**20MCA132 - OBJECT ORIENTED PROGRAMMING LAB**

*Lab Report Submitted By*

**NIKHIL JAIS**

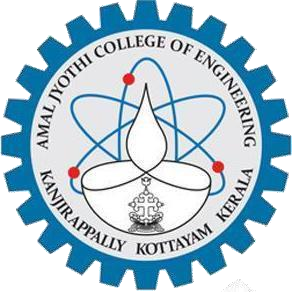
**Reg. No.: AJC21MCA-2083**

*In Partial fulfillment for the Award of the Degree Of*

**MASTER OF COMPUTER APPLICATIONS (2 Year)**

**(MCA)**

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING KANJIRAPPALLY**

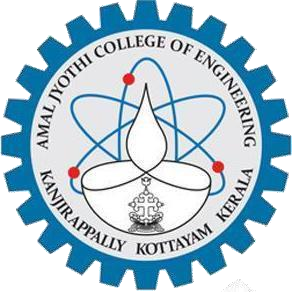
[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with ‘A’ grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

**2021-2022**

# DEPARTMENT OF COMPUTER APPLICATIONS

**AMAL JYOTHI COLLEGE OF ENGINEERING**

**KANJIRAPPALLY**



**CERTIFICATE**

This is to certify that the Lab report, **“20MCA132 OBJECT ORIENTED**

# PROGRAMMING LAB” is the bonafide work of NIKHIL JAIS

**(Reg.No: AJC21MCA-2083 )** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-22.

**MS. Gloriya Mathew Rev.Fr.Dr.Rubin Thottupurathu Jose**

**Lab In-Charge Head of the Department**

**Internal Examiner External Examiner**

# CONTENT

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **Content** | **Date** | **Page No** |
| **1** | 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. | **29-03-2022** | **1-3** |
| **2** | Read 2 matrices from the console and perform matrix addition. | **06-04-2022** | **4-7** |
| **3** | Add complex numbers | **06-04-2022** | **8-9** |
| **4** | Read a matrix from the console and check whether it is symmetric or not. | **06-04-2022** | **10-12** |
| **5** | 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.. | **17-05-2022** | **13-14** |
| **6** | Search an element in an array | **22-04-2022** | **15-16** |
| **7** | Write a program to Perform string manipulations | **22-04-2022** | **17** |
| **8** | Program to Sort strings | **22-04-2022** | **18-19** |
| **9** | 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. | **22-04-2022** | **20-22** |
| **10** | Area of different shapes using overloaded functions | **17-05-2022** | **23-24** |
| **11** | 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 | **17-04-2022** | **25-26** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | display function to display all the data members. Use array of objects to display details of N teachers. |  |  |
| **12** | 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 | **18-05-2022** | **27-32** |
| **13** | 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. | **18-05-2022** | **33-36** |
| **14** | Create classes Student and Sports. Create another class Result inherited from Student and Sports.  Display the academic and sports score of a student. | **24-05-2022** | **37-40** |
| **15** | 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.. | **24-05-2022** | **41-45** |
| **16** | Prepare bill with the given format using calculate method from interface. | **24-05-2022** | **46-47** |
| **17** | 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 | **31-05-2022** | **48-50** |
| **18** | Write a user defined exception class to authenticate the user name and password. | **31-05-2022** | **51-52** |
| **19** | Find the average of N positive integers, raising a user defined exception for each negative input. | **31-05-2022** | **53-54** |
| **20** | Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface). | **07-06-2022** | **55-56** |

|  |  |  |  |
| --- | --- | --- | --- |
| **21** | Program to create a generic stack and do the Push and Pop operations. | **07-06-2022** | **57-58** |
| **22** | Maintain a list of Strings using ArrayList from collection framework, perform built-in operations | **07-06-2022** | **59** |
| **23** | Program to demonstrate the creation of queue object using the PriorityQueue class | **07-06-2022** | **60-63** |
| **24** | Program to demonstrate the addition and deletion of elements in deque. | **07-06-2022** | **64-65** |
| **25** | Write a Java program to compare two hash set | **07-06-2022** | **66-67** |
| **26** | Program to demonstrate the working of Map interface by adding, changing and removing Elements. | **07-06-2022** | **68** |
| **27** | Program to find maximum of three numbers using AWT. | **09-06-2022** | **69-70** |
| **28** | Implement a simple calculator using AWT components | **09-06-2022** | **71-73** |
| **29** | Develop a program to handle all mouse events and window events | **29-05-2022** | **74-75** |
| **30** | Develop a program to handle Key events. | **29-05-2022** | **76-77** |
| **31** | Write a program to write to a file, then read from the file and display the contents on the console. | **30-05-2022** | **78-79** |
| **32** | Write a program to copy one file to another | **30-05-2022** | **80-81** |
| **33** | Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files. | **30-05-2022** | **82-83** |

|  |
| --- |
| **Name: Nikhil Jais**  **Roll No: 25**  **Batch: B**  **Date: 29/03/22** |

**OBJECT ORIENTED PROGRAMMING LAB**

**Experiment No.: 1**

## Aim

Define a class ‘product’ with data members pcode, pname and price. Create three objects of the class and find the product having the lowest price .

## Procedure

public class Product{

String pcode, pname; double price;

public void details(){

System.out.println("The product name is : "+pname);

System.out.println("The product code is : "+pcode);

System.out.println("The product price is : "+price); System.out.println("\n");

}

public static void main(String[] args){

Product prod1= new Product(); prod1.pcode= "P101"; prod1.pname= "Book"; prod1.price= 55.5; prod1.details();

Product prod2= new Product(); prod2.pcode= "P202"; prod2.pname= "Pencil"; prod2.price= 10.0; prod2.details();

Product prod3= new Product(); prod3.pcode= "P303"; prod3.pname= "Pen"; prod3.price= 20.2; prod3.details();

System.out.println("\n"); if((prod1.price < prod2.price)&&(prod1.price < prod3.price)) {

System.out.println("The price of "+prod1.pname+" is the lowest"); }

else if((prod2.price < prod1.price)&&(prod2.price < prod3.price))

{

### System.out.println("The price of "+prod2.pname+" is the lowest");

}  else{

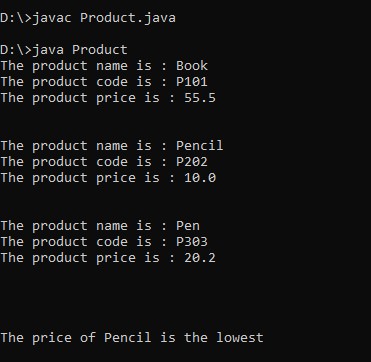
System.out.println("The price of "+prod3.pname+" is the lowest");

}

}

}

**Output Screenshot**



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No: 25**  **Batch: B**  **Date: 06/04/22** |

**Experiment No.: 2**

## Aim

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

import java.util.\*;

class MatrixAddition{

public static void main(String[] args){

int row, col;

Scanner sc= new Scanner(System.in);

System.out.print("Enter the number of rows for the Matrices : "); row= sc.nextInt();

System.out.print("Enter the number of columns for the Matrices : "); col= sc.nextInt();

int[][] matrixA= new int[row][col]; int[][] matrixB= new int[row][col]; int[][] matrixSum= new int[row][col];

System.out.println("Enter the elements for the Matrix A : "); for(int i=0;i<row;i++){

for(int j=0;j<col;j++){ matrixA[i][j]= sc.nextInt();

}

}

System.out.println("\n");

System.out.println("Enter the elements for the Matrix B : "); for(int i=0;i<row;i++){ for(int j=0;j<col;j++){ matrixB[i][j]= sc.nextInt();

}

}

System.out.println("\n");

System.out.println("Matrix A is : "); for(int i=0;i<row;i++){ for(int j=0;j<col;j++){

System.out.print(matrixA[i][j]+" ");

}

System.out.println("\n");

}

System.out.println("Matrix B is : "); for(int i=0;i<row;i++){ for(int j=0;j<col;j++){

System.out.print(matrixB[i][j]+" ");

}

System.out.println("\n");

}

for(int i=0;i<row;i++){ for(int j=0;j<col;j++){ matrixSum[i][j]= matrixA[i][j] + matrixB[i][j]; }

}

System.out.println("Resultant of the Matrix Addition is : "); for(int i=0;i<row;i++){ for(int j=0;j<col;j++){

System.out.print(matrixSum[i][j]+" ");

}

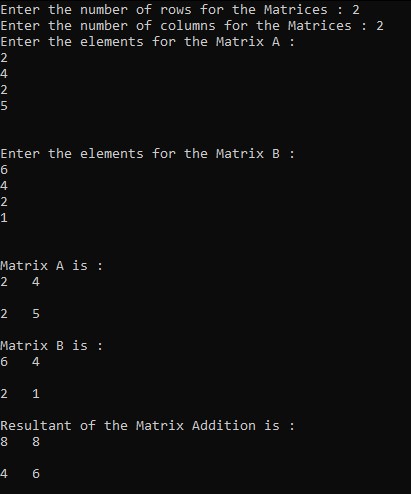
System.out.println("\n");

