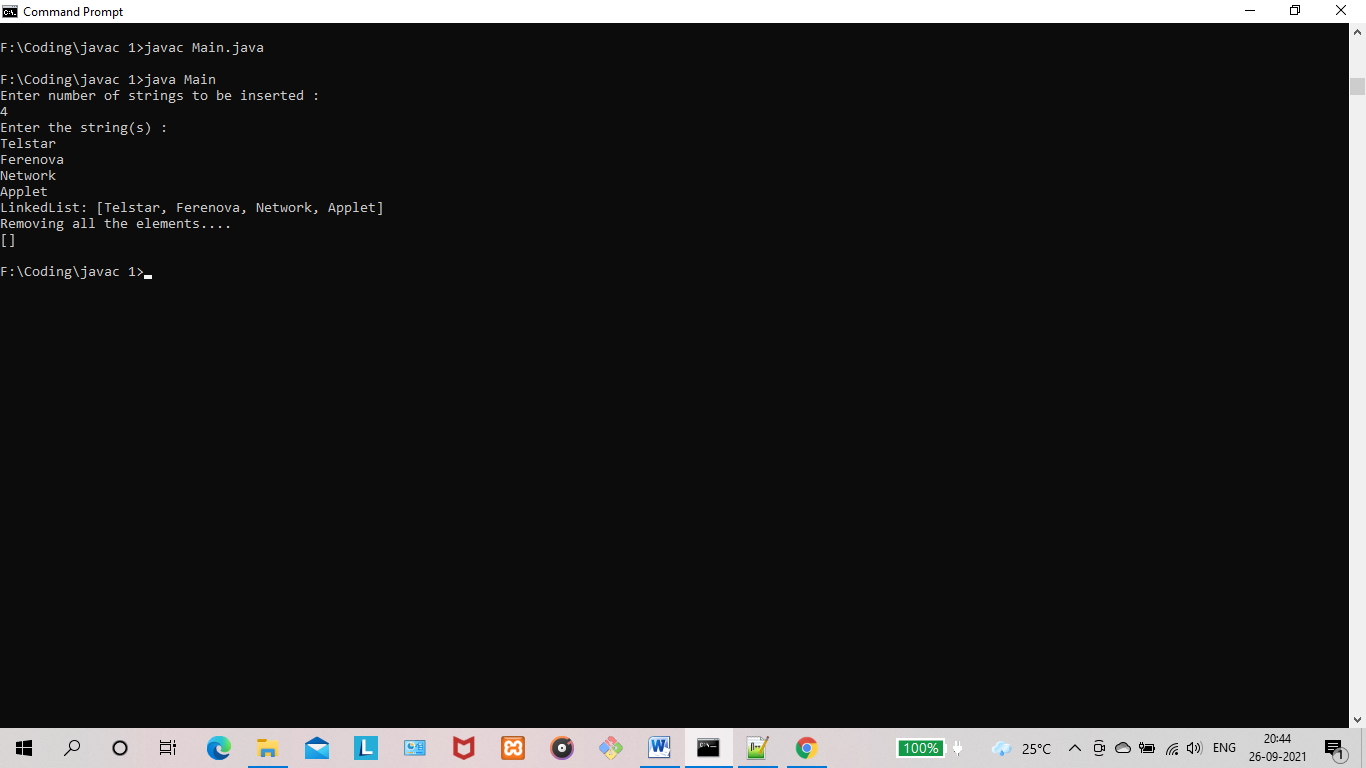
**RESULT:** The program is successfully executed and the output is verified.

**PROGRAM NO : 27**

**AIM:** Program to remove all the elements from a linked list

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.\*;  public class Main {  public static void main(String[] args) {  int n;  String data;  LinkedList<String> linkedList = new LinkedList<String>();  System.out.println("Enter number of strings to be inserted : ");  Scanner scanner = new Scanner(System.in);  n = scanner.nextInt();  System.out.println("Enter the string(s) : ");  scanner.nextLine();  for (int i = 0; i < n; i++) {  data = scanner.nextLine();  linkedList.add(data);  }  System.out.println("LinkedList: " + linkedList);  System.out.println("Removing all the elements....");  linkedList.clear();  System.out.println(linkedList);  }  } |

**OUTPUT :**

**RESULT:** The program is successfully executed and the output is verified.

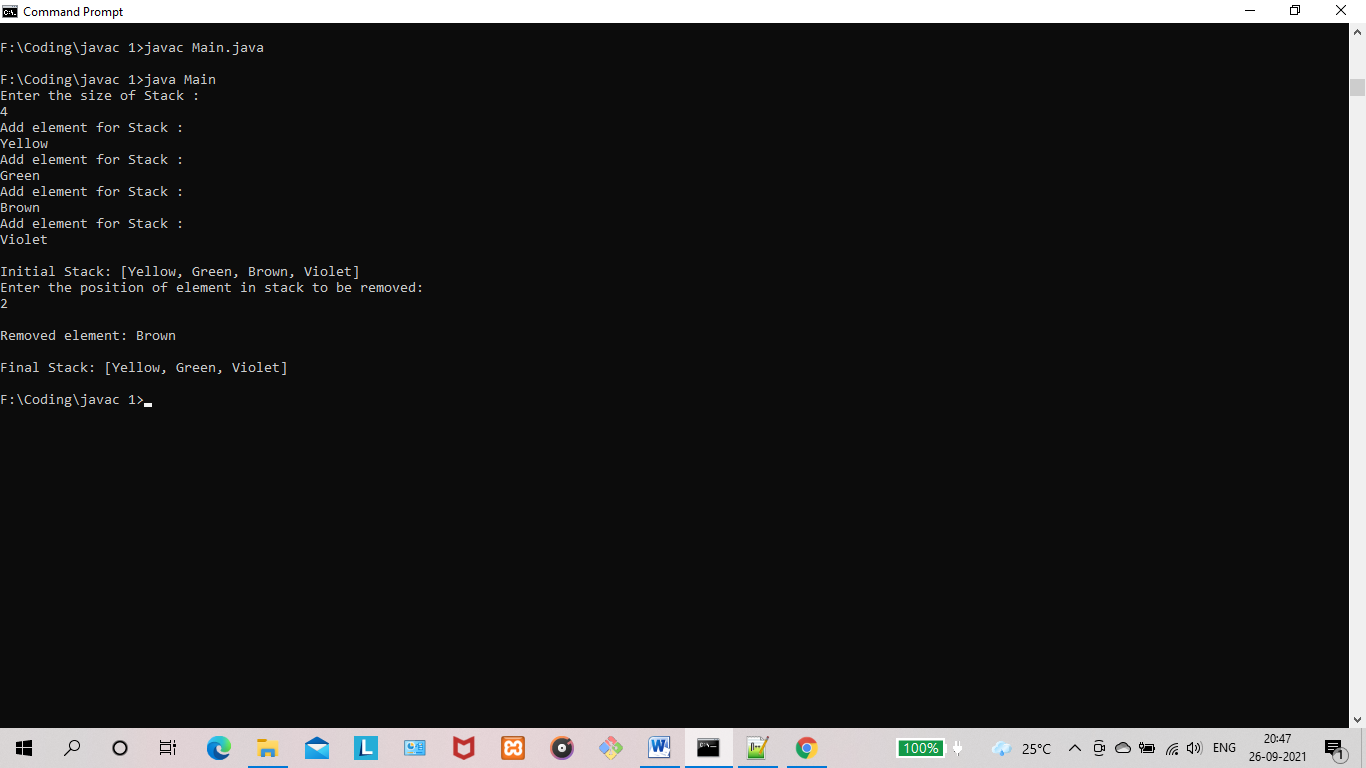
**PROGRAM NO : 28**

**AIM:** Program to remove an object from the Stack when the position is passed as parameter.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.Scanner;  import java.util.Stack;  public class Main {  public static void main(String[] args) {  Stack<String> stack = new Stack<String>();  Scanner scanner=new Scanner(System.in);  System.out.println("Enter the size of Stack : ");  int num=scanner.nextInt();  for(int i =0;i<num;i++)  {  System.out.println("Add element for Stack : ");  String str=scanner.next();  stack.add(str);  }  System.out.println();  System.out.println("Initial Stack: " + stack);  System.out.println("Enter the position of element in stack to be removed: ");  int pos=scanner.nextInt();  String rem = stack.remove(pos);  System.out.println("\nRemoved element: "+ rem);  System.out.println("\nFinal Stack: " + stack);  }  } |

**OUTPUT :**



**RESULT:** The program is successfully executed and the output is verified.

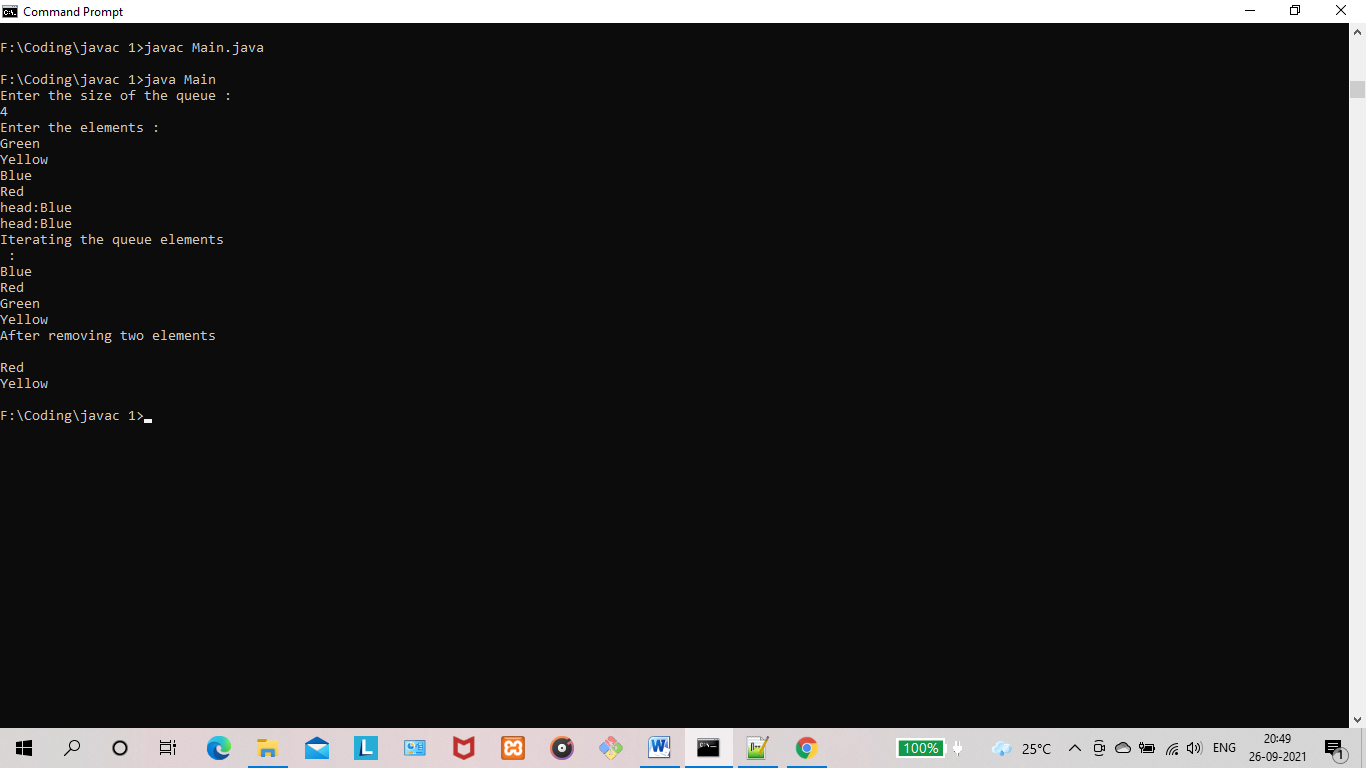
**PROGRAM NO : 29**

**AIM:** Program to demonstrate the creation of queue object using the PriorityQueue Class

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.PriorityQueue;  import java.util.Scanner;  public class Main {  public static void main(String[] args)  {  PriorityQueue<String> queue=new PriorityQueue<String>();  Scanner scanner=new Scanner(System.in);  System.out.println("Enter the size of the queue : ");  int n=scanner.nextInt();  System.out.println("Enter the elements : ");  for(int i =0;i<n;i++)  {  String st=scanner.next();  queue.add(st);  }  System.out.println("head:"+queue.element());  System.out.println("head:"+queue.peek());  System.out.println("Iterating the queue elements\n : ");  for (String value : queue) {  System.out.println(value);  }  queue.remove();  queue.poll();  System.out.println("After removing two elements \n");  for (String s : queue) {  System.out.println(s);  }  }  } |

