

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

BJ KEERTANA (1BM23CS059)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

Sep-2024 to Jan-2025

**B.M.S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)**
Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled “Object Oriented Java Programming (23CS3PCOOJ)” carried out by **BJ KEERTANA(1BM23CS059)**, who is a bonafide student of **B.M.S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Lab faculty Incharge Swathi Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
---	---

Index

Sl. No.	Date	Experiment Title	Page No.
1	01-10-2024	Quadratic Equation Implementation	4-7
2	08-10-2024	Calculate SGPA of student	8-12
3	15-10-2024	Class Book, <code>toString()</code> , array of objects	13-16
4	22-10-2024	Abstract class implementation, finding area	17-20
5	29-10-2024	Class Bank, Savings account and current account	21-27
6	12-11-2024	Packages, CIE,SEE, calculating total marks	28-32
7	26-11-2024	Interfaces	33-36
8	26-11-2024	Exception handling	37-40
9	3-12-2024	Threads	41-43
10	3-12-2024	User interface	44-47

Github Link:

<https://github.com/keertanabj/OOJ/tree/main>

Program 1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a , b , c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

Algorithm:

1) Output:
3) ~~Program~~ Enter two nos:
2
4
Sum: 6
Difference: -2
2) Output:
Enter two nos:
5
7
Product: 35
3) Program that prints all real solutions to quadratic equation $ax^2 + bx + c = 0$. Read a, b, c & use quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating there are no real solutions.

```
import java.util.*;  
class Quadratic  
{  
    public static void main (String args[]){  
        double a, b, c, d, r1, r2;  
        System.out.println ("Enter coefficients a, b, c ");  
        Scanner sc = new Scanner (System.in);  
        a = sc.nextDouble();  
        b = sc.nextDouble();  
        c = sc.nextDouble();  
        d = b * b - 4 * a * c;  
        if (d < 0)  
            System.out.println ("No real roots");  
        else  
            r1 = (-b + Math.sqrt(d)) / (2 * a);  
            r2 = (-b - Math.sqrt(d)) / (2 * a);  
            System.out.println ("Roots are " + r1 + " and " + r2);  
    }  
}
```

```

a = 1c. next Double ();
b = 1c. next Double ();
c = 1c. next Double ();
d = b*b - 4*a*c; // calculate discriminant
if (d > 0.0) // calculate square root of discriminant
    double r1 = (-b + Math.sqrt(d)) / (2.0*a);
    double r2 = (-b - Math.sqrt(d)) / (2.0*a);
    System.out.println ("Root are " + r1 + " and " + r2);
else if (d == 0.0)
    double r1 = -b / (2.0*a);
    System.out.println ("Root is " + r1);
else
    System.out.println ("No real solutions");
}

```

Output:

Enter coefficients: 3
4
6
No real coefficients

seen

10/24

Code:

```
import java.util.Scanner;  
import java.lang.Math;
```

```
class Quadratic
{
int a,b,c;
double r1,r2,d1,d,r,x,y;
Scanner s=new Scanner(System.in);

void getval()
{
Scanner s=new Scanner(System.in);
```

```

System.out.println("Enter coefficient of a: ");
a=s.nextInt();
System.out.println("Enter coefficient of b: "); b=s.nextInt();

System.out.println("Enter coefficient of c: ");
c=s.nextInt();
} void calc()
{ if(a==0)
{
System.out.println("Not quadratic, Enter a non zero non negative value of a: ");
a=s.nextInt(); }

d1=(b*b)-(4*a*c);
d=Math.sqrt(d1);
if(d>0)
{
System.out.println("Real Distinct roots: ");
r1=(-b-d)/(2*a); r2=(-b+d)/(2*a);
System.out.println("r1= "+r1+ " r2= "+r2);
} else
if(d==0)
{
System.out.println("Real equal roots: ");
r=(-b)/(2*a);
System.out.println("r1=r2= "+r);
}
else
{
System.out.println("Imaginary roots: "); x=(-b)/(double)(2*a); y=Math.sqrt(-d1)/(double)(2*a);
System.out.println(x+"i"+y);
}
}
class a
{
public static void main(String args[])
{
System.out.println("Name: BJ KEERTANA \nUSN: 1BM23CS034");
Quadratic q=new Quadratic(); q.getval();
q.calc();
}
}