}

}

}

**Output Screenshot**



OBJECT ORIENTED PROGRAMMING LAB Dept. of Computer Applications

|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:06-04-2022** |

**Experiment No.: 3**

## Aim

Add complex numbers **Procedure**

import java.util.\*; class Complex {

int real, imaginary;

Complex(){}

Complex(int tempReal, int tempImaginary){

real = tempReal; imaginary = tempImaginary; }

Complex addComp(Complex C1, Complex C2){ Complex temp = new Complex(); temp.real = C1.real + C2.real;

temp.imaginary = C1.imaginary + C2.imaginary; return temp;

}

Complex subtractComp(Complex C1, Complex C2){ Complex temp = new Complex(); temp.real = C1.real - C2.real;

temp.imaginary = C1.imaginary - C2.imaginary; return temp;

}

void printComplexNumber(){

System.out.println("Complex number: "

+ real + " + "

+ imaginary + "i");

OBJECT ORIENTED PROGRAMMING LAB Dept. of Computer Applications

}

}

public class ComplexNumber { public static void main(String[] args){

Complex C1 = new Complex(5, 2);

C1.printComplexNumber();

Complex C2 = new Complex(8, 7);

C2.printComplexNumber();

Complex C3 = new Complex();

C3 = C3.addComp(C1, C2);

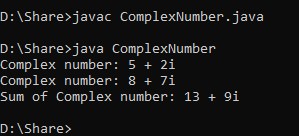
System.out.print("Sum of ");

C3.printComplexNumber();

}

}

## Output Screenshot



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No: 25**  **Batch: B**  **Date: 06/04/22** |

**Experiment No.: 4**

**Aim**

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

## Procedure

import java.util.\*;

class SymmetricMatrix{

public static void main(String[] args){

int row, col;

Scanner sc= new Scanner(System.in); boolean isSymmetic= true;

System.out.print("Enter the number of rows for the Matrices : "); row= sc.nextInt();

System.out.print("Enter the number of columns for the Matrices : "); col= sc.nextInt();

int[][] matrix= new int[row][col];

System.out.println("Enter the elements for the Matrix : "); for(int i=0;i<row;i++){ for(int j=0;j<col;j++){ matrix[i][j]= sc.nextInt(); }

}

System.out.println("\n");

System.out.println("The entered matrix is : "); for(int i=0;i<row;i++){ for(int j=0;j<col;j++){

System.out.print(matrix[i][j]+" ");

}

System.out.println("\n"); }

for(int i=0;i<row;i++){

for(int j=0;j<col;j++){ if(i!=j){

if(matrix[i][j]!=matrix[j][i]){

isSymmetic= false;

break;

}

}

}

if(!isSymmetic)

break;

}

if(isSymmetic){

System.out.println("The entered matrix is Symmetric Matrix");

} else{

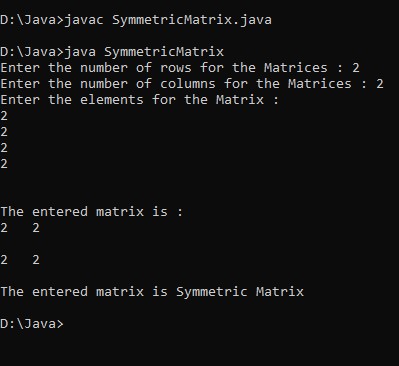
### System.out.println("The entered matrix is not a Symmetric Matrix");

}

}

}

**Output Screenshot**



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No: 25**  **Batch: B**  **Date: 17/05/22** |

**Experiment 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.

## Procedure class Cpu{

float price; class Processor{ float cores=51;

String manufacturer="Intel";

}

static class Ram

{ static float memory=512; String m\_manufacturer="Asus";

}

}

public class CpuRam{

public static void main(String[] args){

Cpu MyCpu = new Cpu();

Cpu.Processor MyProcessor = MyCpu.new Processor();

Cpu.Ram MyRam = new Cpu.Ram();

System.out.println("Processor core = " + MyProcessor.cores);

System.out.println("Processor manufacture = " +

MyProcessor.manufacturer);

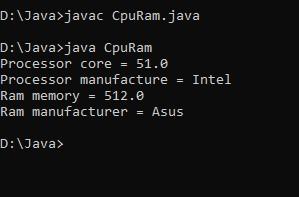
System.out.println("Ram memory = " + Cpu.Ram.memory);

System.out.println("Ram manufacturer = " + MyRam.m\_manufacturer);

}

}

**Output Screenshot**



### Experiment No.: 6 Name: Nikhil Jais

**Aim Roll No:25**

**Batch: B** Search an element in an array.

**Date:22/04/22**

### PROCEDURE

import java.util.Scanner; public class Search\_Element

{ public static void main(String[] args)

{ int n, x, flag = 0, i = 0;

Scanner s = new Scanner(System.in);

System.out.print("Enter no. of elements you want in array:"); n = s.nextInt(); int a[] = new int[n];

System.out.println("Enter all the elements:"); for(i = 0; i < n; i++)

{ a[i] = s.nextInt();

}

System.out.print("Enter the element you want to find:"); x = s.nextInt(); for(i = 0; i < n; i++)

{ if(a[i] == x)

{ flag = 1; break; } else

{ flag = 0;

}

}

if(flag == 1)

{

System.out.println("Element found at position:"+(i + 1));

} else

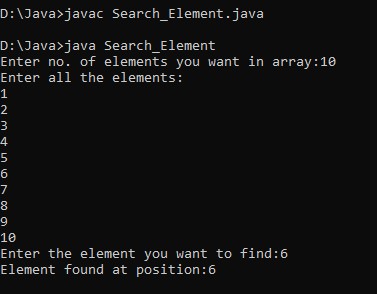
{

System.out.println("Element not found"); }

}

}

### OUTPUT



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: MCA B**  **Date:22/04/22** |

**Experiment No.: 7 Aim**

Perform string manipulations

### PROCEDURE

class Method\_string {

public static void main(String[] args) {

String greet = "Welcome to Java";

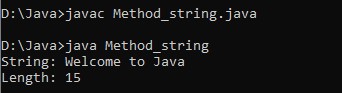
System.out.println("String: " + greet);

// get the length of greet int length = greet.length(); System.out.println("Length: " + length);

}

}

### OUTPUT



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:22/04/22** |

## Experiment No.: 8 Aim

Program to Sort strings

### PROCEDURE

import java.util.\*; public class SortStrings {

public static void main(String[] args){

String names[] = {

"Arun","Ram","Sunil","Cibin","Manu"};

System.out.println("The names' order before sorting : "); for (int i = 0; i < names.length; i++)

System.out.println(names[i]);

Arrays.sort(names);

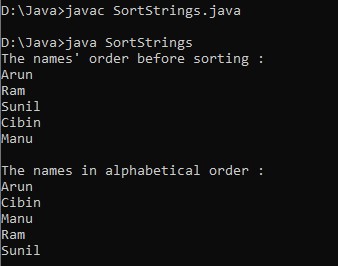
System.out.println("\nThe names in alphabetical order : "); for (int i = 0; i < names.length; i++)

System.out.println(names[i]);

}

}

### OUTPUT



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:22/04/22** |

## Experiment 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.

### PROCEDURE

import java.util.\*; public class Employe{ int eNo; String eName; float eSalary; public void empl(){

Scanner s = new Scanner(System.in);

System.out.print("Enter the Employee Number:"); eNo = s.nextInt();

System.out.print("Enter the Employee Name:"); eName = s.next();

System.out.print("Enter the Employee Salary:"); eSalary = s.nextFloat();

}

public void display(){

System.out.println(" Employee Number : " + eNo);

System.out.println(" Employee Name : " + eName);

System.out.println(" Employee Salary : " + eSalary);

} public static void main(String args[]) { int n;

Scanner sc = new Scanner(System.in);

System.out.print("Enter the number of Employees:"); n = sc.nextInt();

Employe obj[] = new Employe[n]; for(int i=0;i<n;i++){

obj[i] = new Employe(); obj[i].empl();

}

System.out.println(".............Employee Details ............ "); for(int i=0;i<n;i++)

{

obj[i].display();

}

int x;

System.out.println("enter the number to search an employee"); x=sc.nextInt(); int flag=0,i; for(i=0;i<n;i++)

{ if(obj[i].eNo==x)

{

flag=1;

break;

} else

{

flag=0;

}

} if(flag==1)

{

obj[i].display();

}

else

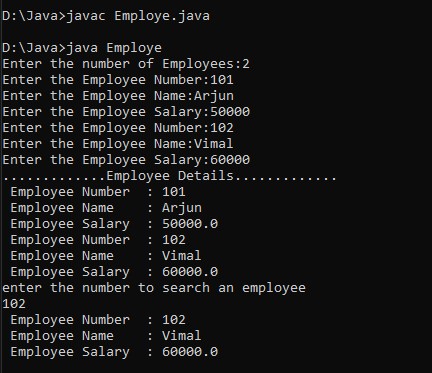
{

System.out.println("not found");

}

}}

### OUTPUT



Dept. of Computer Applications

|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: MCA B**  **Date:17-05-2022** |

**Experiment No.: 10**

**Aim**

Area of different shapes using overloaded functions .