**OUTPUT :**



**RESULT:** The program is successfully executed and the output is verified.

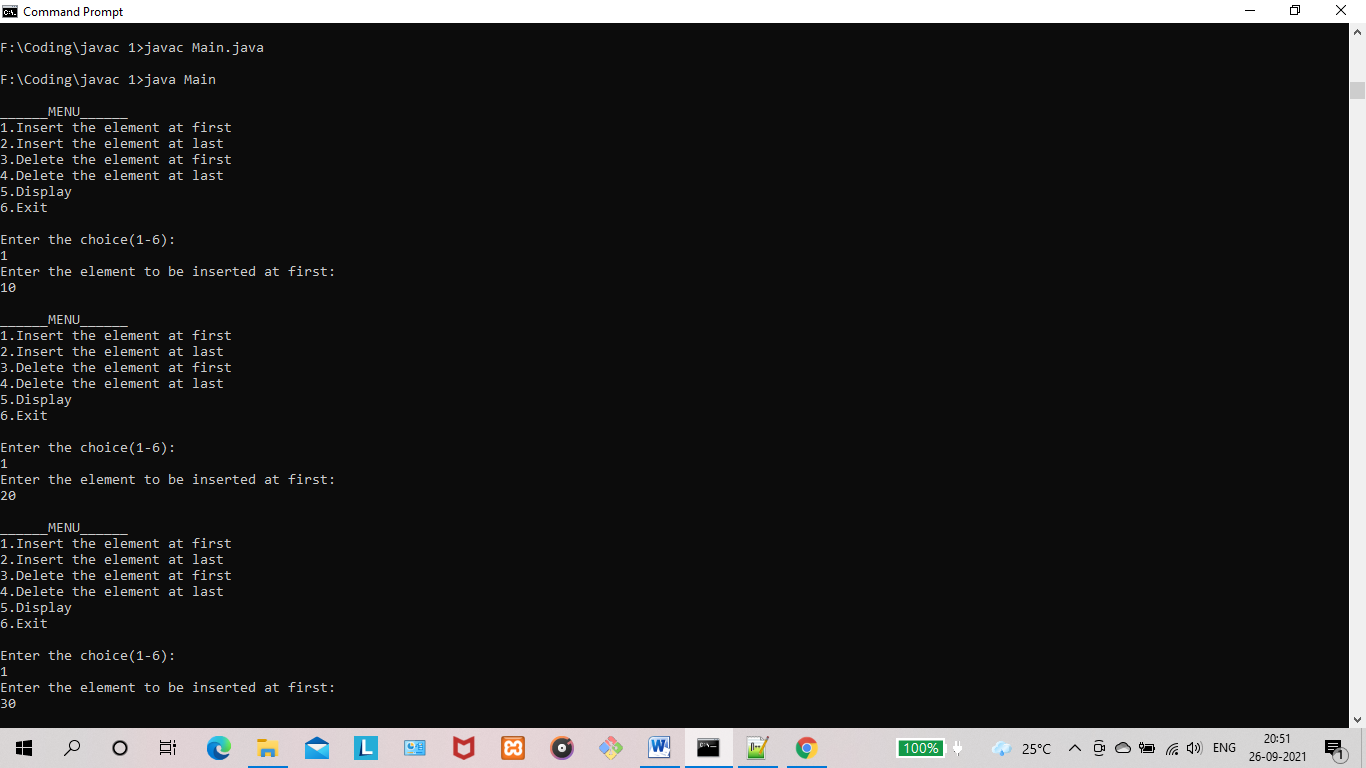
**PROGRAM NO : 30**

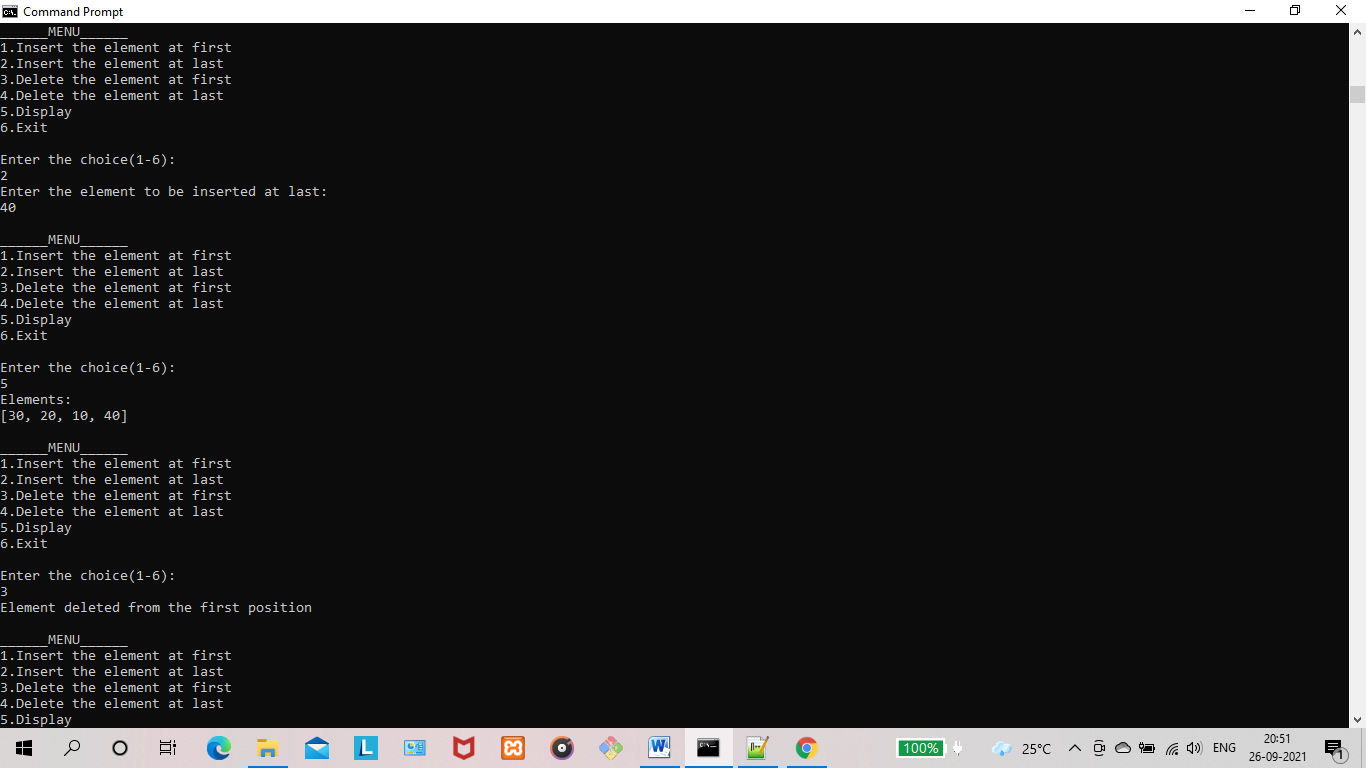
**AIM:** Program to demonstrate the addition and deletion of elements in deque

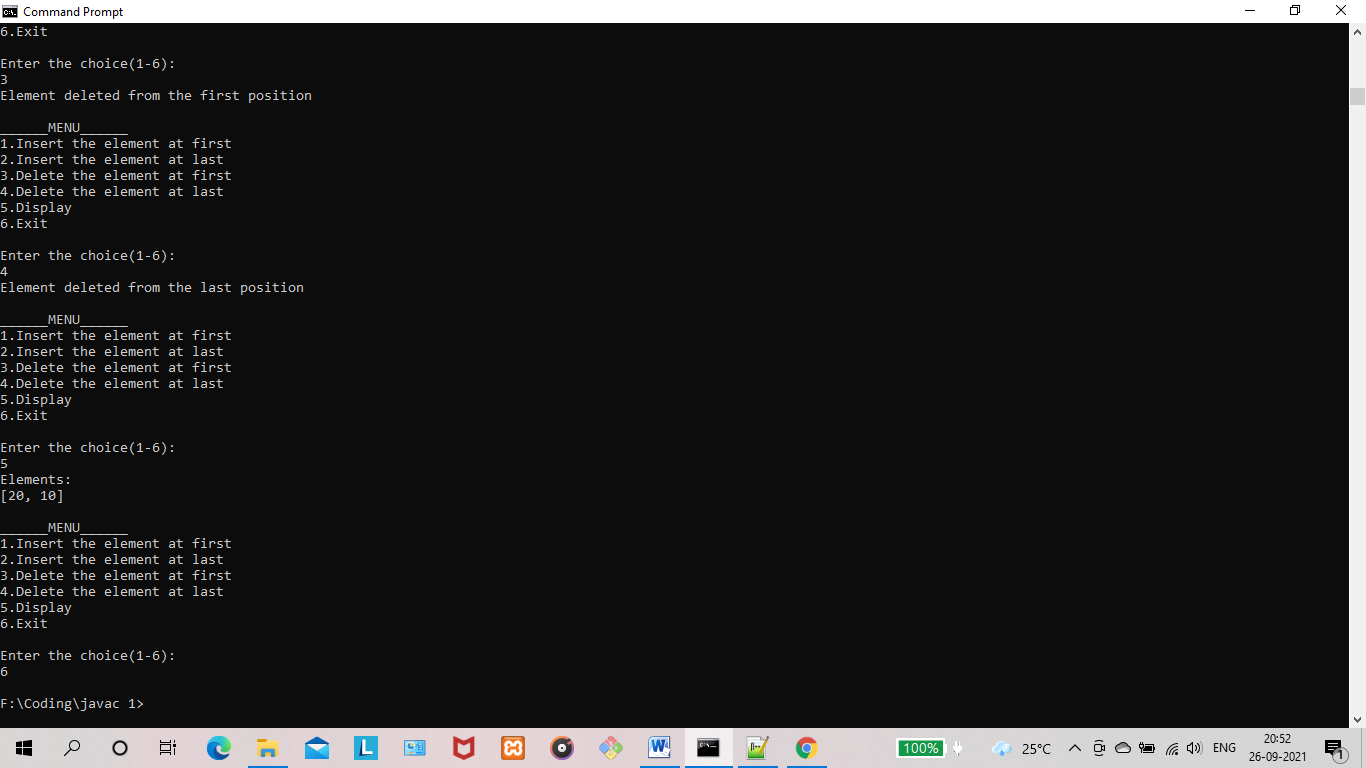
**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.\*;  public class Main {  public static void main(String[] args) {  int ch, data;  Deque<Integer> dq = new LinkedList<Integer>();  Scanner sc = new Scanner(System.in);  do  {  System.out.println("\n\_\_\_\_\_\_MENU\_\_\_\_\_\_");  System.out.println("1.Insert the element at first");  System.out.println("2.Insert the element at last");  System.out.println("3.Delete the element at first");  System.out.println("4.Delete the element at last");  System.out.println("5.Display");  System.out.println("6.Exit");  System.out.println("\nEnter the choice(1-6):");  ch = sc.nextInt();  sc.nextLine();  switch(ch)  {  case 1: System.out.println("Enter the element to be inserted at first:");  data = sc.nextInt();  dq.addFirst(data);  break;  case 2: System.out.println("Enter the element to be inserted at last:");  data = sc.nextInt();  dq.addLast(data);  break;  case 3: System.out.println("Element deleted from the first position");  dq.removeFirst();  break;  case 4: System.out.println("Element deleted from the last position");  dq.removeLast();  break;  case 5: System.out.println("Elements:");  System.out.println(dq);  break;  case 6: System.exit(0);  break;  default:System.out.println("Invalid choice...");  }  }while(true);  }  } |

**OUTPUT :**







**RESULT:** The program is successfully executed and the output is verified.

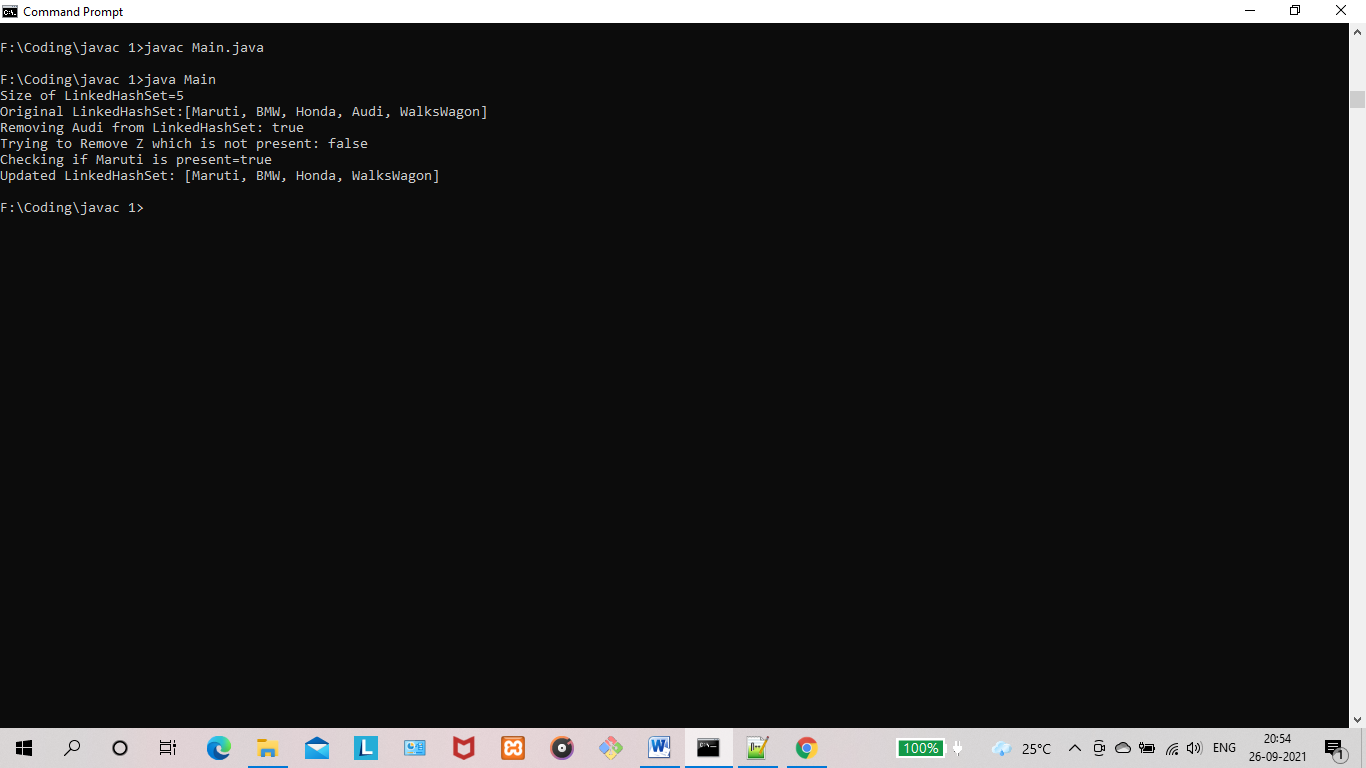
**PROGRAM NO : 31**

**AIM:** Program to demonstrate the creation of Set object using the LinkedHashset Class.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.LinkedHashSet;  public class Main {  public static void main(String[] args) {  LinkedHashSet<String> linkedset = new LinkedHashSet<String>();  // Adding element to LinkedHashSet  linkedset.add("Maruti");  linkedset.add("BMW");  linkedset.add("Honda");  linkedset.add("Audi");  linkedset.add("Maruti"); //This will not add new element as Maruti already exists  linkedset.add("WalksWagon");  System.out.println("Size of LinkedHashSet=" + linkedset.size());  System.out.println("Original LinkedHashSet:" + linkedset);  System.out.println("Removing Audi from LinkedHashSet: " + linkedset.remove("Audi"));  System.out.println("Trying to Remove Z which is not present: "  + linkedset.remove("Z"));  System.out.println("Checking if Maruti is present=" + linkedset.contains("Maruti"));  System.out.println("Updated LinkedHashSet: " + linkedset);  }  } |