```

Output:

```
Name: BJ KEERTANA
USN: 1BM23CS059
Enter coefficient of a:
7
Enter coefficient of b:
9
Enter coefficient of c:
4
Imaginary roots:
-0.6428571428571429i0.3976974544878587
```

Program 2

Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

Algorithm:

LAB-2

Develop a program to create class Student with members usn, name, an array credits & an array marks. Include methods to accept & display details and a method to calculate SGPA of a student.

```

import java.util.*;
class Student
{
    String name, usn;
    int [] credits;
    int [] marks;

    void accept()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter name: ");
        name = sc.nextLine();
        System.out.println("Enter USN: ");
        usn = sc.nextLine();
        System.out.println("Enter marks: ");
        marks = sc.nextInt();
        int n = sc.nextInt();
        credits = new int[n];
        marks = new int[n];
        for (int j = 0; j < n; j++)
        {
            for (int k = 0; k < n; k++)
            {
                System.out.println("Enter credits: ");
                credits[j] = sc.nextInt();
                System.out.println("Enter marks: ");
                marks[k] = sc.nextInt();
            }
        }
    }
}

```

```

void calc()
{
    double sgpa;
    int c = 0, m = 0;
    for (int j = 0; j < n; j++)
    {
        c += credits[j];
        m += marks[j];
    }
}

```

```

for (int i = 0; i < n; i++)
{
    int s = (marks[i] * credits[i]);
    sgpa = s / c;
    sgpa = sgpa / 10;
}

```

```

void display()
{
    System.out.println("Name: " + name);
    System.out.println("USN: " + usn);
    System.out.println("SGPA: " + sgpa);
    for (int j = 0; j < n; j++)
    {
        System.out.println("Credits: " + credits[j]);
        System.out.println("Marks: " + marks[j]);
    }
}

```

```

public static void main (String args[])
{
    Student obj = new Student ();
    obj.accept ();
    obj.cal ();
    obj.display ();
}

y
y
Op: 8210 + 78 marks total : 160
Output: enter usn : CS019
enter name : Keerthana
enter no of subjects : 2
82
enter credits and marks : 2
78
enter credits and marks : 3
67
USN : CS019
name : Keerthana
Credits for subject 1 is : 2
marks for subject 1 is : 78
Credits for subject 2 is : 3
marks for subject 2 is : 67
total credits : 5
total marks : 145
SGPA is : 6.4
enter usn :
CS020
enter name :
Antra
enter no of subjects :
2

```

Code:

```
import java.util.Scanner;
```

```
class Student
{
    String usn; String
    name; int[] marks=new
    int[8]; int[] credits=new
    int[8]; double sgpa;
```

```
Student() {
```

```

this.name=n
ame; }

void get_details()
{
Scanner sc=new Scanner(System.in);
System.out.print("Enter USN: ");
usn=sc.nextLine();
System.out.print("Enter Name: ");
name=sc.nextLine();

for(int i=0;i<8;i++)
{
System.out.print("Enter marks: ");
marks[i]=sc.nextInt();
System.out.print("Enter credits: ");
credits[i]=sc.nextInt();
}
sc.close();
}

void calc_sgpa()
{ int
tot_marks=0; int
tot_credits=0;

for(int i=0; i<8;i++) {
tot_marks+=marks[i];
tot_credits+=credits[i];
} double sgp=0; for(int
i=0;i<8;i++) {
sgp+=(marks[i]/10.0)*credits[i];
} sgpa=sgp/tot_credits; }

}

void display_details()
{
System.out.println("USN: "+usn);
System.out.println("Name: "+name);

for(int i=0;i<8;i++)
{
}