## Procedure class Area

{ int shape(int l, int b)

{ return l\*b;

}

double shape(double l,double b)

{ return (0.5\*l\*b);

} double shape(double l)

{ return (3.14\*l\*l);

}

} public class MainArea

{ public static void main(String Args[])

{

Area A =new Area();

System.out.println("Area of rectangle=" + A.shape(4,5));

Amal Jyothi College Of Engineering,Kanjirappally 1 Dept. of Computer Applications

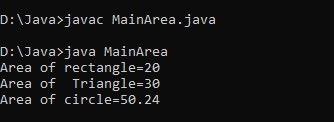
System.out.println("Area of Triangle=" + A.shape(6,5));

System.out.println("Area of circle=" + A.shape(4));

}

}

## Output Screenshot



Amal Jyothi College Of Engineering,Kanjirappally 2  Dept. of Computer Applications

|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: MCA - B**  **Date:17-04-2022** |

**Experiment 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.

## Procedure

public class Employee{ public static void main(String[] args) {

Teacher teacObj[] = new Teacher[2]; teacObj[0]=new Teacher("1","Aravind","srenilayam house",50000,"MCA","Computer Graphics"); teacObj[1] = new Teacher("2","Basker","Anars House",23000,"FT","Food Microbiology"); teacObj[0].display(); teacObj[1].display();

} }

class Employees {

String Empid;

String Name; String Address; int Salary;

Employees(String id,String name,String addr,int salary){ this.Empid = id; this.Name = name; this.Address = addr; this.Salary = salary;

} void display(){

Amal Jyothi College of Engineering, Kanjirappally

Dept. of Computer Applications

System.out.println("EmpID : " + this.Empid);

System.out.println("Name : " + this.Name);

System.out.println("Address : " + this.Address);

System.out.println("Salary : " + this.Salary); } }

class Teacher extends Employees{

String Department;

String Subject;

Teacher(String id,String name,String addr,int salary,String dept,String subj){ super(id,name,addr,salary); this.Department=dept; this.Subject=subj;

} void display(){

System.out.println(" ......................................... "); super.display();

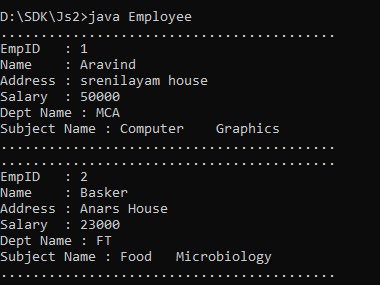
System.out.println("Dept Name : " + this.Department);

System.out.println("Subject Name : " + this.Subject);

System.out.println(" ......................................... "); }

}

## Output Screenshot



Amal Jyothi College of Engineering, Kanjirappally

|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: MCA B**  **Date:18/05/2022** |

**Experiment 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.

## Procedure

import java.util.\*; class Person{

String Name;

String Gender;

String Address; String Age; public Person(String Name,String Gender,String Address,String Age)

{

this.Name=Name; this.Gender=Gender; this.Address=Address; this.Age=Age;

}

}

1

class Employee extends Person

{

String Empid;

String Company\_Name;

String Qualification;

String Salary;

public Employee(String Name,String Gender,String Address,String Age ,String Empid,String Company\_Name, String Qualification,String Salary)

{

super(Name,Gender,Address,Age); this.Empid= Empid; this.Company\_Name=Company\_Name; this.Qualification=Qualification; this.Salary=Salary;

}

} class Teacher extends Employee

{

String Teacherid;

String Department; String Subject;

public Teacher(String Name,String Gender,String Address,String Age,String

Empid,String Company\_Name,String Qualification,String Salary,String Teacherid, String Department,String Subject)

{ super(Name,Gender,Address,Age,Empid,Name,Qualification, Salary); this.Teacherid=Teacherid; this.Department=Department;

2

this.Subject=Subject;

} public void read()

{ Scanner in =new Scanner(System.in);

System.out.println("enter the Name=");

Name=in.nextLine();

System.out.println("enter the Gender=");

Gender=in.nextLine();

System.out.println("enter the Address=");

Address=in.nextLine();

System.out.println("enter the Age=");

Age=in.nextLine();

System.out.println("enter the Employ id=");

Empid=in.nextLine();

System.out.println("enter the Company Name=");

Company\_Name=in.nextLine();

System.out.println("enter the Qualification=");

Qualification=in.nextLine();

System.out.println("enter the Salary=");

Salary=in.nextLine();

System.out.println("enter the Teacher id=");

Teacherid=in.nextLine();

System.out.println("enter the Department=");

Department=in.nextLine();

System.out.println("Enter the Subject=");

Subject=in.nextLine();

} public void display()

{ System.out.println(" Employee Details ");

3

System.out.println("Name="+ Name);

System.out.println("Gender=" + Gender);

System.out.println("Address=" + Address);

System.out.println("Age=" + Age);

System.out.println("Empid=" + Empid);

System.out.println("Company Name=" + Company\_Name);

System.out.println("Qualification=" + Qualification);

System.out.println("Salary=" + Salary);

System.out.println("Teacher id=" + Teacherid);

System.out.println("Department=" + Department);

System.out.println("Subject=" + Subject);

System.out.println("+++++++++++++++++++++++++++++++++++++++++++");

}

}

public class InheritancePerson {

public static void main(String Args[])

{ int i,n;

Scanner in =new Scanner(System.in);

System.out.println("Enter the Number of employee="); n=in.nextInt();

Teacher T[] = new Teacher[n]; for(i=0;i<n;i++)

{

T[i]=new

Teacher("Name","Gender","Address","Age","Empid","Name","Qualification","Salary","

Teacherid","Department","Subject");

T[i].read();

}

4

for(i=0;i<n;i++)

{

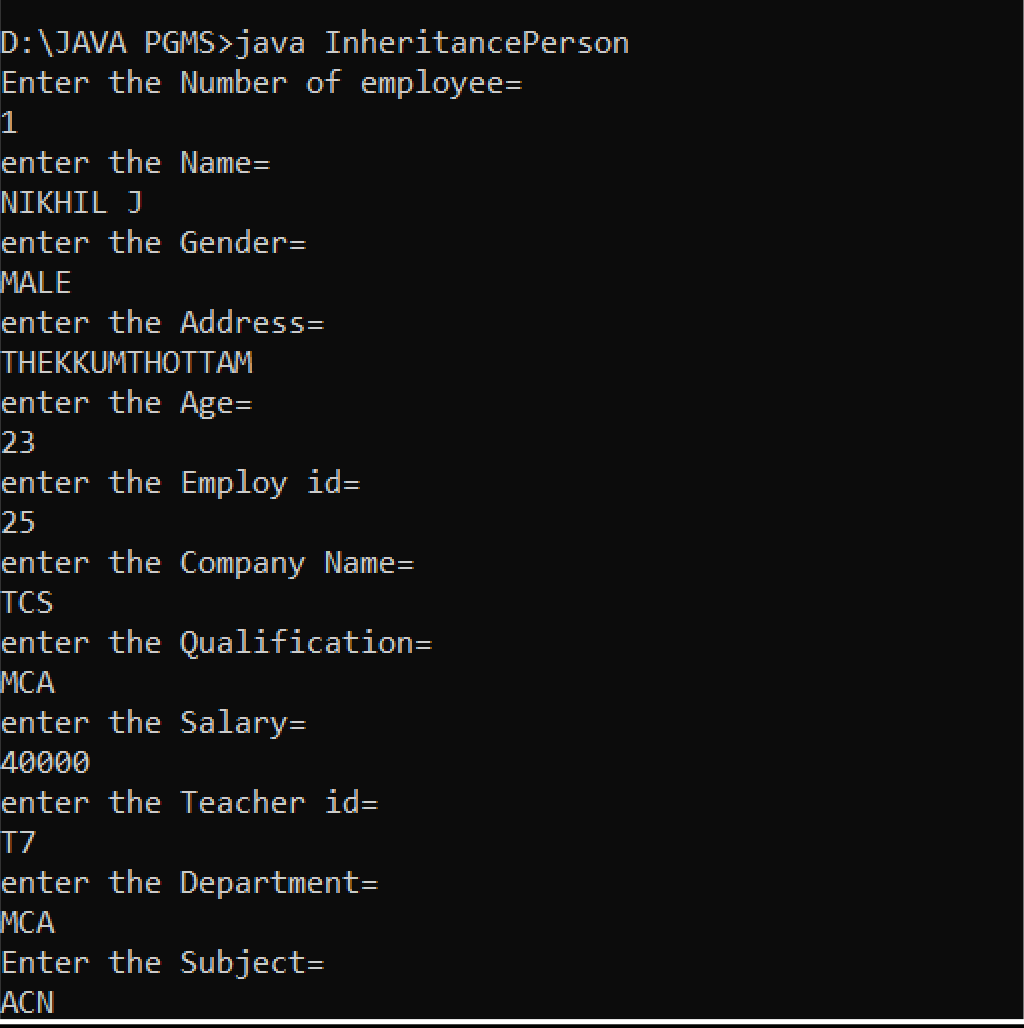
T[i].display();