**OUTPUT :**



**RESULT:** The program is successfully executed and the output is verified.

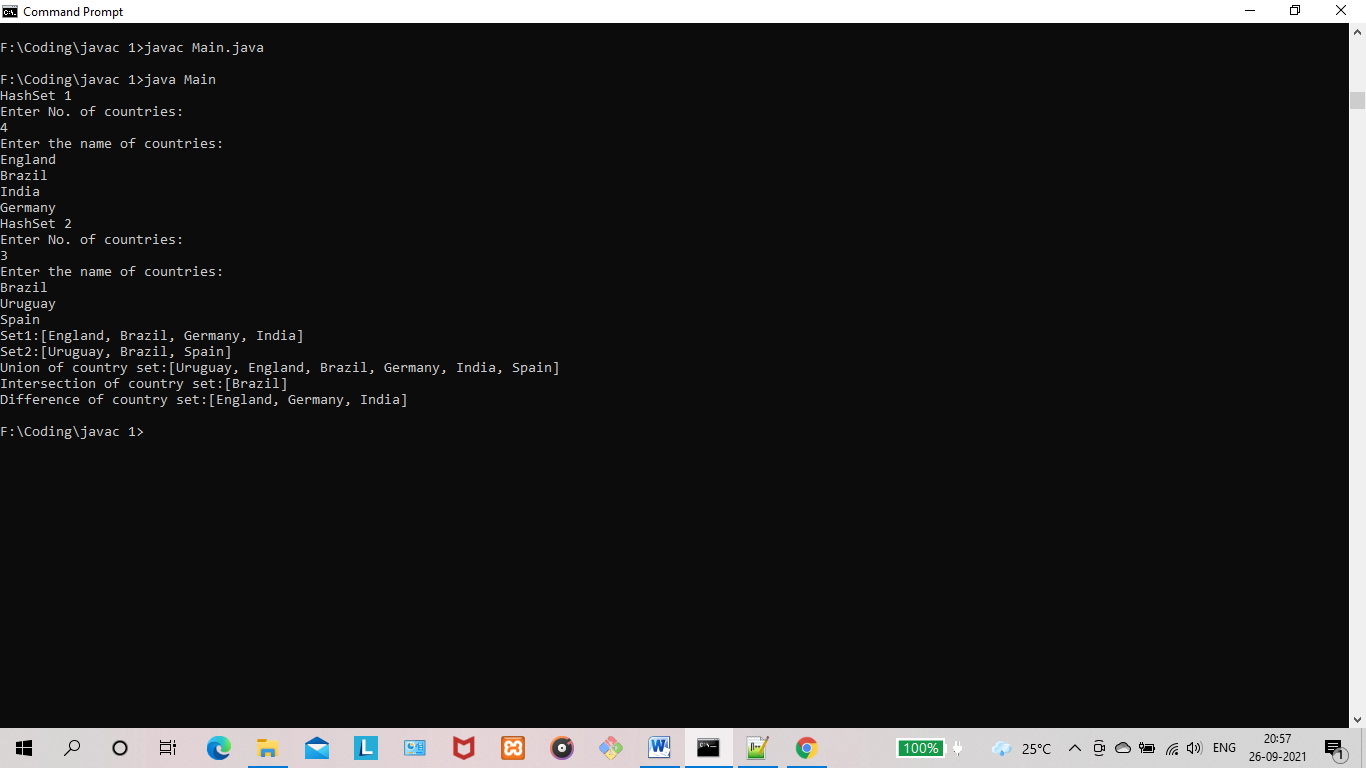
**PROGRAM NO : 32**

**AIM:** Write a Java program to compare two hash sets.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.\*;  public class Main {  public static void main(String[] args) {  int n;  String str;  HashSet<String> set1= new HashSet<String>();  System.out.println("HashSet 1");  System.out.println("Enter No. of countries:");  Scanner sc=new Scanner(System.in);  n=sc.nextInt();  System.out.println("Enter the name of countries:");  Scanner sc1=new Scanner(System.in);  for(int i=0;i<n;i++) {  str=sc1.nextLine();  set1.add(str);  }  System.out.println("HashSet 2");  HashSet<String> set2= new HashSet<String>();  System.out.println("Enter No. of countries:");  n=sc.nextInt();  System.out.println("Enter the name of countries:");  for(int i=0;i<n;i++) {  str=sc1.nextLine();  set2.add(str);  }  System.out.println("Set1:"+set1);  System.out.println("Set2:"+set2);  HashSet<String> a= new HashSet<String>(set1);  a.addAll(set2);  System.out.println("Union of country set:"+a);  HashSet<String> b= new HashSet<String>(set1);  b.retainAll(set2);  System.out.println("Intersection of country set:"+b);  HashSet<String> c= new HashSet<String>(set1);  c.removeAll(set2);  System.out.println("Difference of country set:"+c);  }  } |

**OUTPUT :**



**RESULT:** The program is successfully executed and the output is verified.

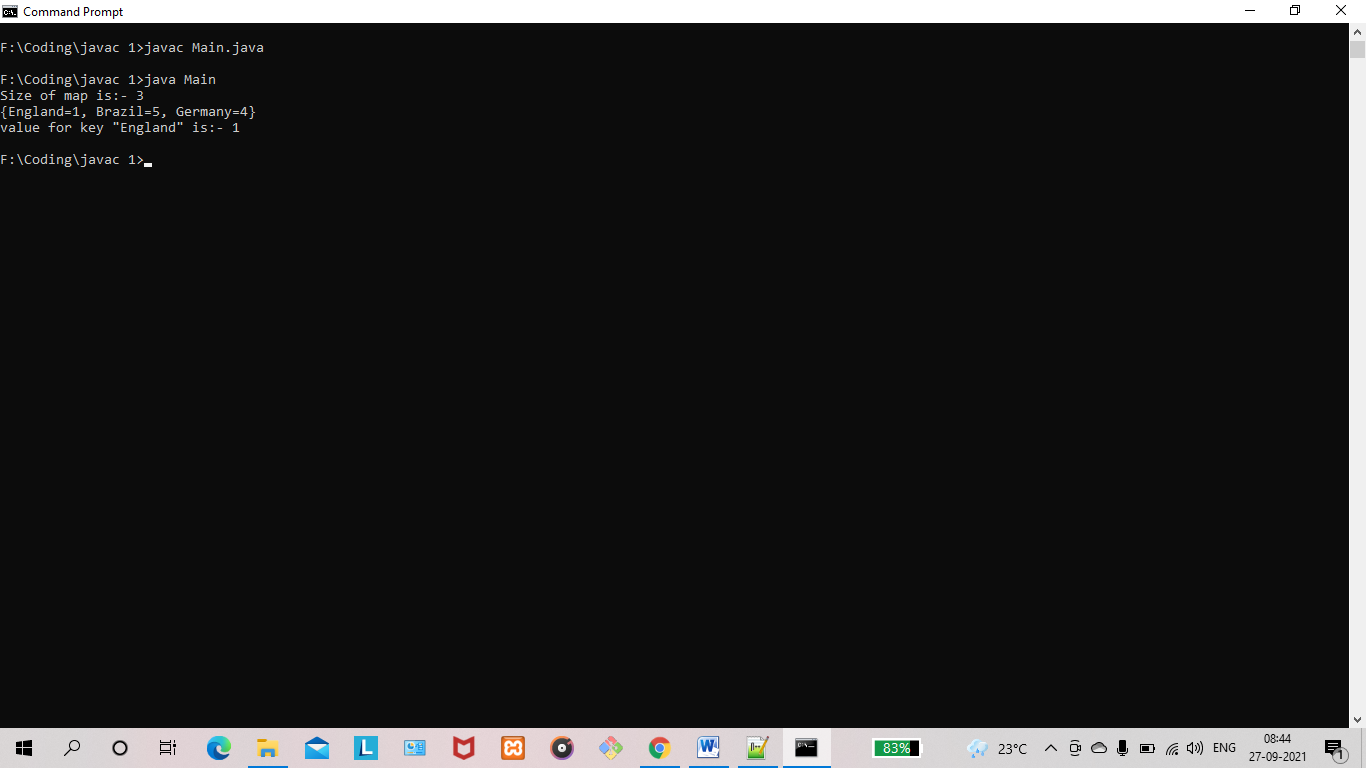
**PROGRAM NO : 33**

**AIM:** Program to demonstrate the working of Map interface by adding, changing and removing elements.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | // Java program to illustrate HashMap class of java.util  // package  // Importing HashMap class  import java.util.HashMap;  // Main class  public class Main {  // Main driver method  public static void main(String[] args)  {  // Create an empty hash map by declaring object  // of string and integer type  HashMap<String, Integer> map = new HashMap<>();  // Adding elements to the Map  // using standard add() method  map.put("Germany", 4);  map.put("England", 1);  map.put("Brazil", 5);  // Print size and content of the Map  System.out.println("Size of map is:- "  + map.size());  // Printing elements in object of Map  System.out.println(map);  // Checking if a key is present and if  // present, print value by passing  // random element  if (map.containsKey("England")) {  // Mapping  Integer a = map.get("England");  // Printing value fr the corresponding key  System.out.println("value for key"  + " \"England\" is:- " + a);  }  }  } |

**OUTPUT :**



**RESULT:** The program is successfully executed and the output is verified.

**PROGRAM NO : 34**

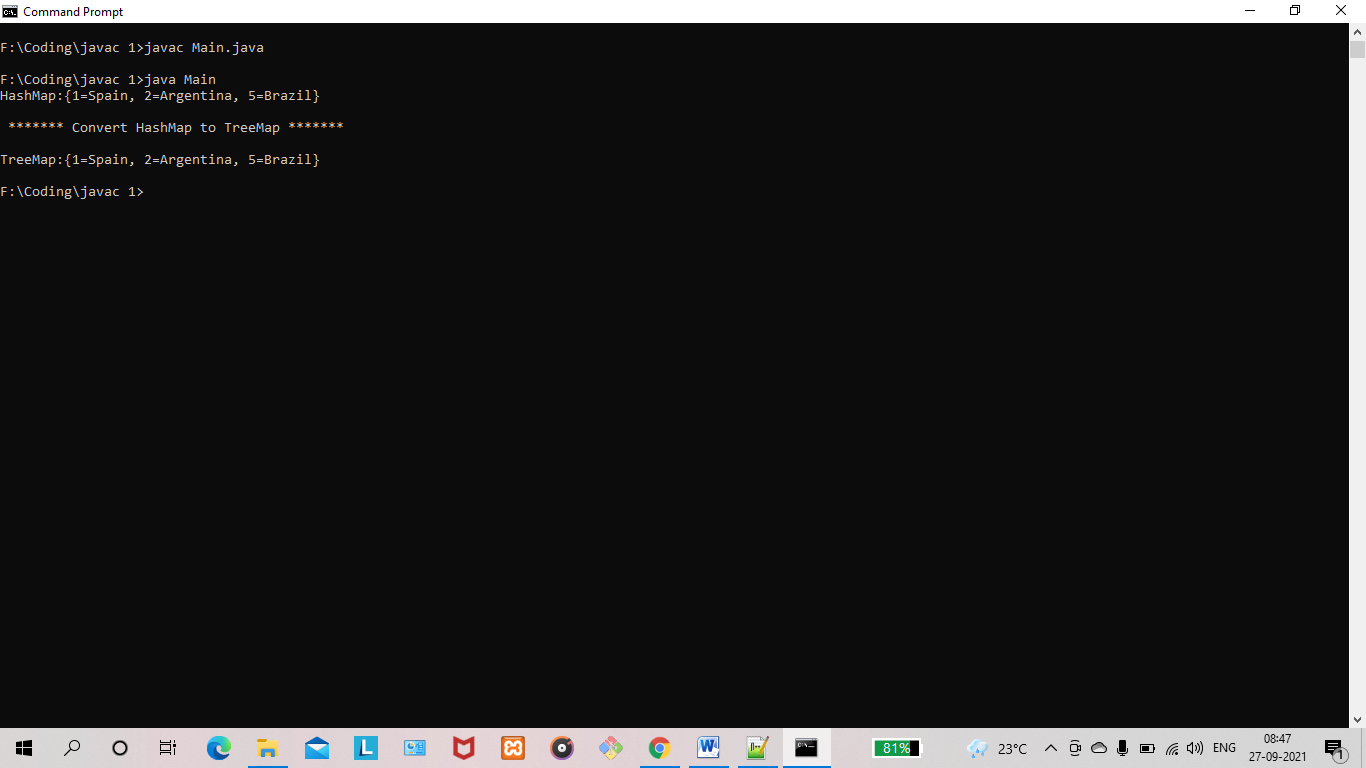
**AIM:** Program to Convert HashMap to TreeMap.