```

```

System.out.println("marks of subject: "+(i+1)+" is = "+marks[i]);
}

for(int i=0;i<8;i++)
{
System.out.println("credits of subject: "+(i+1)+" is = "+credits[i]);
}

void display_sgpa()
{
System.out.println("SGPA : "+sgpa);

}

} class
Run {
public static void main(String[] args)
{
System.out.println("Name: Ananya N Gowda ");

System.out.println("USN: 1BM23CS034 ");

Student[] arr=new Student[3];
for(int i=0;i<3;i++)
{ arr[i]=new
Student();
arr[i].get_details();

arr[i].display_details();
arr[i].calc_sgpa();
arr[i].display_sgpa();
}

}
}

```

Output:

```
Name: BJ KEERTANA
USN: 1BM23CS059
Enter USN: 68568
Enter Name: keertana
Enter marks: 78
Enter credits: 45
Enter marks: 89
Enter credits: 23
```

Program 3

Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

Algorithm:

LAB-3

Create a class Book with 4 members name, author, price, num-pages. Inside a constructor set values of members. Include the methods of get, set and `toString()` method to display.

```

import java.util.*;
class Book
{
    private String name;
    private String author;
    private int price;
    private int numPages;

    public Book (String name, String author, int price,
                int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public String toString()
    {
        return "Name: " + name + "Author: " + author +
               "Price of book: " + price + "Number of
               pages: " + numPages;
    }

    public static void main (String args[])
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter no. of books");
        int n = sc.nextInt();
        Book [] b = new Book[n];
        for (int i=0; i < n; i++)
        {
            System.out.println ("Enter name of book:");
            String Name = sc.nextLine();
            b[i] = new Book (Name, "Author", 100, 100);
        }
    }
}

```

```

System.out.println("Enter name of author:");
String Author = sc.nextLine();
System.out.println("Enter price of book:");
int Price = sc.nextInt();
System.out.println("Enter number of pages:");
int NumPages = sc.nextInt();
y Book[1] = new Book(name, author, Price, Pages);
b.setName();
b.setAuthor();
b.setPrice();
b.setNumPages();
b.tostring();
y
Output:
enter no. of books: 2
enter name of book: new + " ageop
abcd
enter name of author:
efg
Price of book:
345
number of pages:
256
enter name of book:
enter name of author:
ewgh
Price of book:
567
number of pages:
890
    
```

Book 1:
 Book: abcd
 Author: efg
 Price: 345
 Number of pages: 256

Book 2:
 Book:
 Author: ewgh
 Price: 567
 Number of pages: 890

Code:

```
import java.util.Scanner;
```

```
class Book
```

```

{
String name,author;
double price; int
num_page,book_count;

void get_details()
{
Scanner s1=new Scanner(System.in);
System.out.println("Name: ");
name=s1.nextLine();
System.out.println("Author: ");
author=s1.nextLine();
System.out.println("Price: ");
price=s1.nextDouble();
System.out.println("Page number: ");
num_page=s1.nextInt();
}
public String toString()
{
String name,author,price,num_page; name="Name:
"+this.name+"\n"; author = "Author: " + this.author +
"\n"; price = "Price: " + this.price + "\n"; num_page =
"Number of pages: " + this.num_page + "\n"; return name
+ author + price + num_page;
} } class
dontrun {
public static void main(String[] args)
{
System.out.println("Name: BJ KEERTANA ");
System.out.println("USN: 1BM23CS059");
Scanner s2=new Scanner(System.in); int n;
System.out.println("Number of books: ");
n=s2.nextInt();
Book[] b=new Book[n];
for(int i=0;i<n;i++) {
b[i]=new Book();
b[i].get_details();
}
for(int i=0;i<n;i++)
{
System.out.println(b[i].toString());
}
}
}

```

Output:

Name: BJ KEERTANA

USN: 1BM23CS059

Number of books:

7

Name:

sgvjn

Author:

jgbhjb

Price:

677

Page number:

599

Program 4

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

Algorithmn:

Q. Develop a program to create abstract class shape that contains two integers & empty method printArea(). Provide class rectangle, triangle & circle & print area.

```
import java.util.*;  
abstract class Shape  
{  
    int a, b;  
    Shape (int a, int b)  
    {  
        this.a = a;  
        this.b = b;  
    }  
    abstract void area();  
}  
  
class Rectangle extends Shape  
{  
    Rectangle (int a, int b)  
    {  
        super(a, b);  
    }  
    void area()  
    {  
        int area = (a * b);  
        System.out.println("Area of rectangle: " + area);  
    }  
}
```

```
class Triangle extends Shape  
{  
    Triangle (int a, int b)  
    {  
        super(a, b);  
    }  
}
```

```
void calc()  
{  
    double sgpa;  
    int c=0, m=0;  
    for (int j=0; j<n; j++)  
    {  
        c += credits[j];  
        m += marks[j];  
    }  
    System.out.println("Total credits: " + c);  
    System.out.println("Total marks: " + m);  
    for (int j=0; j<n; j++)  
    {  
        int s = (marks[j] * credits[j]);  
        sgpa = s/c;  
        sgpa = sgpa/10;  
    }  
    void display()  
{  
        System.out.println("Name: " + name);  
        System.out.println("USN: " + usn);  
        System.out.println("SGPA: " + sgpa);  
        for (int j=0; j<n; j++)  
        {  
            System.out.println("Credits: " + credits[j]);  
            System.out.println("Marks: " + marks[j]);  
        }  
    }  
}
```

Output:

Enter sides of rectangle:
4
5
Area of rectangle: 20.0

Enter sides of triangle
65
7
Area of triangle: 224.5

Enter radius of circle: 5
Area of circle is: 78.5

Sol
32.10

Code:

```
import java.util.Scanner;

abstract class Shape
{
int a,b; abstract void
printArea(); }

class Rectangle extends Shape
{
Rectangle(int len,int breadth)
{
this.a=len;
this.b=breadth;
}

void printArea()
{ double
area=a*b;
System.out.println("Rectangle area: "+area);
}

class Triangle
extends Shape
```

```

{
Triangle(int base,int height)
{
this.a=base;
this.b=height;
}

void printArea() {
double area=0.5*a*b;
System.out.println("Triangle area: "+area);
}

class Circle extends Shape
{
Circle(int radius)
{
this.a=radius;

}

void printArea()
{ double
area=a*a;

System.out.println("Circle area: "+area);
}

class Main { public static
void main(String[] args)
{
System.out.println("Name: BJ KEERTANA ");
System.out.println("USN: 1BM23CS059");

Shape rect=new Rectangle(5,10);
rect.printArea();

Shape tri=new Triangle(4,21);
tri.printArea();

Shape circle=new Circle(7);

```

```
circle.printArea();
}
}
```

Output

```
Name: BJ KEERTANA
USN: 1BM23CS059
Rectangle area: 50.0
Triangle area: 42.0
Circle area: 153.93804002589985
```

Program 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: a) Accept deposit from customer and update the balance. b) Display the balance. c) Compute and deposit interest d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

Algorithm:

AB →

Develop a Java program to create class Bank that maintains two kinds of account for its customers, one called savings and other current account. The saving account provides compound interest and withdrawal facilities but no cheque book facility. Current account provides cheque book facility but no interest. Current account holders should also maintain min. balance and if balance below the service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive classes Cur-acct and Sav-acct to make more specific to their requirement. Include necessary method to achieve:

- Accept deposit from customer and update the balance
- Display the balance
- Compute and deposit interest
- Permit withdrawal and update balance

Check for minimum balance, impose penalty if necessary and update balance

class Account

```
String customername;
int accountnumber;
String accounttype;
double balance;
```

```
public Account (String customername, int accountnumber, String accounttype, double balance)
```

this. customername = customername;
this. accountnumber = accountnumber;
this. accounttype = accounttype;
this. balance = balance;

```
public void deposit (double amount)
{
    if (amount > 0)
        balance = balance + amount;
    System.out.println ("New balance : " + balance);
}
```

```
public void withdraw (double amnt)
{
    if (balance >= amnt)
        balance = balance - amnt;
    System.out.println ("New balance : " + balance);
    else
        System.out.println ("Insufficient balance");
}
```