}

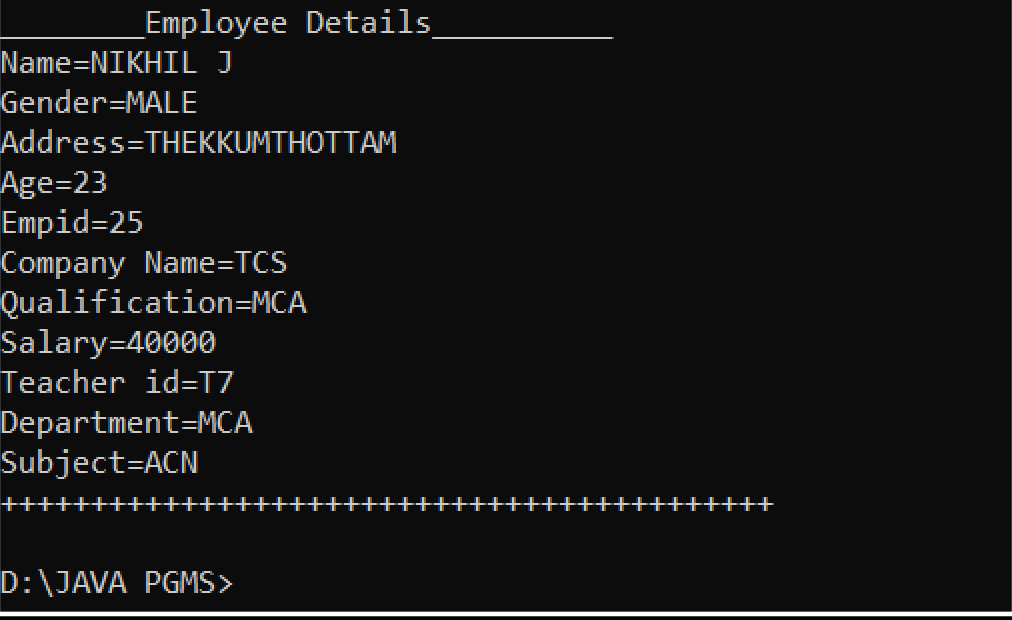
}

}

## Output Screenshot



5



6

|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: MCA B**  **Date:18/05/2022** |

**Experiment 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. **Procedure**

import java.util.Scanner;

class Publisher{ int publisher\_id;

String publisher\_name;

Publisher(int publisher\_id, String publisher\_name){ this.publisher\_id= publisher\_id; this.publisher\_name= publisher\_name;

}

}

class Book extends Publisher{

int book\_id;

String book\_name;

Book(int publisher\_id, String publisher\_name, int book\_id, String book\_name) { super(publisher\_id, publisher\_name); this.book\_id= book\_id;

1

this.book\_name= book\_name;

}

}

class Literature extends Book{

int literature\_id;

String literature\_theme;

Literature(int publisher\_id, String publisher\_name, int book\_id, String book\_name, int literature\_id, String literature\_theme) { super(publisher\_id, publisher\_name, book\_id, book\_name); this.literature\_id= literature\_id; this.literature\_theme= literature\_theme;

}

void displayDetails() {

System.out.println("The publisher ID of the book is: " + this.publisher\_id);

System.out.println("The publisher name of the book is: " + this.publisher\_name);

System.out.println("The Book ID of the book is: " + this.book\_id);

System.out.println("The Book name of the book is: " + this.book\_name);

System.out.println("The Literature ID of the book is: " + this.literature\_id);

System.out.println("The Literature theme of the book is: " + this.literature\_theme);

}

}

class Fiction extends Book{

int fiction\_id;

2

String fiction\_theme;

Fiction(int publisher\_id, String publisher\_name, int book\_id, String book\_name, int fiction\_id, String fiction\_theme) { super(publisher\_id, publisher\_name, book\_id, book\_name); this.fiction\_id= fiction\_id; this.fiction\_theme= fiction\_theme;

}

void displayDetails() {

System.out.println("The publisher ID of the book is: " + this.publisher\_id);

System.out.println("The publisher name of the book is: " + this.publisher\_name);

System.out.println("The Book ID of the book is: " + this.book\_id);

System.out.println("The Book name of the book is: " + this.book\_name);

System.out.println("The Fiction ID of the book is: " + this.fiction\_id);

System.out.println("The Fiction theme of the book is: " + this.fiction\_theme);

}

} public class BookInheritance { public static void main(String[] args) {

Literature literature= new Literature(10,"Robert Kiyozaki",200,"Rich Dad Poor Dad",2001,"Drama");

Fiction fiction= new Fiction(101, "F. Scott Fitzgerald", 301, "The Great Gatsby",

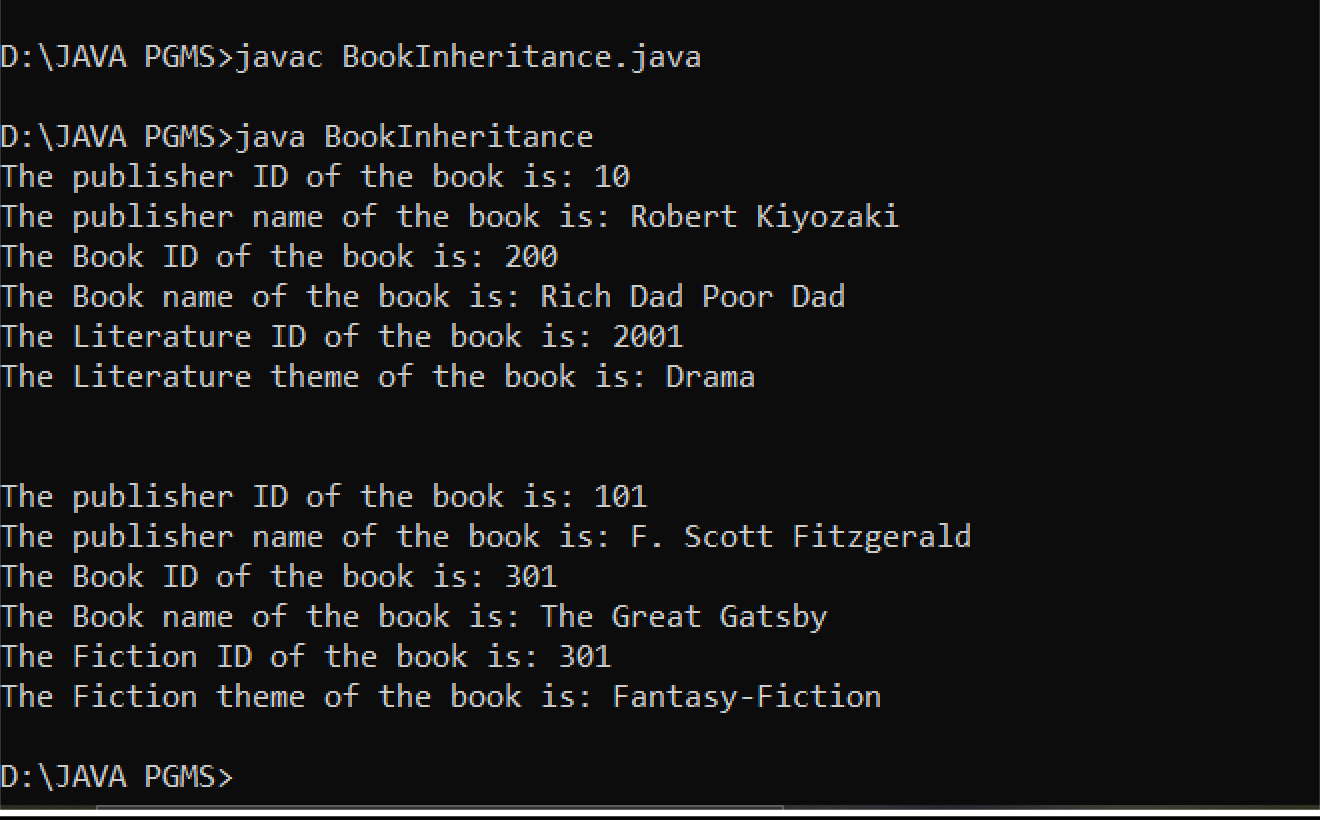
301, "Fantasy-Fiction"); literature.displayDetails(); System.out.println("\n"); fiction.displayDetails();

}

}

3

## Output Screenshot



4

|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:24/05/22** |

**Experiment 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.

### PROCEDURE

import java.util.Scanner;

class student{ int roll;

String name;

int phy,eng,maths;

student()

{

Scanner sc1= new Scanner(System.in); System.out.println("Enter the roll number:"); roll =sc1.nextInt();

System.out.println("Enter name:"); name=sc1.next();

System.out.println("Enter physics mark:"); phy =sc1.nextInt();

System.out.println("Enter english mark:"); eng =sc1.nextInt();

System.out.println("Enter maths mark:"); maths =sc1.nextInt();

}

}

class sports extends student

{

int fscore,cscore; sports()

{

Scanner sc2= new Scanner(System.in);

System.out.println("Enter football score:"); fscore=sc2.nextInt();

System.out.println("Enter Cricket score:"); cscore=sc2.nextInt();

}

}

class Result extends sports

{ void display()