.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.util.\*;  public class Main {  public static void main(String[] args) {  // TODO Auto-generated method stub  Map<Integer,String> hm=new LinkedHashMap<>();  hm.put(1,"England");  hm.put(1,"Spain");  hm.put(2,"France");  hm.put(5,"Brazil");  hm.put(2,"Argentina");  System.out.println("HashMap:"+hm);  Map<Integer,String> tm=new TreeMap<>(hm);  System.out.println("\n \*\*\*\*\*\*\* Convert HashMap to TreeMap \*\*\*\*\*\*\*\n");  System.out.println("TreeMap:"+tm);  }  } |

**OUTPUT :**



**RESULT:** The program is successfully executed and the output is verified.

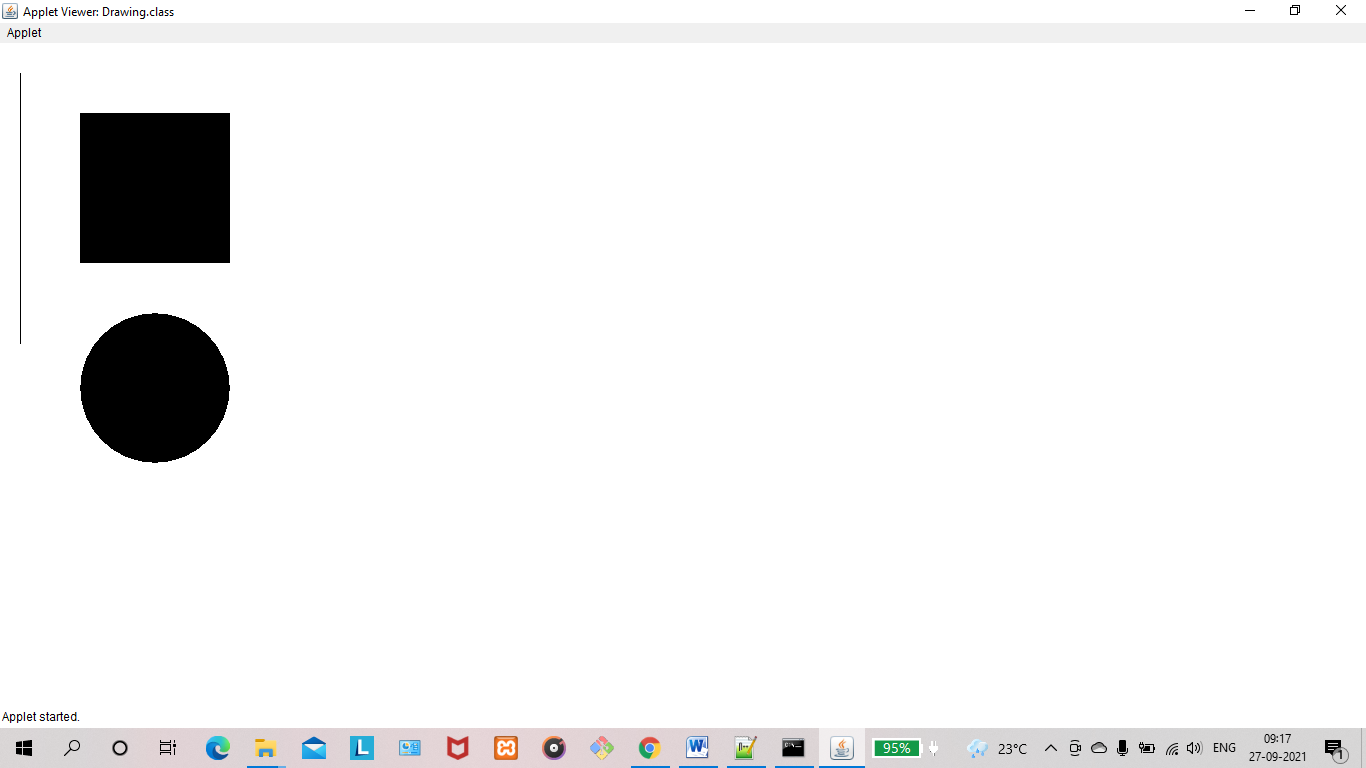
**PROGRAM NO : 35**

**AIM:** Program to draw Circle, Rectangle, Line in Applet.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Drawing.java | import java.applet.Applet;  import java.awt.\*;  class Drawing extends Applet{  public void paint(Graphics g){  g.drawLine(20,30,20,300);  g.fillRect(80,70,150,150);  g.fillOval(80,270,150,150);  }  } |
| Index.html | <html>  <body>  <applet code="Drawing.class" width="300" height="300">  </applet>  </body>  </html> |

**OUTPUT:**



**RESULT:** The program is successfully executed and the output is verified.

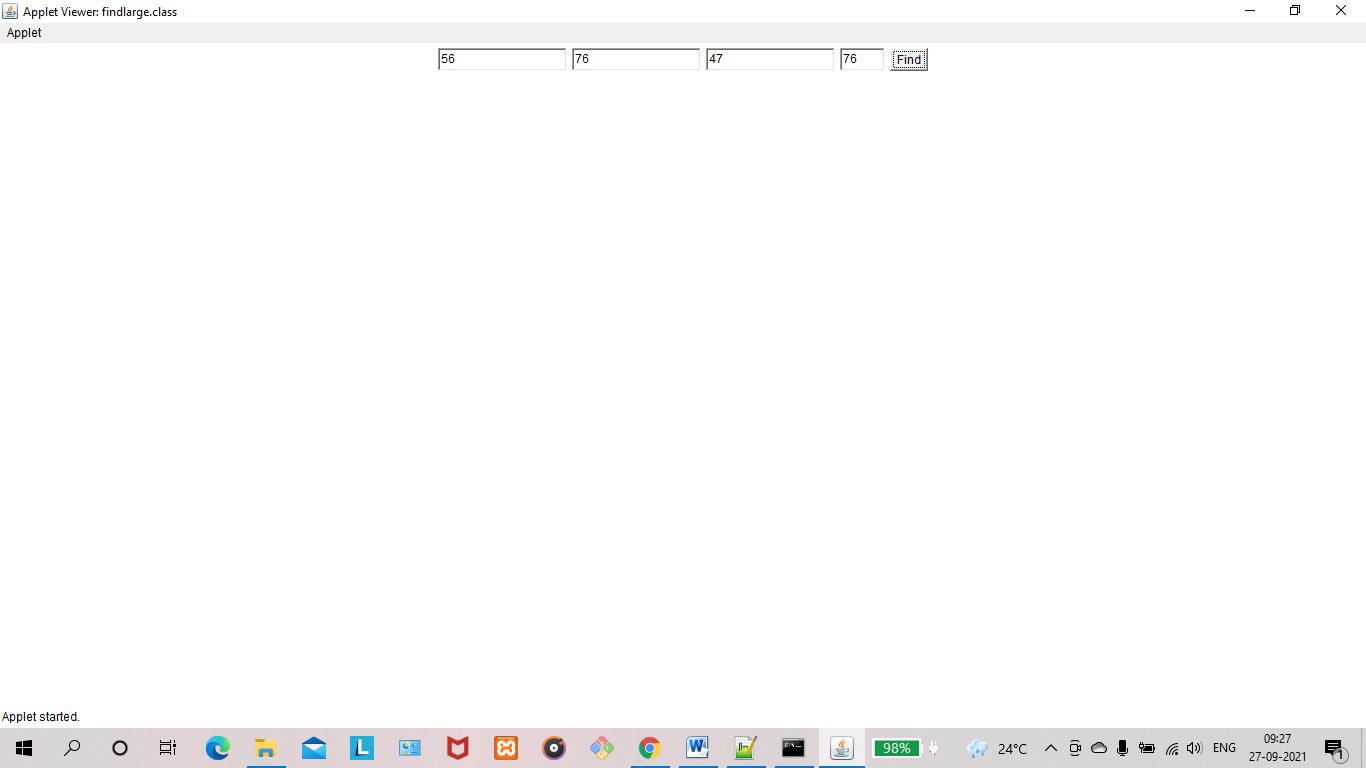
**PROGRAM NO : 36**

**AIM:** To write a program to find maximum of three numbers using AWT.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.awt.\*;  import java.applet.\*;  import java.awt.event.\*;  public class findlarge extends Applet implements ActionListener  {  TextField t1,t2,t3,t4;  Button b1;  public void init()  {  t1=new TextField(15);  t1.setBounds(100,25,50,20);  t2=new TextField(15);  t2.setBounds(100,25,50,20);  t3=new TextField(15);  t3.setBounds(100,25,50,20);  t4=new TextField("Ans");  t4.setBounds(175,50,50,20);  b1=new Button("Find");  b1.setBounds(175,65,50,40);  add(t1);  add(t2);  add(t3);  add(t4);  add(b1);  b1.addActionListener(this);  }  public void actionPerformed(ActionEvent e)  {  int i,j,k;  i=Integer.parseInt(t1.getText());  j=Integer.parseInt(t2.getText());  k=Integer.parseInt(t3.getText());  if(i<j)  {  if(j<k)  t4.setText(""+k);  else  t4.setText(""+j);  }  else  t4.setText(""+i);  }  } |
| index.html | <html>  <head>  </head>  <body>  <div align="center">  <applet code="findlarge.class"width="800"height="500">  </applet>  </div>  </body>  </html> |

**OUTPUT:**



**RESULT:** The program is successfully executed and the output is verified.

## PROGRAM NO: 37

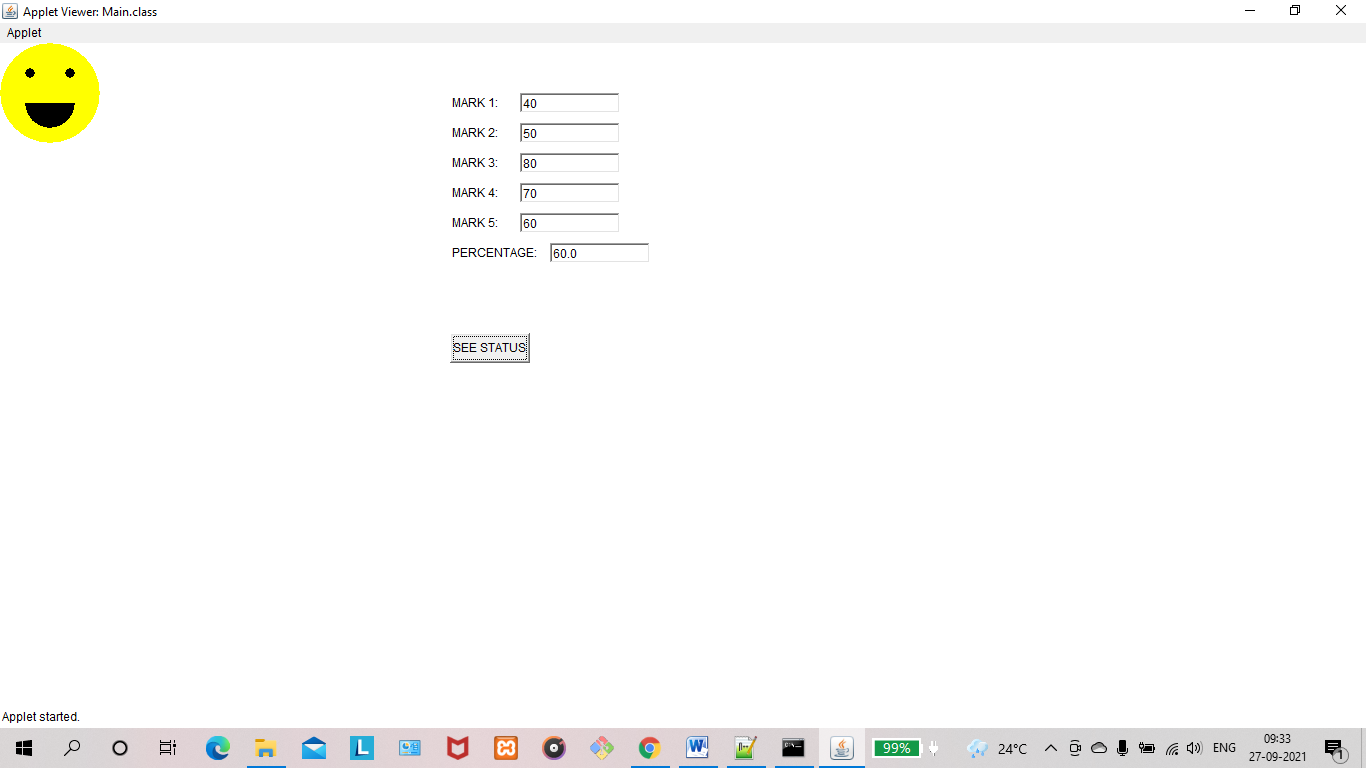
### AIM:

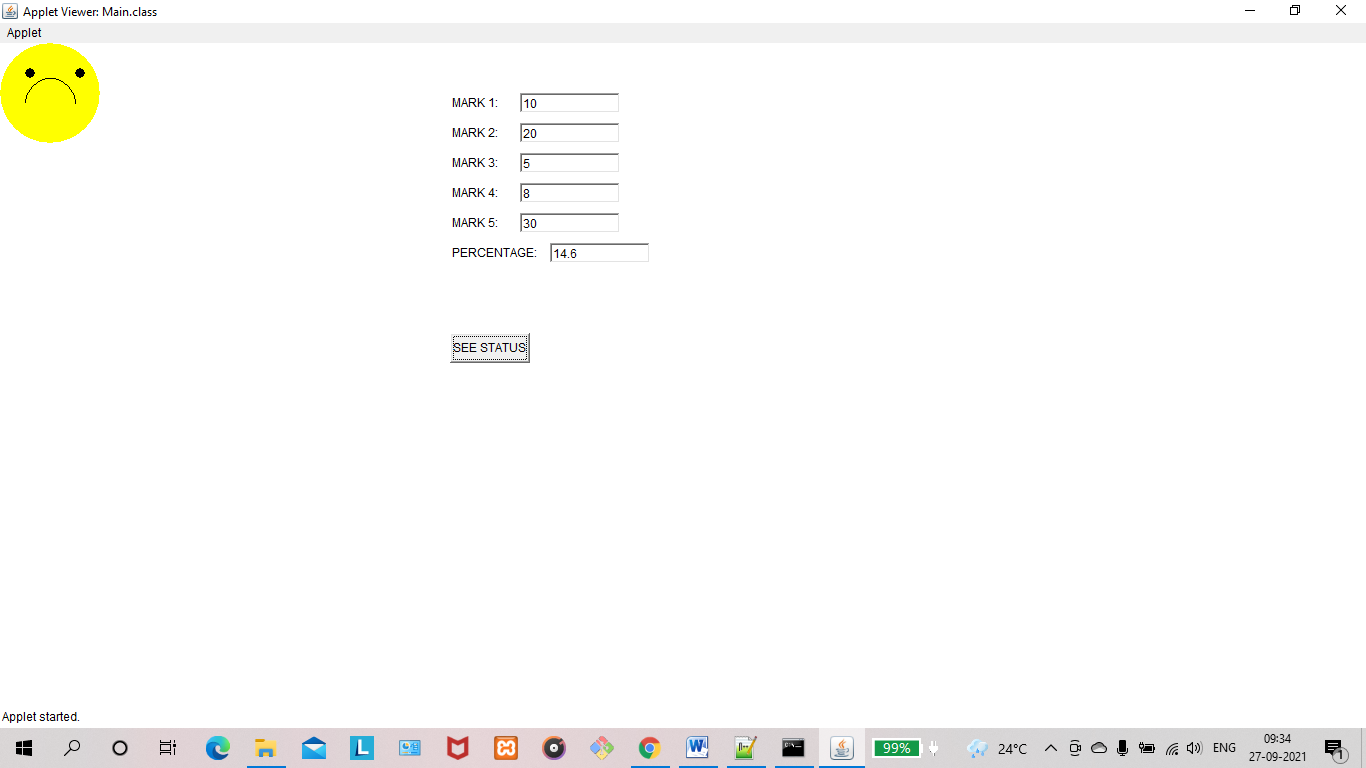
To find the percentage of marks obtained by a student in 5 subjects. Display a happy face if he secures above 50% or a sad face if otherwise.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.applet.\*;  import java.awt.\*;  import java.awt.Graphics;  import java.awt.event.\*;  public class Main extends Applet implements ActionListener {  Label l1,l2,l3,l4,l5,l6;  TextField t1,t2,t3,t4,t5,t6;  Button b;  public void init(){  l1 = new Label("MARK 1:");  t1 = new TextField();  l2 = new Label("MARK 2:");  t2 = new TextField();  l3 = new Label("MARK 3:");  t3 = new TextField();  l4 = new Label("MARK 4:");  t4 = new TextField();  l5 = new Label("MARK 5:");  t5 = new TextField();  l6 = new Label("PERCENTAGE:");  t6 = new TextField();  b = new Button("SEE STATUS");  setLayout(null);  l1.setBounds(450,50,70,20);  t1.setBounds(520,50,100,20);  l2.setBounds(450,80,70,20);  t2.setBounds(520,80,100,20);  l3.setBounds(450,110,70,20);  t3.setBounds(520,110,100,20);  l4.setBounds(450,140,70,20);  t4.setBounds(520,140,100,20);  l5.setBounds(450,170,70,20);  t5.setBounds(520,170,100,20);  l6.setBounds(450,200,100,20);  t6.setBounds(550,200,100,20);  b.setBounds(450,290,80,30);  add(l1);  add(l2);  add(l3);  add(l4);  add(l5);  add(l6);  add(t1);  add(t2);  add(t3);  add(t4);  add(t5);  add(t6);  add(b);  b.addActionListener(this);  }  public void actionPerformed(ActionEvent e){  float m1, m2,m3, m4,m5,percent;  m1= Float.parseFloat(t1.getText());  m2= Float.parseFloat(t2.getText());  m3= Float.parseFloat(t3.getText());  m4= Float.parseFloat(t4.getText());  m5= Float.parseFloat(t5.getText());  percent=((m1+m2+m3+m4+m5)\*100)/500;  t6.setText(String.valueOf(percent));  repaint();  }  public void paint(Graphics g){  float p;  p= Float.parseFloat(t6.getText());  if(p> 50.0) {  g.setColor(Color.YELLOW);  g.fillOval(0,0,100,100);  g.setColor(Color.black);  g.fillOval(25,25,10,10);  g.fillOval(65,25,10,10);  g.setColor(Color.black);  g.fillArc (25,35,50,50,0,-180);  }  else {  g.setColor(Color.YELLOW);  g.fillOval(0,0,100,100);  g.setColor(Color.black);  g.fillOval(25,25,10,10);  g.fillOval(75,25,10,10);  g.setColor(Color.black);  g.drawArc(25,35,50,50,0,180);  }  }  } |
| Main.html | <html>  <body>  <applet code="Main.class" width="300" height="300">  </applet>  </body>  </html> |

**OUTPUT:**





**RESULT:** The program is successfully executed and the output is verified.

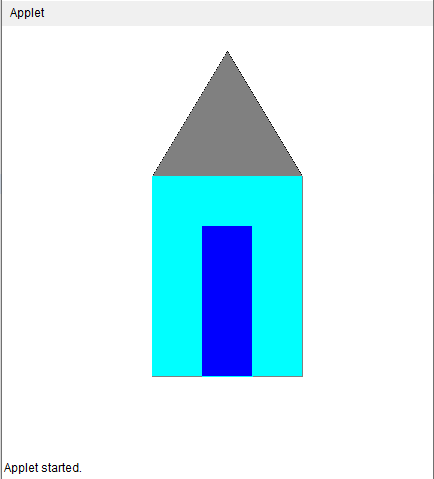
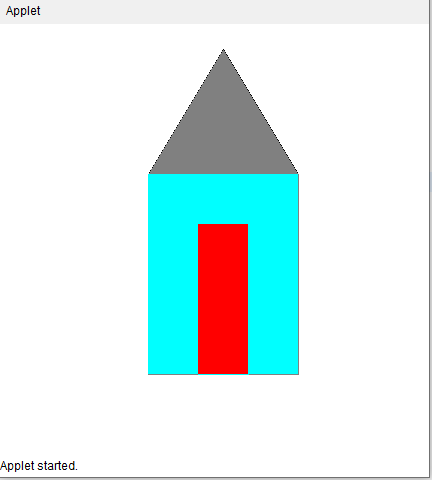
**PROGRAM NO : 38**

**AIM:** Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| House.java | import java.awt.\*;  import java.applet.\*;  import java.awt.event.\*;  public class House extends Applet implements MouseListener  {  int a,b;  public void init()  {  addMouseListener( this);  }  public void paint(Graphics g)  {  int x[]={150,300,225};  int y[]={150,150,25};  g.drawPolygon(x,y,3);  g.setColor(Color.GRAY);  g.fillPolygon(x,y,3);  g.drawRect(150,150,150,200);//House  g.setColor(Color.CYAN);  g.fillRect(150,150,150,200);  g.drawRect(200,200,50,150);//Door  g.setColor(Color.blue);  g.fillRect(200,200,50,150);  if(a>200 && a<300 && b>200 && b<300)  {  g.setColor(Color.red);  g.fillRect(200, 200, 50, 150);  }  }  public void mouseClicked(MouseEvent e)  {  }  public void mouseEntered(MouseEvent e)  {  }  @Override  public void mouseExited(MouseEvent e) {  }  public void mousePressed(MouseEvent e)  {  a=e.getX();  b=e.getY();  repaint();  }  public void mouseReleased(MouseEvent e)  {  }  } |
| Index.html | <html>  <body>  <applet code="House.class" width="600" height="600">  </applet>  </body>  </html> |

**OUTPUT:**

**** ****

**RESULT:** The program is successfully executed and the output is verified.

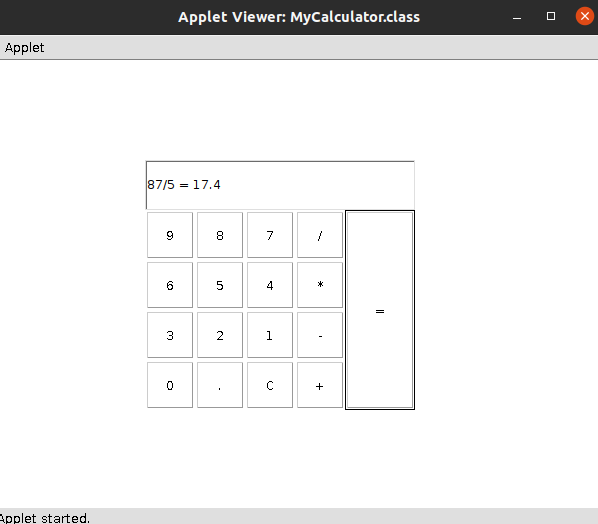
**PROGRAM NO : 39**

**AIM:** To implement a simple calculator using AWT components.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| MyCalculator.java | import java.awt.\*;  import java.applet.\*;  import java.awt.event.\*;  public class MyCalculator extends Applet implements ActionListener  {  TextField inp;  public void init()  {  setBackground(Color.white);  setLayout(null);  int i;  inp = new TextField();  inp.setBounds(150,100,270,50);  this.add(inp);  Button button[] = new Button[10];  for(i=0;i<10;i++)  {  button[i] = new Button(String.valueOf(9-i));  button[i].setBounds(150+((i%3)\*50),150+((i/3)\*50),50,50);  this.add(button[i]);  button[i].addActionListener(this);  }  Button dec=new Button(".");  dec.setBounds(200,300,50,50);  this.add(dec);  dec.addActionListener(this);  Button clr=new Button("C");  clr.setBounds(250,300,50,50);  this.add(clr);  clr.addActionListener(this);  Button operator[] = new Button[5];  operator[0]=new Button("/");  operator[1]=new Button("\*");  operator[2]=new Button("-");  operator[3]=new Button("+");  operator[4]=new Button("=");  for(i=0;i<4;i++)  {  operator[i].setBounds(300,150+(i\*50),50,50);  this.add(operator[i]);  operator[i].addActionListener(this);  }  operator[4].setBounds(350,150,70,200);  this.add(operator[4]);  operator[4].addActionListener(this);  }  String num1="";  String op="";  String num2="";  //Function to calculate the expression  public void actionPerformed(ActionEvent e)  {  String button = e.getActionCommand();  char ch = button.charAt(0);  if(ch>='0' && ch<='9'|| ch=='.')  {  if (!op.equals(""))  num2 = num2 + button;  else  num1 = num1 + button;  inp.setText(num1+op+num2);  }  else if(ch=='C')  {  num1 = op = num2 = "";  inp.setText("");  }  else if (ch =='=')  {  if(!num1.equals("") && !num2.equals(""))  {  double temp;  double n1=Double.parseDouble(num1);  double n2=Double.parseDouble(num2);  if(n2==0 && op.equals("/"))  {  inp.setText(num1+op+num2+" = Zero Division Error");  num1 = op = num2 = "";  }  else  {  if (op.equals("+"))  temp = n1 + n2;  else if (op.equals("-"))  temp = n1 - n2;  else if (op.equals("/"))  temp = n1/n2;  else  temp = n1\*n2;  inp.setText(num1+op+num2+" = "+temp);  num1 = Double.toString(temp);  op = num2 = "";  }  }  else  {  num1 = op = num2 = "";  inp.setText("");  }  }  else  {  if (op.equals("") || num2.equals(""))  op = button;  else  {  double temp;  double n1=Double.parseDouble(num1);  double n2=Double.parseDouble(num2);  if(n2==0 && op.equals("/"))  {  inp.setText(num1+op+num2+" = Zero Division Error");  num1 = op = num2 = "";  }  else  {  if (op.equals("+"))  temp = n1 + n2;  else if (op.equals("-"))  temp = n1 - n2;  else if (op.equals("/"))  temp = n1/n2;  else  temp = n1\*n2;  num1 = Double.toString(temp);  op = button;  num2 = "";  }  }  inp.setText(num1+op+num2);  }  }  } |
| index.html | <html>  <body>  <applet code="MyCalculator.class" width="600" height="600">  </applet>  </body>  </html> |