{

System.out.println("Academic Details"+"\n"+" ");

System.out.println("Name : " + name);

System.out.println("Roll No : " + roll);

System.out.println("");

System.out.println("MARKS" +"\n" + " ");

System.out.println("Physics :" + phy); System.out.println("English :" + eng);

System.out.println("Maths :" + maths);

System.out.println("Total subject mark:"+(phy+eng+maths)); System.out.println("");

System.out.println("SPORTS SCORE" +"\n"+" ");

System.out.println("Football : " + fscore);

System.out.println("Cricket : " + cscore);

System.out.println("Total Sports mark:"+(fscore+cscore));

}

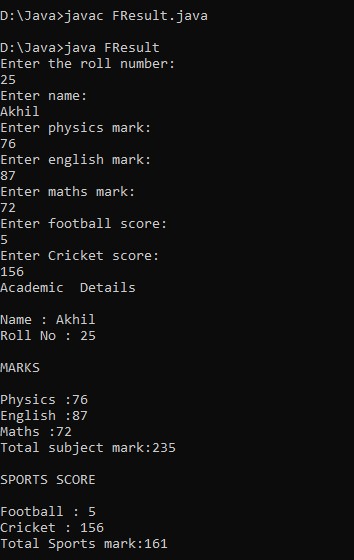
}

public class FResult{ public static void main(String[] args) { Result rs =new Result(); rs.display();

}

}

### OUTPUT



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: MCA B**  **Date:24/05/2022** |

**Experiment 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.

## Procedure

import java.util.\*; import java.lang.\*;

interface Shape { float pi = 3.14F; float area(); float perimeter();

}

class Circle implements Shape { Scanner sc = new Scanner(System.in); int r;

public float area() {

System.out.print("Enter the radius : "); r = Integer.parseInt(sc.nextLine()); return (pi \* r \* r);

}

public float perimeter() {

System.out.print("Enter the radius : "); r = Integer.parseInt(sc.nextLine());

1

return (2 \* pi \* r);

}

}

class Rectangle implements Shape { Scanner sc = new Scanner(System.in); int l, b;

public float area() {

System.out.print("Enter the Length : "); l = Integer.parseInt(sc.nextLine()); System.out.print("Enter the breadth : "); b = Integer.parseInt(sc.nextLine()); return (l \* b);

}

public float perimeter() {

System.out.print("Enter the Length : "); l = Integer.parseInt(sc.nextLine()); System.out.print("Enter the breadth : "); b = Integer.parseInt(sc.nextLine()); return (2 \* (l + b));

}

}

class ShapeInterface { public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

Circle c = new Circle();

2

Rectangle r = new Rectangle(); int ch; while (true) {

System.out.println("1:Area of Circle");

System.out.println("2:Perimeter of Circle");

System.out.println("3:Area of Rectangle");

System.out.println("4:Perimter of Rectangle");

System.out.println("5:EXIT"); System.out.print("Enter choice : "); ch = Integer.parseInt(sc.nextLine()); switch (ch) { case 1:

float ar = c.area();

System.out.println("Area :" + ar);

System.out.println("\*\*-----------\*\* --------- \*\*");

break;

case 2:

float pr = c.perimeter();

System.out.println("Perimeter of Circle = "+pr);

System.out.println("\*\*-----------\*\* --------- \*\*");

break;

case 3:

float a = r.area();

System.out.println("Area :" + a);

System.out.println("\*\*-----------\*\* --------- \*\*");

break;

case 4:

float pr1 = r.perimeter();

System.out.println("Perimeter of Rectangle = "+pr1);

3

System.out.println("\*\*-----------\*\* --------- \*\*");

break;

case 5:

System.out.println("Exiting the Program!!!!!");

System.exit(0); default:

System.out.println("invalid!"); }

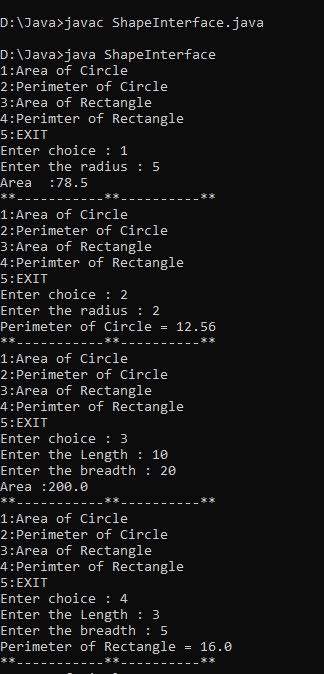
}

}

}

4

## Output Screenshot



5

20MCA1342– Dept. of Computer Applications

|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:24/05/22** |

### Experiment No.: 16 Aim

Prepare bill with the given format using calculate method from interface.

### PROCEDURE

import java.text.SimpleDateFormat; import java.util.Date; interface bill

{ void cal();

}

class details1 implements bill{ int pid=101,q=2,uprice=25,t1; String name1="A"; public void cal()

{ t1=q\*uprice;

} }

class details2 extends details1 { int pid2=102,q2=1,uprice2=100,t2;

String name2="B";

SimpleDateFormat f=new SimpleDateFormat("dd/MM/yy");

Date d= new Date(); public void cal()

{ super.cal(); t2=q2\*uprice2;

}

20MCA1342– Dept. of Computer Applications

public void display()

{

System.out.println("Order No.384\n");

System.out.println("Date: "+f.format(d));

System.out.println("\nProduct Id\tName\t\tQuantity\tunit price\tTotal");

System.out.println(" ");

System.out.println(pid+"\t\t"+name1+"\t\t"+q+"\t\t"+uprice+"\t\t"+t1);

System.out.println(pid2+"\t\t"+name2+"\t\t"+q2+"\t\t"+uprice2+"\t\t"+t2);

System.out.println(" ");

System.out.println("\t\t\t\t\tNet.Amount"+"\t\t"+(t1+t2));

}

}

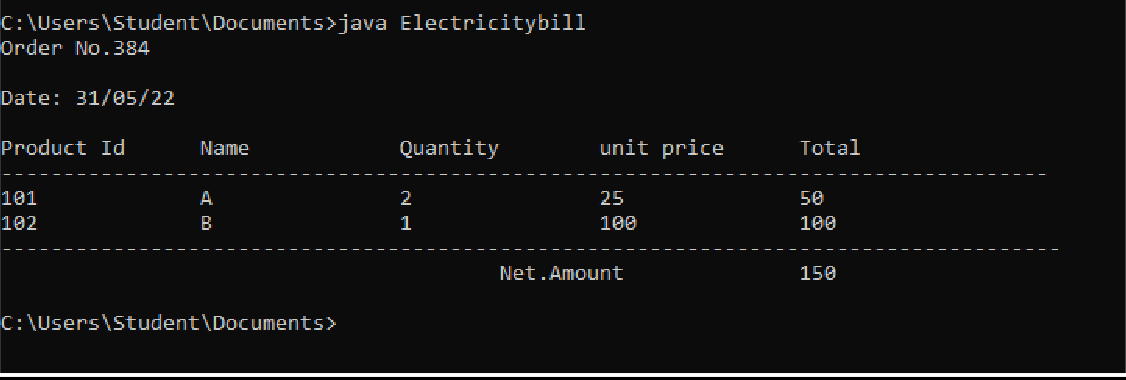
public class Electricitybill{ public static void main(String[] args) {

details2 obj2=new details2(); obj2.cal(); obj2.display();

}

}

### OUTPUT



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: MCA B**  **Date: 31/05/2022** |

**Experiment 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.

**Procedure package\_graphics.java** package package\_graphics;

interface interface\_graphics{ public float recArea(int l, int h); public float cirArea(int r); public float squArea(int a); public float triArea(int l, int h);

}

public class package\_graphics implements interface\_graphics {

public float recArea(int l, int h){ return l\*h;

} public float cirArea(int r){ return r\*r\*(float)3.14;

} public float squArea(int a){ return a\*a;

} public float triArea(int l, int h){

return l\*h\*(float)(.5);

}

}

## main\_graphics.java

import package\_graphics.\*; import java.util.\*; public class main\_graphics { public static void main(String []args){ package\_graphics testObj = new package\_graphics();

int l,h,r,a,c,d;

Scanner s=new Scanner(System.in);

System.out.println("Enter the length for rectangle"); l=s.nextInt();

System.out.println("Enter the breadth for rectangle"); h=s.nextInt();

System.out.println("Enter the radius of circle"); r=s.nextInt();

System.out.println("Enter the side for Square"); a=s.nextInt();

System.out.println("Enter the breadth for triangle"); c=s.nextInt();

System.out.println("Enter the height for triangle"); d=s.nextInt();

System.out.println(testObj.recArea(l,h)); System.out.println(testObj.cirArea(r));

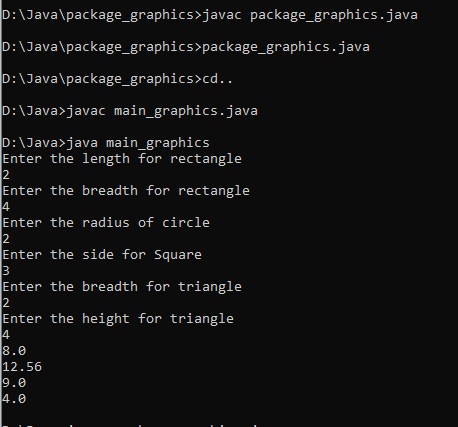
System.out.println(testObj.squArea(a));

System.out.println(testObj.triArea(c,d));

}

}

### Output Screenshot



#### Experiment No.: 18 Aim

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

#### PROCEDURE

import java.util.Scanner;

class authException extends Exception

{

public authException(String s) {

super(s);

}

|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:31/05/22** |

}

public class Userauthentication

{ public static void main(String[] args) {

String username = "student";

String passcode = "student123";

String user\_name,password; Scanner sc = new Scanner(System.in); try

{

System.out.println("Enter the username:"); user\_name = sc.nextLine();

System.out.println("Enter the password:"); password = sc.nextLine();

if(username.equals(user\_name) && passcode.equals(password))

{

System.out.println("Authentication successful...");

} else

throw new authException("Invalid user credentials");

}

catch(authException e)

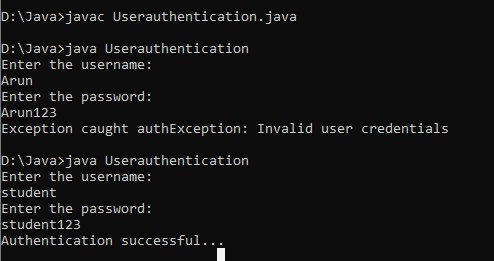
{

System.out.println("Exception caught "+e); }

}

}

#### OUTPUT



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:31/05/22** |

#### Experiment No.: 19 Aim

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

#### PROCEDURE

import java.util.Scanner; class NegException extends Exception

{ public NegException(String s)

{ super(s);

}

}

public class Average {

public static void main(String[] args)

{ int i;

double sum=0,avg=0;

Scanner sc=new Scanner(System.in); System.out.println("Enter n numbers:"); int n=sc.nextInt();

for(i=1;i<=n;i++)

{ try

{

System.out.println("Enter number"+i); int a=sc.nextInt();

if(a<0)

{

i--;

throw new NegException("Negative numbers not allowed,Try again");

} else

{

sum=sum+a;

} }

catch(NegException e)

{

System.out.println("NEGETIVE EXCEPTION OCCURED:"+e);

}

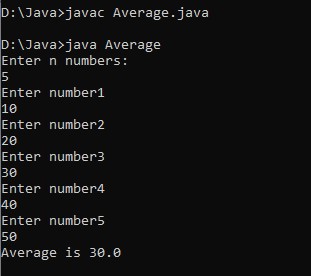
} avg=sum/n;

System.out.println("Average is "+avg); sc.close();

}

}

#### OUTPUT



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:07/06/22** |

**20**

#### Aim

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

#### PROCEDURE

import java.util.Scanner;

class Fib extends Thread{ int f,n1=0,n2=1,n3;

Fib(int c){ this.f=c;

}

public void run(){

System.out.println("fib is "+n1); System.out.println("fib is "+n2);

for(int i=2;i<this.f;++i) {

n3=n1+n2;

System.out.println("fib is "+n3);

n1=n2; n2=n3;

}

}

}

class even extends Thread{ int range; even(int range){ this.range=range;

}

public void run(){ for(int i=0;i<this.range;i++){ if(i%2==0){

System.out.println("even num is "+i); }

}

}

}

public class mulThread { public static void main(String [] args){ int c,range;

Scanner sc=new Scanner(System.in); System.out.println("enter the count of Fibinooci"); c=sc.nextInt(); Fib fi=new Fib(c);

System.out.println("enter the range of even number"); range=sc.nextInt();

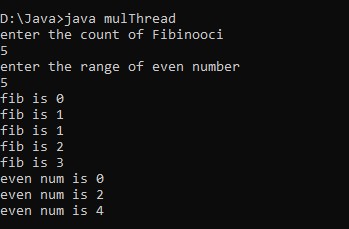
even ev = new even(range);

fi.start(); ev.start();

}

}

#### OUTPUT



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:07/06/22** |

**21**

**Aim**

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

#### PROCEDURE

class Stack { private int arr[]; private int top; private int capacity;

Stack(int size) { arr = new int[size]; capacity = size; top = -1; } public void push(int x) {

if (isFull()) {

System.out.println("Stack OverFlow");

System.exit(1);

}

System.out.println("Inserting " + x); arr[++top] = x;

}

public int pop() {

if (isEmpty()) {

System.out.println("STACK EMPTY");

System.exit(1);

} return arr[top--];

}

public int getSize() {

return top + 1;

}

public Boolean isEmpty() {

return top == -1;

}

public Boolean isFull() {

return top == capacity - 1;

}

public void printStack() {

for (int i = 0; i <= top; i++) {

System.out.print(arr[i] + "\t");

}

}

public static void main(String[] args) { Stack stack = new Stack(5); stack.push(1); stack.push(2); stack.push(3);

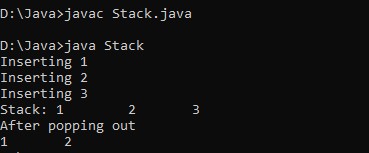
System.out.print("Stack: "); stack.printStack(); stack.pop();

System.out.println("\nAfter popping out"); stack.printStack();

}

}

#### OUTPUT



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:07/06/22** |

**22**

#### Aim

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

.

#### PROCEDURE

import java.util.\*;

public class arraylist{ public static void main(String[] args) {

ArrayList<String> arrayList= new ArrayList<>(); arrayList.add("Bibin"); arrayList.add("Rony"); arrayList.add("Tarun"); arrayList.add("Jack");

System.out.println("The elements of the arraylist is - "+arrayList);

Collections.sort(arrayList);

System.out.println("\nThe ArrayList Sort : "+arrayList); // ArrayList Sort

Collections.addAll(arrayList,"Karun","Vimal","Shan","Ram","Gibin");

System.out.println("\nAdding new items in the arraylist is : "+arrayList); // ArrayList AddAll

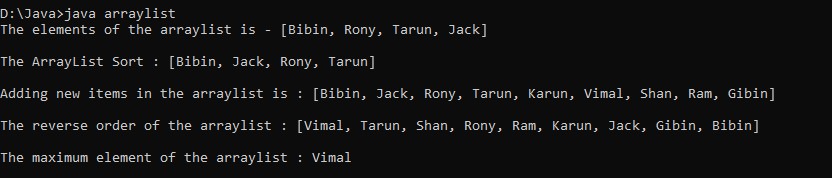
Collections.sort(arrayList, Collections.reverseOrder()); //Arraylist in reverse order

System.out.println("\nThe reverse order of the arraylist : "+arrayList);

System.out.println("\nThe maximum element of the arraylist : "+Collections.max(arrayList)); //Max elements in the arraylist }

}

#### OUTPUT



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:07/06/22** |

**Experiment No.: 23**

### Aim

Program to demonstrate the creation of queue object using the Priority Queue class.

.

### PROCEDURE

import java.util.\*;

class Collection\_Framework\_Queue { public static void main(String args[]) {

Queue<Integer> q = new PriorityQueue<Integer>(new Comp()); int ch;

Scanner sc = new Scanner(System.in); do {

System.out.println("\n1.ADD\n2.PEEK\n3.POLL or

REMOVE\n4.DISPLAY\n5.EXIT");

System.out.println("Enter your choice : "); ch = sc.nextInt(); switch (ch) { case 1:

System.out.println("\n\tEnter Integer : "); int n1 = sc.nextInt(); q.add(n1);

System.out.println("\n\tADDED SUCCESSFULLY ! ! ! "); break;

case 2:

if (q.isEmpty()) {

System.out.print("\n\tQueue Empty ! ! !");

} else {

System.out.print("\n\tPeeked element is " + q.peek());

} break;

case 3:

if (!q.isEmpty()) {

System.out.print("\n\tRemoved element is " + q.poll());

} else {

System.out.print("\n\tQueue Empty ! ! !");

} break;

case 4:

if (!q.isEmpty()) {

System.out.print("\nSize of queue : " + q.size());

System.out.print("\nQueue elements : " + q); System.out.println("\nQueue elements are"); for (int i : q) {

System.out.println(i);

}

} else {

System.out.print("\n\tQueue Empty ! ! !");

} break;

case 5:

break;

default:

System.out.println("\n\tPlease enter valid choice ! ! ! ");

}

} while (ch != 5);

}

}

#### class Comp implements Comparator<Integer> {

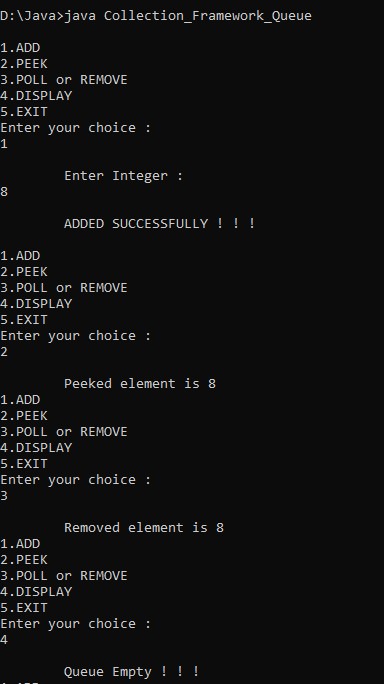
public int compare(Integer a, Integer b) {

return a % 10 > b % 10 ? 1 : -1;

}

}

##### OUTPUT



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:07/06/22** |

**Experiment No.: 24**

**Aim**

Program to demonstrate the addition and deletion of elements in deque.