**OUTPUT:**



**RESULT:** The program is successfully executed and the output is verified.

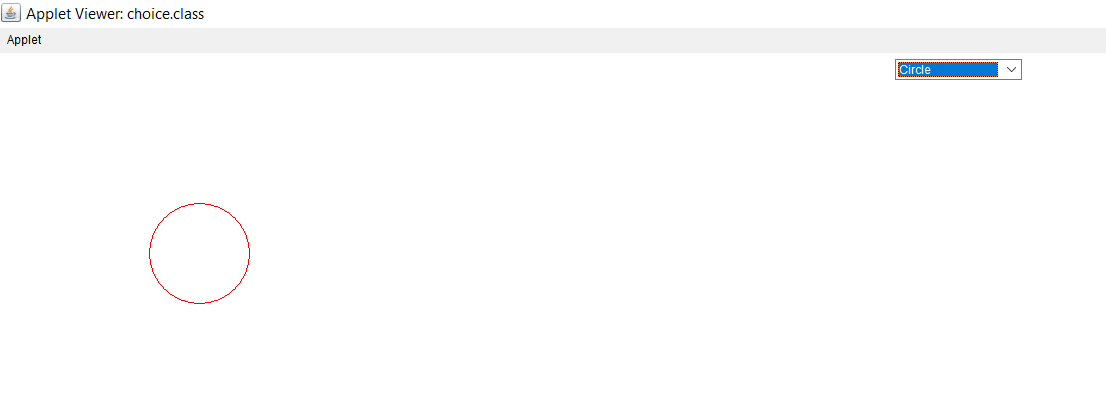
**PROGRAM NO : 40**

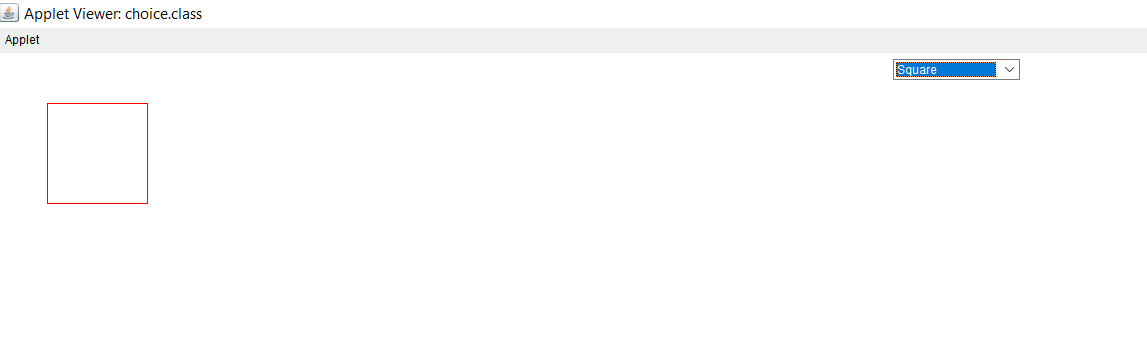
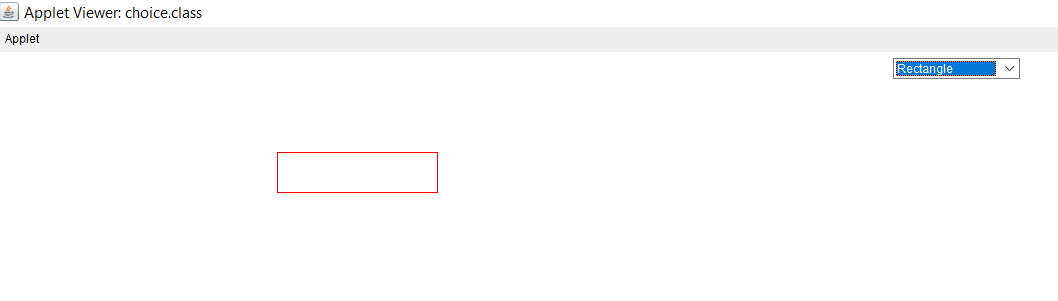
**AIM:** To develop a program that has a Choice component which contains the names of shapes such as rectangle, triangle, square and circle. Draw the corresponding shapes for given parameters as per user’s choice.

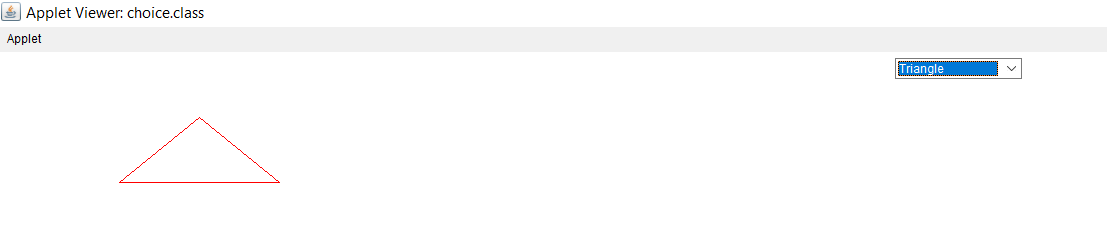
**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.applet.\*;  import java.awt.\*;  import java.awt.Graphics;  import java.awt.event.\*;  public class Main extends Applet implements ItemListener  {  Choice figure = new Choice();  int Select;  public void init()  {  figure.addItem("Select your choice");  figure.addItem("Rectangle");  figure.addItem("Square");  figure.addItem("Circle");  figure.addItem("Triangle");  add(figure);  figure.addItemListener(this);  }  public void itemStateChanged (ItemEvent e)  {  Select = figure.getSelectedIndex();  repaint();  }  public void paint(Graphics g)  {  g.setColor(Color.red);  super.paint(g);  if (Select == 1)  {  g.drawRect(280, 100, 160,40);  }  if (Select == 2)  {  g.drawRect(50,50,100,100);  }  if (Select == 3)  {  g.drawOval(150,150,100,100);  }  if (Select ==4)  {  g.drawLine(120, 130, 280, 130);  g.drawLine(120, 130, 200, 65);  g.drawLine(200, 65, 280, 130);  }  }  } |
| index.html | <html>  <body>  <applet code="Main.class" width="600" height="600">  </applet>  </body>  </html> |

**OUTPUT:**







**RESULT:** The program is successfully executed and the output is verified.

**PROGRAM NO : 41**

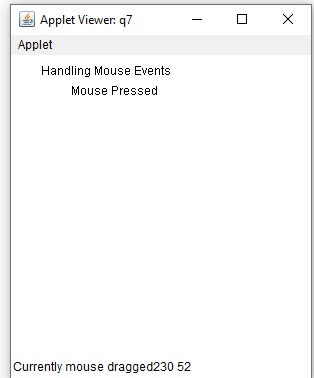
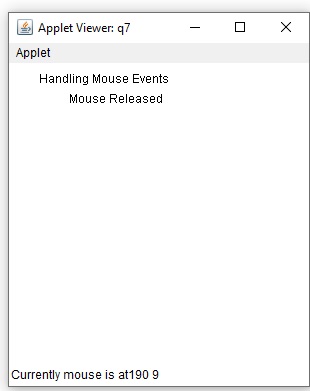
**AIM:** To develop a program to handle all mouse events and window events.

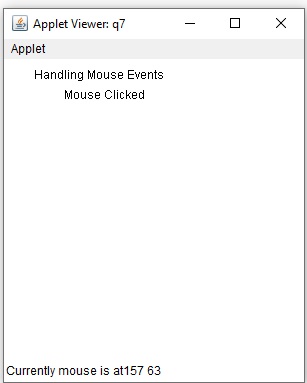
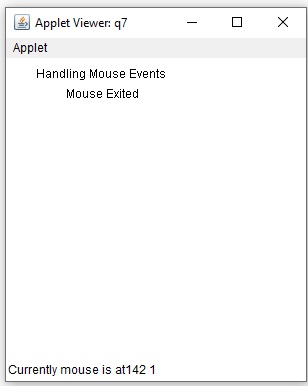
**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.awt.\*;  import java.applet.\*;  import java.awt.event.\*;  public class Main extends Applet implements MouseListener,MouseMotionListener  {  int mx=0;  int my=0;  String msg="";  public void init()  {  addMouseListener(this);  addMouseMotionListener(this);  }  public void mouseClicked(MouseEvent me)  {  mx=20;  my=40;  msg="Mouse Clicked";  repaint();  }  public void mousePressed(MouseEvent me)  {  mx=30;  my=60;  msg="Mouse Pressed";  repaint();  }  public void mouseReleased(MouseEvent me)  {  mx=30;  my=60;  msg="Mouse Released";  repaint();  }  public void mouseEntered(MouseEvent me)  {  mx=40;  my=80;  msg="Mouse Entered";  repaint();  }  public void mouseExited(MouseEvent me)  {  mx=40;  my=80;  msg="Mouse Exited";  repaint();  }  public void mouseDragged(MouseEvent me)  {  mx=me.getX();  my=me.getY();  showStatus("Currently mouse dragged"+mx+" "+my);  repaint(); }  public void mouseMoved(MouseEvent me)  {  mx=me.getX();  my=me.getY();  showStatus("Currently mouse is at"+mx+" "+my);  repaint();  }  public void paint(Graphics g)  {  g.drawString("Handling Mouse Events",30,20);  g.drawString(msg,60,40);  }  } |
| index.html | <html>  <body>  <applet code="Main.class" width="600" height="600">  </applet>  </body>  </html> |

**OUTPUT:**

****

****

****

**RESULT:** The program is successfully executed and the output is verified.

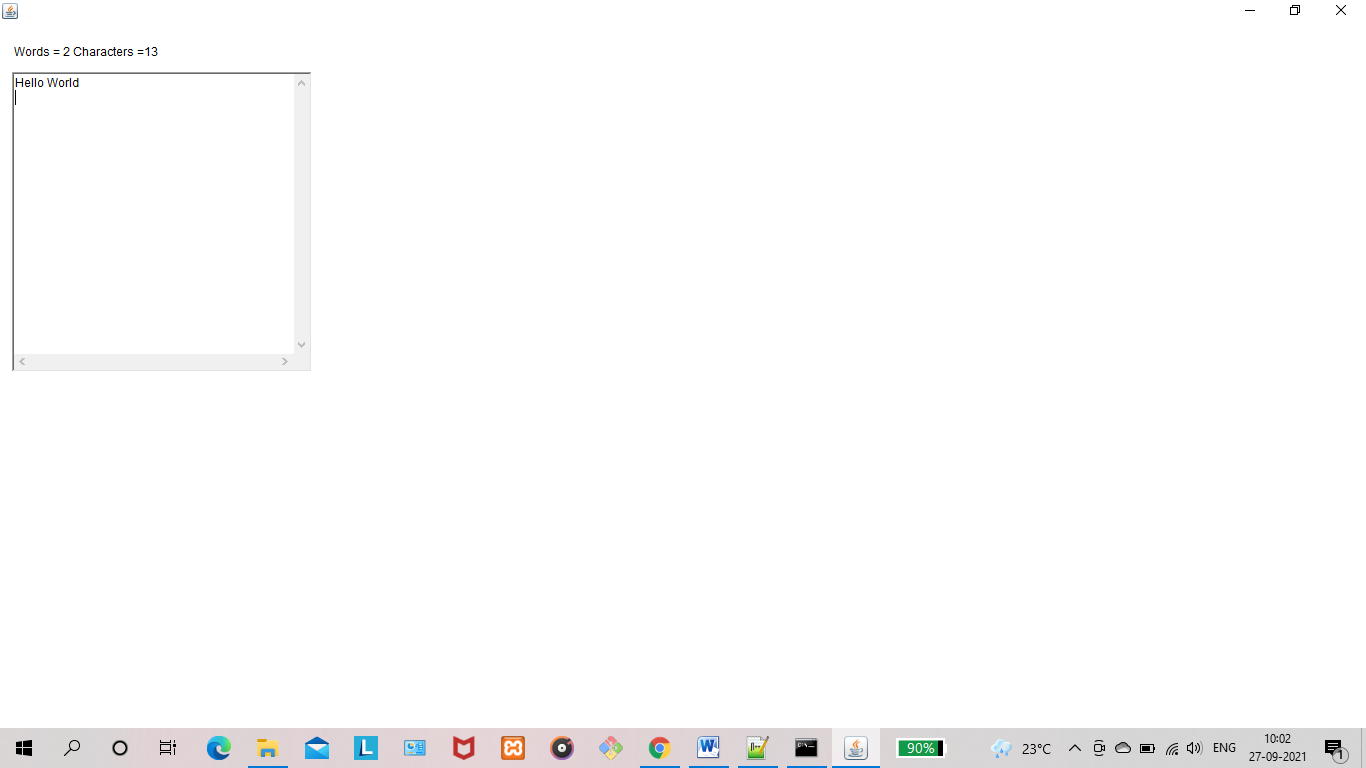
**PROGRAM NO : 42**

**AIM:** To develop a program to handle Key events.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.awt.\*;  import java.awt.event.\*;  public class keyexamp extends Frame implements KeyListener  {  Label l;  TextArea a;  keyexamp()  {  l=new Label();  l.setBounds(20,50,200,20);  a=new TextArea();  a.setBounds(20,80,300,300);  a.addKeyListener(this);  add(l);  add(a);  setSize(400,400);  setLayout(null);  setVisible(true);  }  public void keyPressed(KeyEvent e)  {}  public void keyReleased(KeyEvent e)  {  String t=a.getText();  String w[]=t.split("\\s");  l.setText("Words = "+w.length+" Characters ="+t.length());  }  public void keyTyped(KeyEvent e)  {}  public static void main(String args[])  {  new keyexamp();}  } |