### PROCEDURE

import java.util.\*;

class deque

{ public static void main(String[] args) {

Deque<String> deque = new LinkedList<String>();

deque.add("Java");

deque.addFirst("Python");

deque.addLast("Datastructure");

deque.push("Web-programming");

deque.offer("Networking");

deque.offerFirst("DBMS");

System.out.println(deque + "\n");

deque.removeFirst();

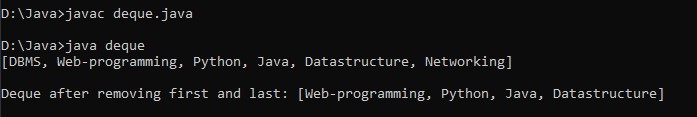
deque.removeLast();

#### System.out.println("Deque after removing " + "first and last: " + deque);

}

}

##### OUTPUT



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:07/06/22** |

**Experiment No.: 25**

**Aim**

Write a Java program to compare two hash set

### Procedure

import java.util.\*;

public class CompareHash { public static void main(String[] args) {

HashSet<String> h\_set = new HashSet<String>(); h\_set.add("Red");

h\_set.add("Green"); h\_set.add("Black"); h\_set.add("White");

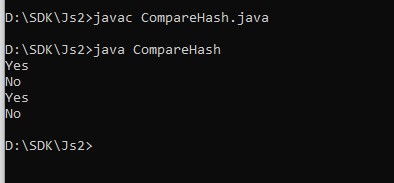
HashSet<String>h\_set2 = new HashSet<String>(); h\_set2.add("Red"); h\_set2.add("Pink"); h\_set2.add("Black"); h\_set2.add("Orange");

HashSet<String>result\_set = new HashSet<String>(); for (String element : h\_set){

System.out.println(h\_set2.contains(element) ? "Yes" : "No"); }

}}

**Output Screenshot**



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:07/06/22** |

**Experiment No.: 26**

### Aim

Program to demonstrate the working of Map interface by adding, changing and removing

Elements.

### Procedure

import java.util.\*; class HashMapDemo {

public static void main(String args[]) {

Map<String, Integer> hm = new HashMap<String, Integer>(); hm.put("Anu", new Integer(1)); hm.put("sinu", new Integer(2)); hm.put("Jinu", new Integer(3));

for (Map.Entry<String, Integer> me : hm.entrySet()) {

System.out.print(me.getKey() + ":");

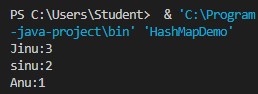
System.out.println(me.getValue());

}

}

}

### Output Screenshot



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:09/06/22** |

**Experiment No.: 27**

### Aim

Program to find maximum of three numbers using AWT. **Procedure**

import java.awt.\*; import java.awt.event.\*;

public class largenum implements ActionListener{

Frame f=new Frame();

Label l1=new Label("First Number");

Label l2=new Label("Second Number");

Label l3=new Label("Third Number");

Label res=new Label("Result");

TextField t1=new TextField();

TextField t2=new TextField();

TextField t3=new TextField();

Button b1=new Button("Largest !");

largenum(){ l1.setBounds(50,100,100,20); l2.setBounds(50,140,100,20); l3.setBounds(50,180,100,20); t1.setBounds(150,100,100,20); t2.setBounds(150,140,100,20); t3.setBounds(150,180,100,20); b1.setBounds(50,220,100,20); res.setBounds(50,260,100,20);

f.add(l1);

f.add(l2);

f.add(l3);

f.add(t1);

f.add(t2);

f.add(t3);

f.add(res);

f.add(b1);

b1.addActionListener(this);

f.setLayout(null);

f.setVisible(true);

f.setSize(400,400);

}

public static void main(String[] args){ new largenum();

}

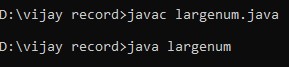
public void actionPerformed(ActionEvent e){ if(e.getSource()==b1){ int n1=Integer.parseInt(t1.getText()); int n2=Integer.parseInt(t2.getText()); int n3=Integer.parseInt(t3.getText());

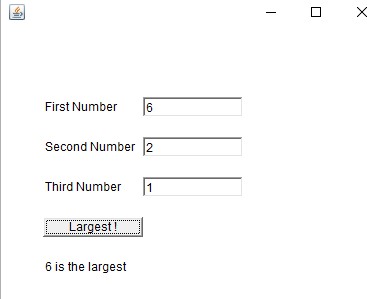
int largeres= (n1 > n2) ? (n1 > n3 ? n1 : n3) : (n2 > n3 ? n2 : n3); res.setText(String.valueOf(largeres)+" is the largest"); }

}

}

### Output Screenshot





|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:09/06/22** |

### Experiment No.: 28

### Aim

Implement a simple calculator using AWT components. **Procedure**

import java.awt.\*; import java.awt.event.\*;

public class Calculator implements ActionListener {

Frame f=new Frame();

Label l1=new Label("First Number");

Label l2=new Label("Second Number");

Label l3=new Label("Result");

TextField t1=new TextField();

TextField t2=new TextField();

TextField t3=new TextField();

Button b1=new Button("Add");

Button b2=new Button("Sub");

Button b3=new Button("Mul");

Button b4=new Button("Div");

Button b5=new Button("Cancel");

Calculator()

{

l1.setBounds(50,100,100,20); l2.setBounds(50,140,100,20); l3.setBounds(50,180,100,20); t1.setBounds(200,100,100,20); t2.setBounds(200,140,100,20); t3.setBounds(200,180,100,20); b1.setBounds(50,250,50,20); b2.setBounds(110,250,50,20); b3.setBounds(170,250,50,20); b4.setBounds(230,250,50,20); b5.setBounds(290,250,50,20);

f.add(l1);

f.add(l2);

f.add(l3);

f.add(t1);

f.add(t2);

f.add(t3);

f.add(b1);

f.add(b2);

f.add(b3);

f.add(b4);

f.add(b5); b1.addActionListener(this); b2.addActionListener(this); b3.addActionListener(this); b4.addActionListener(this); b5.addActionListener(this); f.setLayout(null);

f.setVisible(true);

f.setSize(400,350); }

public void actionPerformed(ActionEvent e)

{ int n1=Integer.parseInt(t1.getText()); int n2=Integer.parseInt(t2.getText()); if(e.getSource()==b1)

{

t3.setText(String.valueOf(n1+n2));

}

if(e.getSource()==b2)

{

t3.setText(String.valueOf(n1-n2));

}

if(e.getSource()==b3)

{

t3.setText(String.valueOf(n1\*n2));

}

if(e.getSource()==b4)

{

t3.setText(String.valueOf(n1/n2));

}

if(e.getSource()==b5)

{

System.exit(0);

} }

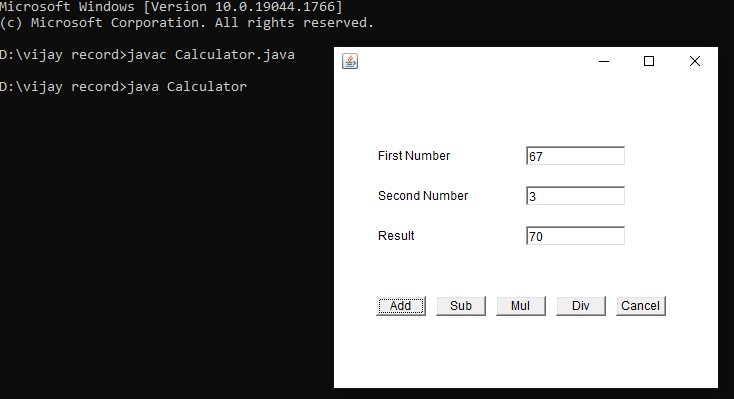
public static void main(String...s)

{ new Calculator();

}

}

### Output Screenshot



**Experiment No.: 29**

**Date:29/05/22**

### Aim

Develop a program to handle all mouse events and window events **Procedure**

import java.awt.\*; import java.awt.event.\*;

public class Mouseevents extends Frame implements MouseListener{ Label l;

Mouseevents(){ addMouseListener(this);

l=new Label();

l.setBounds(20,50,100,20); add(l); setSize(300,300); setLayout(null); setVisible(true);

}

public void mouseClicked(MouseEvent e) {

l.setText("Mouse Clicked"); }

public void mouseEntered(MouseEvent e) {

l.setText("Mouse Entered"); }

public void mouseExited(MouseEvent e) {

l.setText("Mouse Exited"); }

public void mousePressed(MouseEvent e) {

l.setText("Mouse Pressed"); }

public void mouseReleased(MouseEvent e) {

l.setText("Mouse Released");