**OUTPUT:**



**RESULT:** The program is successfully executed and the output is verified.

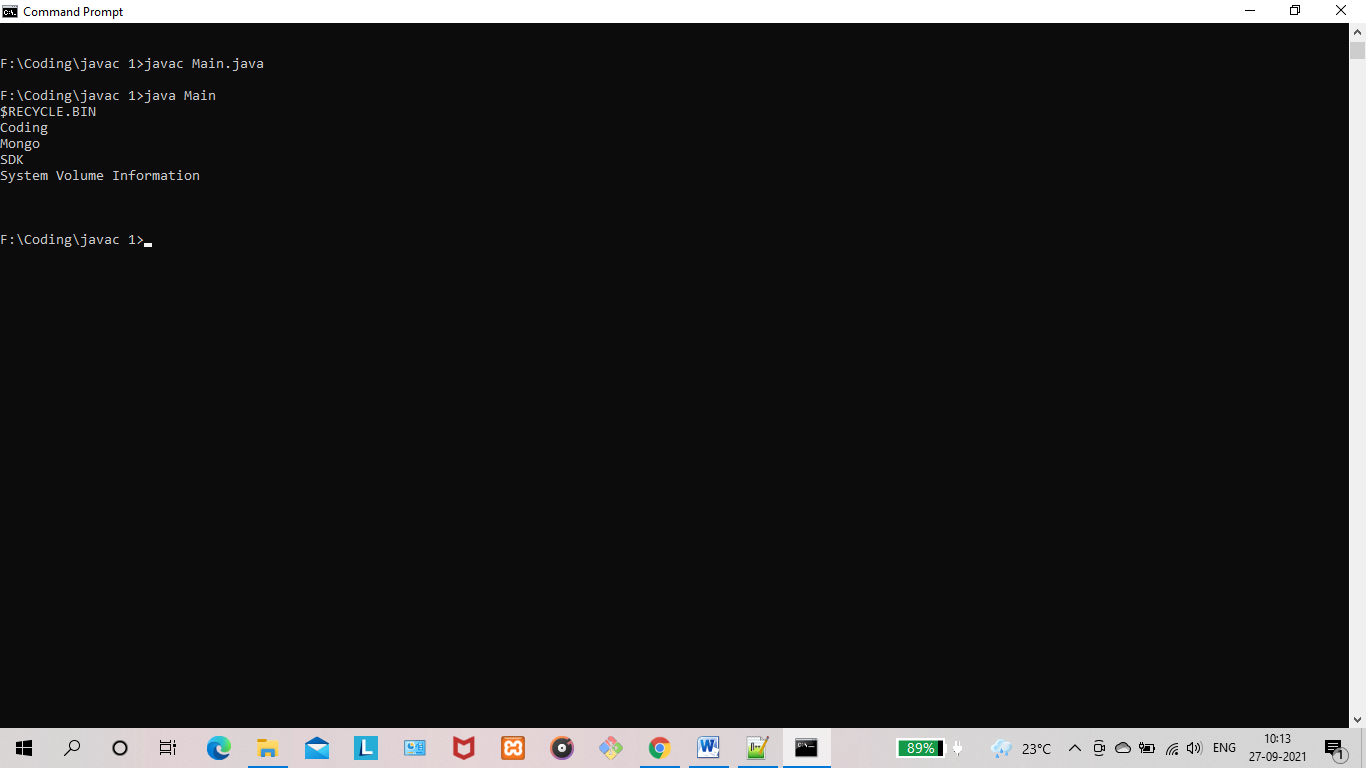
**PROGRAM NO : 43**

**AIM:** To write program to list the sub directories and files in a given directory and also search for a file name.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.io.\*;  public class Main {  public static void main(String[] args) {  File file = new File("../../../");  String[] fileList = file.list();  for(String str : fileList) {  System.out.println(str);  }  FilenameFilter filter = new FilenameFilter() {  public boolean accept (File dir, String name) {  return name.startsWith("A");  }  };  System.out.println("\n");  String[] children = file.list(filter);  if (children == null) {  System.out.println("Either dir does not exist or is not a directory");  } else {  for (int i = 0; i< children.length; i++) {  String filename = children[i];  System.out.println(filename);  }  }  }  } |

**OUTPUT:**

****

**RESULT:** The program is successfully executed and the output is verified.

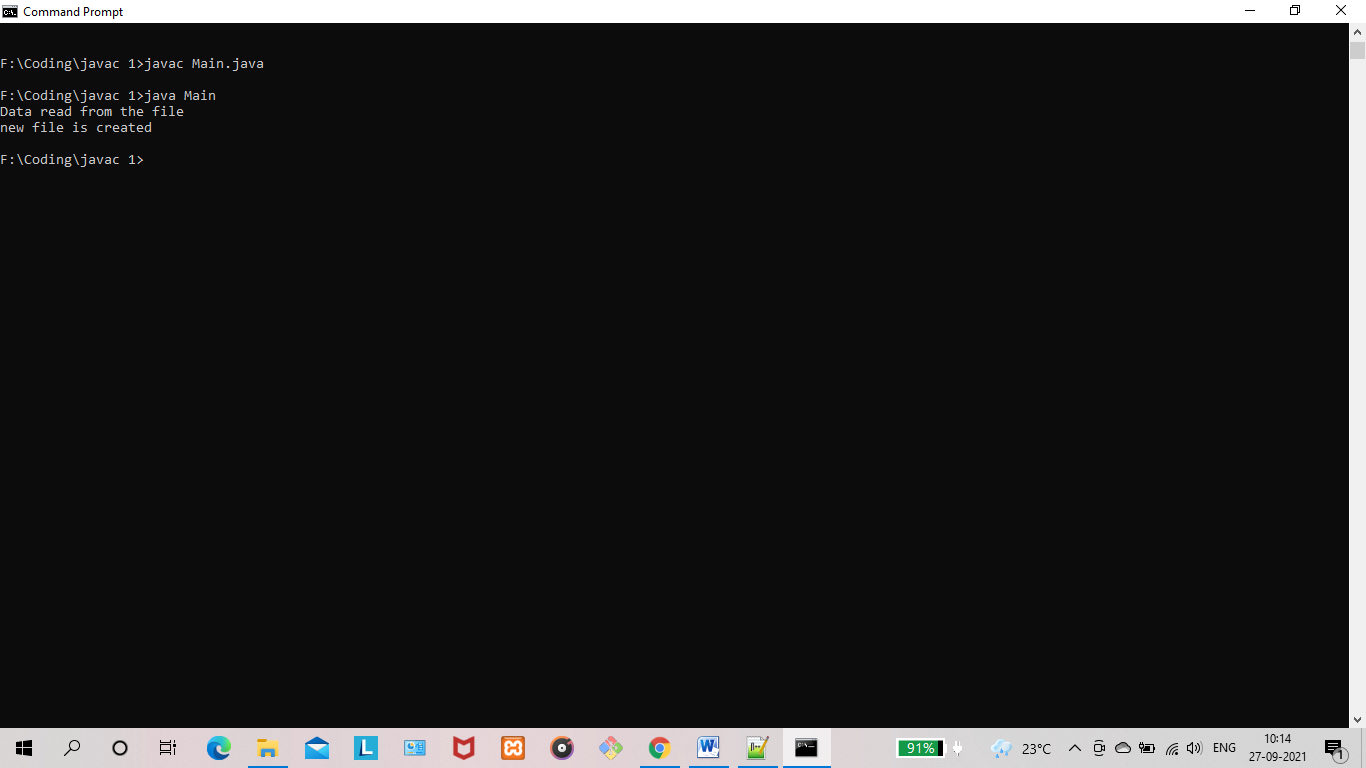
**PROGRAM NO : 44**

**AIM:** To write a program to write to a file, then read from the file and display the contents on the console.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.io.BufferedReader;  import java.io.FileReader;  import java.io.FileWriter;  import java.io.IOException;  public class Main {  public static void main(String[] args) {  try {  FileWriter writer = new FileWriter("java\_write.txt",true);  writer.write("new file is created");  writer.close();  FileReader reader = new FileReader("java\_write.txt");  BufferedReader br= new BufferedReader(reader);  String line;  System.out.println("Data read from the file");  while ((line = br.readLine()) != null) {  System.out.println(line);  }  reader.close();  } catch (IOException e) {  System.out.println("-----Error-----");  }  }  } |

**OUTPUT:**

****

**RESULT:** The program is successfully executed and the output is verified.

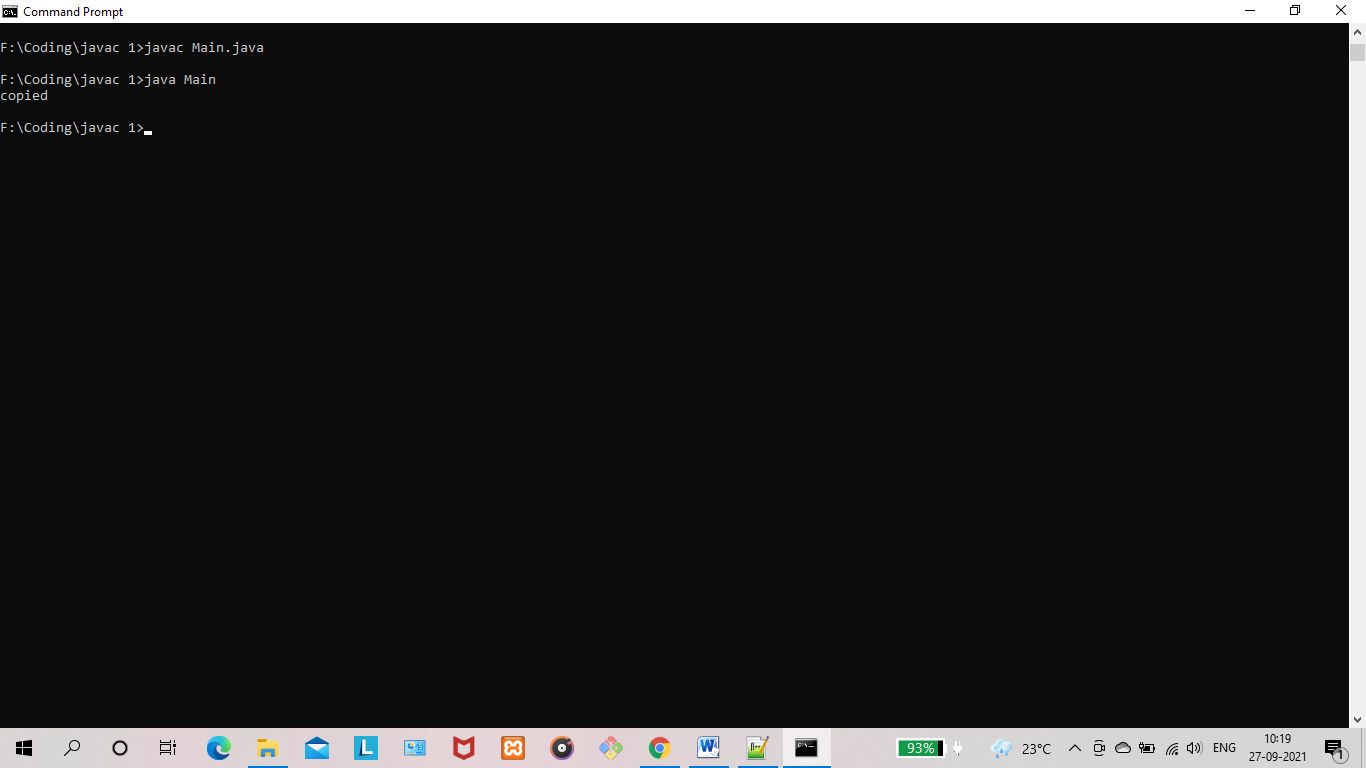
**PROGRAM NO : 45**

**AIM:** To write a program to copy one file to another.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Main.java | import java.io.FileInputStream;  import java.io.FileOutputStream;  import java.io.IOException;  public class Main {  public static void main(String[] args) throws IOException{  // TODO Auto-generated method stub  FileInputStream fileinput = new FileInputStream("source.txt");  FileOutputStream fileoutput = new FileOutputStream("destination.txt");  int i;  while((i = fileinput.read()) != -1){  fileoutput.write(i);  }  System.out.println("copied");  fileinput.close();  fileoutput.close();  }  } |

**OUTPUT:**

****

**RESULT:** The program is successfully executed and the output is verified.

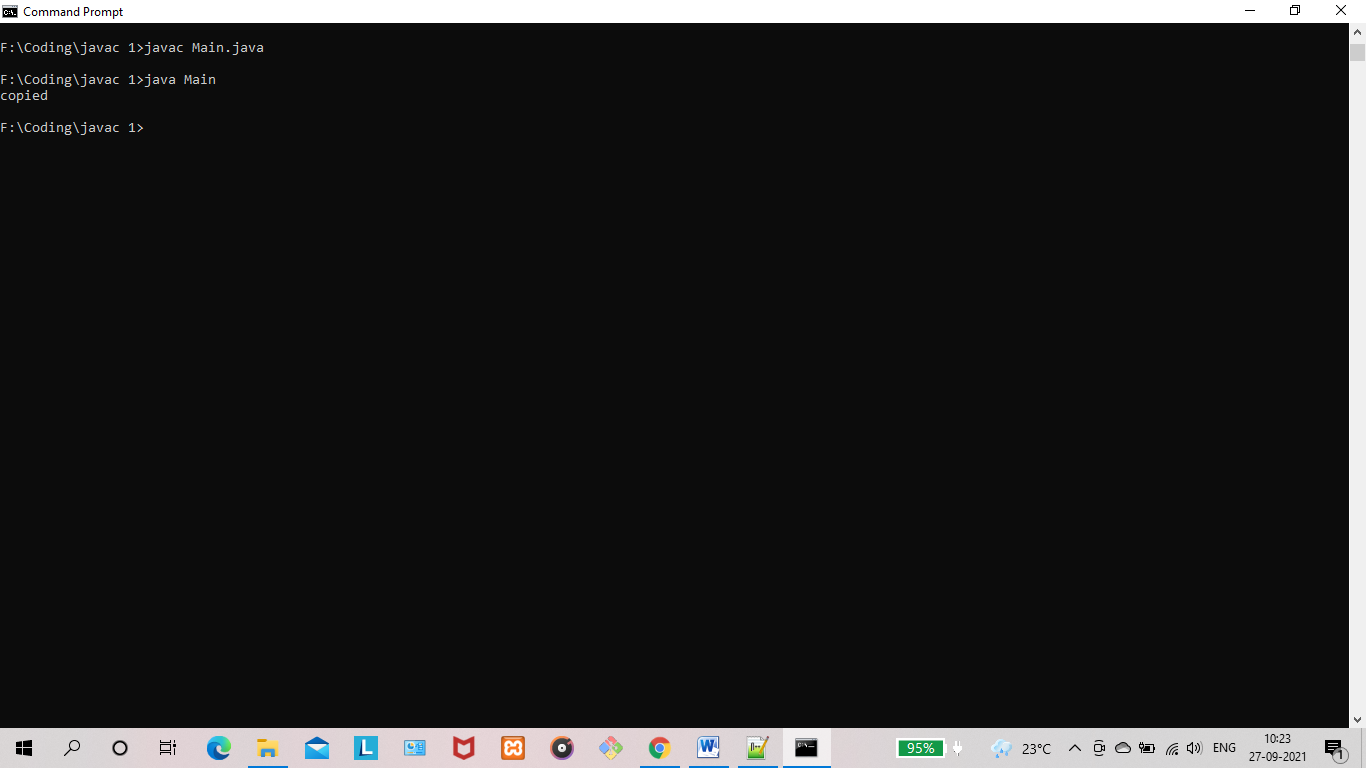
**PROGRAM NO : 46**

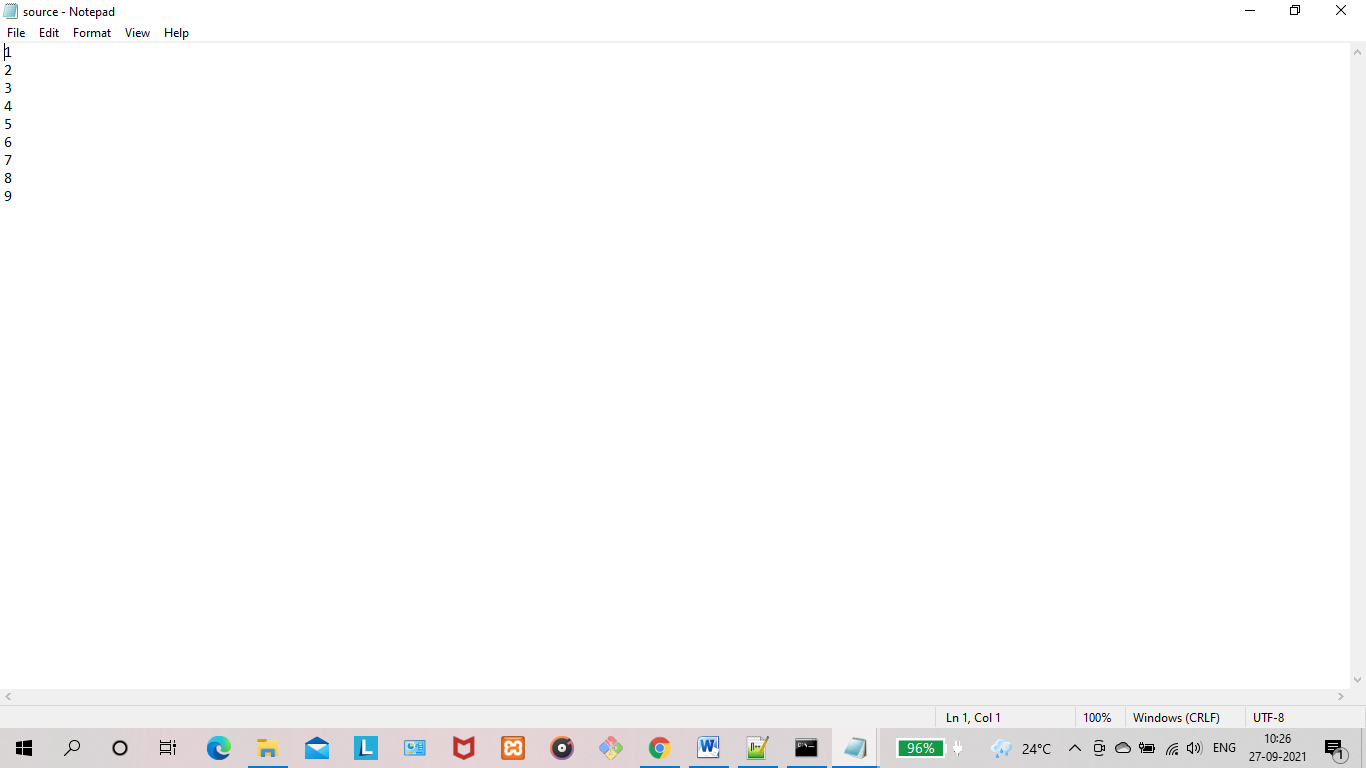
**AIM:** To write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

**PROGRAM CODE:**

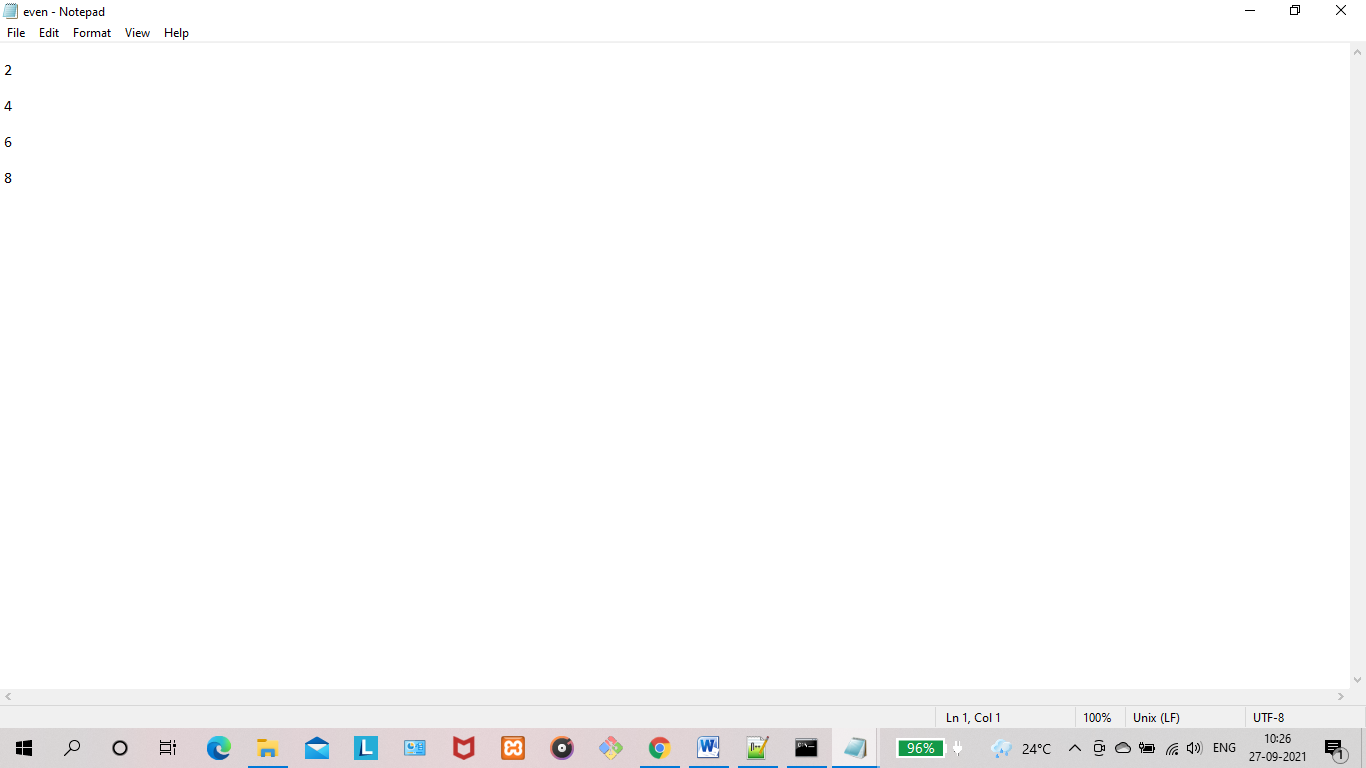
|  |  |
| --- | --- |
| Main.java | import java.io.FileInputStream;  import java.io.FileOutputStream;  import java.io.IOException;  public class Main {  public static void main(String[] args) throws IOException {  // TODO Auto-generated method stub  FileInputStream source = new FileInputStream ("source.txt");  FileOutputStream destination\_odd = new FileOutputStream ("odd.txt");  FileOutputStream destination\_even = new FileOutputStream ("even.txt");  int i;  while((i = source.read()) != -1){  if(i%2==0) {  destination\_even.write(i);  }  else {  destination\_odd.write(i);  }  }  System.out.println("copied");  source.close();  destination\_even.close();  destination\_odd.close();  }  } |

**OUTPUT:**

****

****

****

****

**RESULT:** The program is successfully executed and the output is verified.

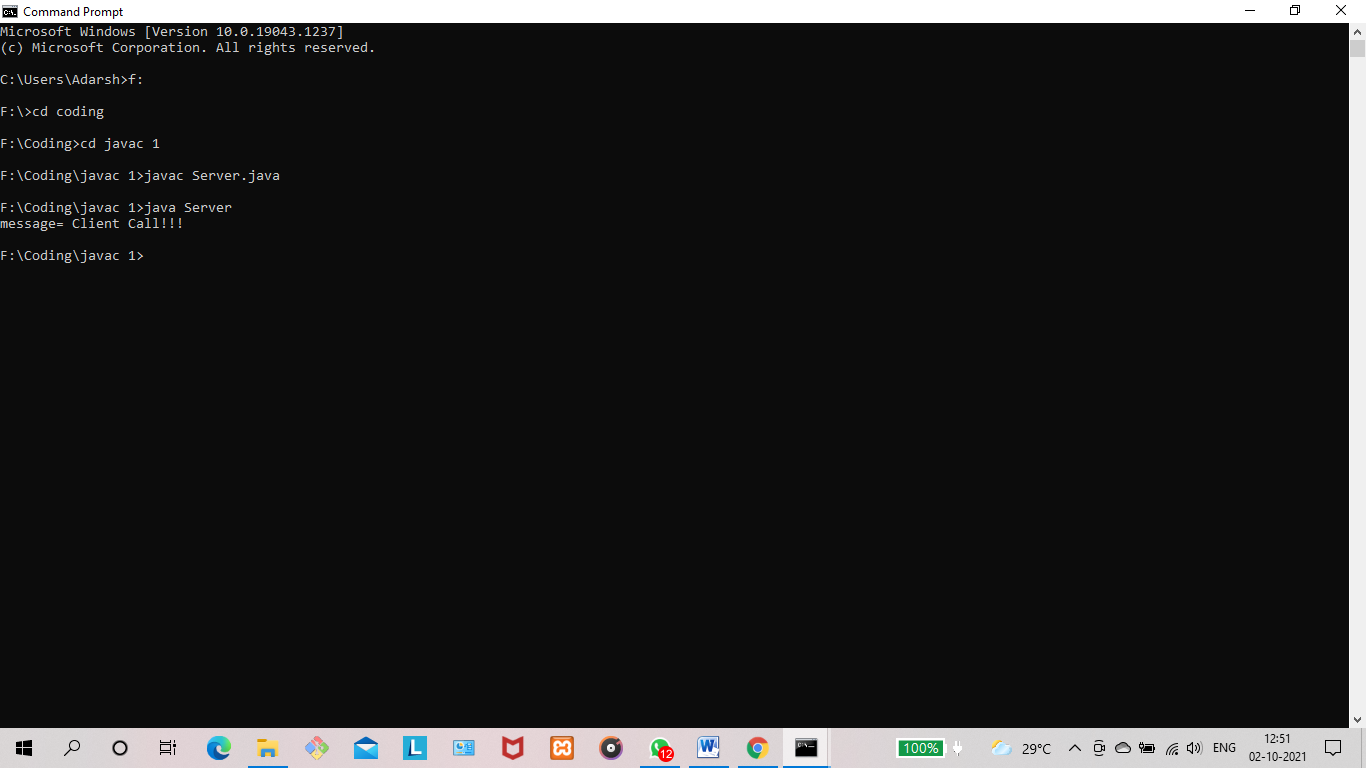
**PROGRAM NO : 47**

**AIM:** To implement client server communication using Socket – TCP/IP.

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Client.java | import java.io.\*;  import java.net.\*;  public class Client {  public static void main(String[] args) {  try{  Socket socket=new Socket("localhost",7011);  DataOutputStream dout=new DataOutputStream(socket.getOutputStream());  dout.writeUTF("Client Call!!!");  dout.flush();  dout.close();  socket.close();  }catch(Exception e){System.out.println(e);}  }  } |
| Server.java | import java.io.\*;  import java.net.\*;  public class Server {  public static void main(String[] args){  try{  ServerSocket serverSocket=new ServerSocket(7011);  Socket socket=serverSocket.accept();  //establishes connection  DataInputStream dis=new DataInputStream(socket.getInputStream());  String str=(String)dis.readUTF();  System.out.println("message= "+str);  serverSocket.close();  }  catch(Exception e){  System.out.println(e);  }  }  } |

**OUTPUT:**

****

**RESULT:** The program is successfully executed and the output is verified.

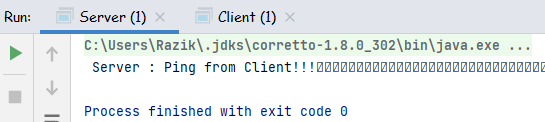
**PROGRAM NO : 48**

**AIM:** Client Server communication using DatagramSocket - UDP

**PROGRAM CODE:**

|  |  |
| --- | --- |
| Client.java | import java.io.\*;  import java.net.\*;  public class Client {  public static void main(String[] args) throws IOException {  DatagramSocket client= new DatagramSocket();  InetAddress add=InetAddress.getByName("localhost");  String str ="Ping from Client!!!";  byte[] bufBytes = str.getBytes();  DatagramPacket datagramPacket=new DatagramPacket(bufBytes,bufBytes.length,add,4220);  client.send(datagramPacket);  client.close();  }  } |
| Server.java | import java.io.\*;  import java.net.\*;  public class Server {  public static void main(String[] args) throws IOException {  DatagramSocket server=new DatagramSocket(4220);  byte[] buf=new byte[256];  DatagramPacket packet=new DatagramPacket(buf,buf.length);  server.receive(packet);  String response =new String(packet.getData());  System.out.println(" Server : "+response);  server.close();  }  } |

**OUTPUT:**

****

**RESULT:** The program is successfully executed and the output is verified.