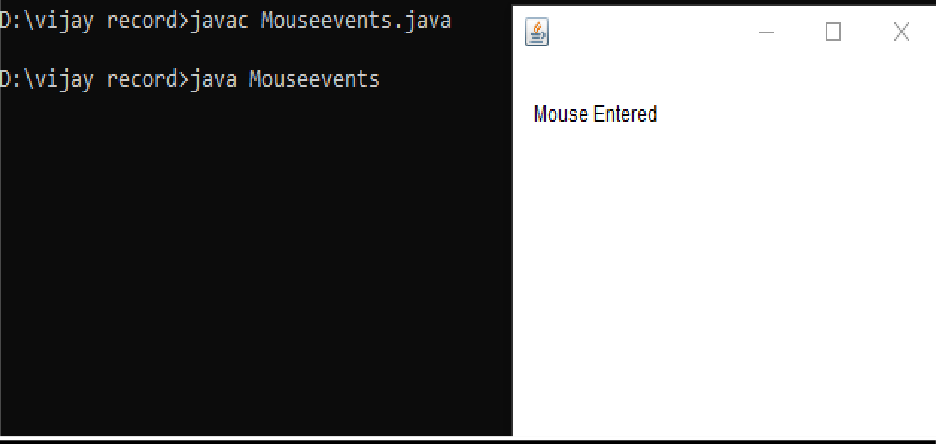
} public static void main(String[] args) {

new Mouseevents();

}

}

### Output Screenshot



**Experiment No.: 30**

### Aim

Develop a program to handle Key events.

**Procedure Date:29/05/22** import java.awt.FlowLayout; import java.awt.Frame; import java.awt.Label; import java.awt.TextField; import java.awt.event.KeyEvent; import java.awt.event.KeyListener;

public class KE implements KeyListener

{

Label lb1, lbl2, lb;

TextField tf1;

Frame fr;

String s;

KE()

{

fr = new Frame("KeyEventListener Example");

lb1= new Label(" Key Events will be displayed based on the actions",

Label.CENTER);

lbl2= new Label(); lb= new Label();

tf1 = new TextField(20); fr.setLayout(new FlowLayout());

fr.add(lb1);

fr.add(tf1);

fr.add(lbl2); tf1.addKeyListener(this); fr.setSize(460,250);

fr.setVisible(true);

}

public void keyPressed(KeyEvent ev) {

lbl2.setText(" Key pressed");

}

public void keyReleased(KeyEvent ev)

{ lbl2.setText("Released");

}

public void keyTyped(KeyEvent ev)

{ lbl2.setText("Key is typed");

fr.setVisible(true);

}

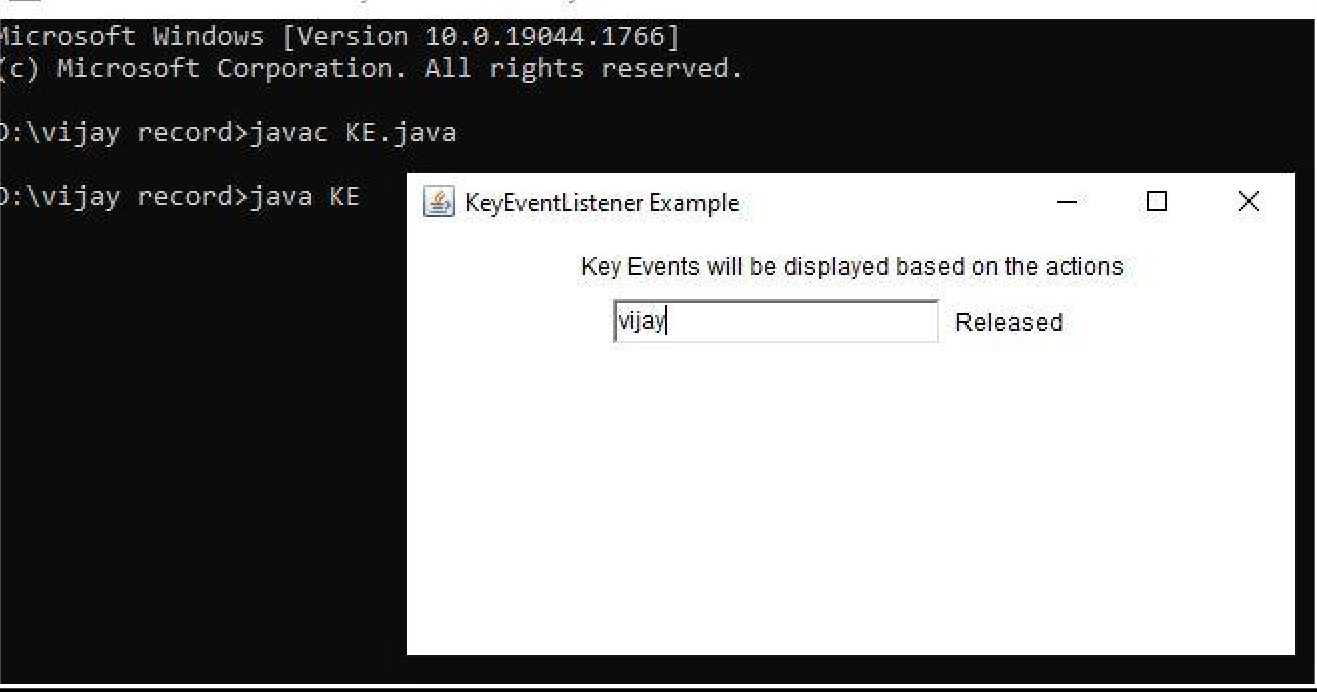
public static void main(String[] args)

{ new KE();

}

}

### Output Screenshot



**Experiment No.: 31**

**Date:30/05/22**

**Aim**

Write a program to write to a file, then read from the file and display the contents on the console.

### Procedure

import java.io.FileReader; import java.io.FileWriter; import java.io.IOException; import java.io.\*; import java.util.\*; import java.io.File; class Read { public static void main(String[] args) {

String var = "";

Scanner scan = new Scanner(System.in);

System.out.println("Enter the text to create file : type ENTER key 3 times to stop"); while (!var.endsWith("\n\n\n")) var = var + scan.nextLine() + "\n"; try {

File file = new File("output.txt"); FileWriter fw = new FileWriter(file); fw.write(var); fw.close();

System.out.println("Reading File content");

FileReader fr = new FileReader("output.txt");

String str = "";

int i;

while ((i = fr.read()) != -1) { str += (char) i;

}

System.out.println(str); fr.close();

} catch (IOException e) {

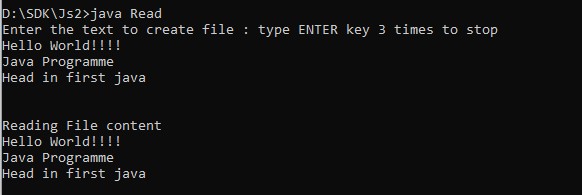
System.out.println("There are some exception");

}

}

}

### Output Screenshot



**Experiment No.: 32**

### Aim

**Date:30/05/22**

Write a program to copy one file to another

### Procedure

import java.io.FileReader; import java.io.FileWriter; import java.io.IOException; import java.io.\*; import java.util.\*; import java.io.File; public class Copy {

public static void main(String[] args) {

Scanner scan=new Scanner(System.in);

System.out.println("Enter the source File Name"); String source=scan.nextLine(); try {

FileReader fr=new FileReader(source); String str = ""; int i;

System.out.println("Reading from file "+source); while ((i = fr.read()) != -1) { str += (char) i;

}

System.out.println(str);

System.out.println("\n Enter the filename to copy");

String destination=scan.nextLine();

File file=new File(destination); FileWriter fw = new FileWriter(file); fw.write(str); fr.close();

fw.close();

System.out.println("Copied from "+source+" to "+destination+ " Successfully..!");

} catch (Exception e) {

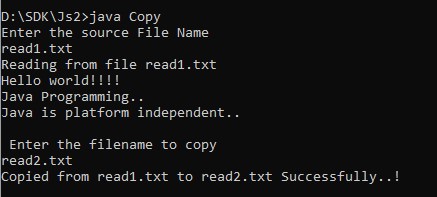
System.out.println("Exception Occured");

}

}

}

### Output Screenshot



|  |
| --- |
| **Name: Nikhil Jais**  **Roll No:25**  **Batch: B**  **Date:30/05/22** |

### Experiment No.: 33

### Aim

Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

### Procedure

import java.io.FileReader; import java.io.FileWriter; import java.io.IOException; import java.io.\*; import java.util.\*; import java.io.File; public class OddEven {

public static void main(String[] args) { try {

FileReader fr = new FileReader("number.txt");

BufferedReader br = new BufferedReader(fr);

File file1 = new File("oddnumber.txt");

FileWriter fw1 = new FileWriter(file1);

File file2 = new File("evennumber.txt");

FileWriter fw2 = new FileWriter(file2); String num;

while ((num = br.readLine()) != null) { if (Integer.parseInt(num) % 2 == 0) { fw2.write(num + "\n");

} else {

fw1.write(num + "\n");

} } fw1.close(); fw2.close(); } catch (Exception e) {

System.out.println("Error");

}

}

}

### Output Screenshot