class Sav-Acc extends Account

```
public Sav-Acc (String customername, int accountnumber, double balance, String accounttype, double rate)
{
    super (customername, accountnumber, balance, accounttype);
    this. rate = rate;
}
```

```
public void setInterest (double rate) super (customername, accountnumber, balance, accounttype);
{
    this. rate = rate;
}
```

```
public void computeInterest (int years)
{
    balance * Math.pow (1 + rate), years);
    System.out.println ("New balance : " + balance);
}
```

```

public void withdraw(double amount)
{
    if (balance >= amount)
    {
        balance = balance - amount;
        System.out.println("New balance : " + balance);
    }
    else
    {
        System.out.println("Insufficient Balance");
    }
}

public class Current_Account extends Account
{
    private double minbalance;
    private double penamt;

    public Current_Account (String customername, int accountnumber, double balance, String accounttype, double minbalance, double penamt)
    {
        super(customername, accountnumber, balance, accounttype);
        this.minbalance = minbalance;
        this.penamt = penamt;
    }

    public void withdraw (double amount)
    {
        if (balance >= amount)
        {
            balance -= amount;
            System.out.println("Amount withdrawn");
        }
        if (balance < minbalance)
        {
            balance -= penamt;
            System.out.println("Penalty amt : " + penamt);
        }
        else
        {
            System.out.println("Insufficient balance");
        }
    }
}

```

```

y
public class Bank
{
    public static void main (String args[])
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter name : ");
        String name = sc.nextLine();
        System.out.println ("Enter ac number");
        System.out.println ("Enter acntGnt");
        int accountnumber = sc.nextInt();
        System.out.println ("Enter balance");
        double balance = sc.nextDouble();
        System.out.println ("Enter account type");
        String accounttype = sc.nextLine();

        switch (accounttype)
        {
            case "savings":
                Sav-Acc s = new Sav-Acc (Systemin);
                s.deposit (1000);
                s.display ();
                s.withdraw (500);
                s.computeInterest ();
                break;
            case "current":
                System.out.println ("Enter min balance and penalty");
                double minbalance = sc.nextDouble();
                double penamt = sc.nextDouble();
        }
    }
}

```

```

Enter your choice : 2
Enter amount : 500
New balance : 900.0
1. Deposit
2. withdraw
3. Display
4. Compute and deposit
5. End
Enter your choice : 3
Current balance : 900.0

```

Code:

```

import java.util.Scanner;

class Account {
    protected String CustName;
    protected int AccNo;
    protected double balance;
    public Account(String CustName, int AccNo, double balance) {
        this.CustName = CustName;
        this.AccNo = AccNo;
        this.balance = balance;
    }

    public void deposit(double amount) {if
(amount > 0) {
        balance += amount;
        System.out.println("Deposited: " + amount);
    } else {
        System.out.println("Invalid Deposit Amount");
    }
}

```

```

        }
    public void displayBalance() {
        System.out.println("Balance: " + balance);
    }
}

public void deposit(double amount) {if
(amount > 0) {
    balance += amount;
    System.out.println("Deposited: " + amount);
} else {
    System.out.println("Invalid Deposit Amount");
}
}

public void displayBalance() {
    System.out.println("Balance: " + balance);
}
}
}

public void deposit(double amount) {if
(amount > 0) {
    balance += amount;
    System.out.println("Deposited: " + amount);
} else {
    System.out.println("Invalid Deposit Amount");
}
}

public void displayBalance() {
    System.out.println("Balance: " + balance);
}
}
}

```

```

class SavAcct extends Account {
    private double interestRate;
    public SavAcct(String customerName, int accountNumber, double balance, double interestRate) {
        super(customerName, accountNumber, balance);
        this.interestRate = interestRate;
    }
    public void computeAndDepositInterest() {
        double interest = balance * (interestRate / 100);
        balance += interest;
        System.out.println("Interest added: " + interest);
    }
}

```

```

    }
    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance for withdrawal");
        }
    }

    class SavAcct extends Account {
        private double interestRate;
        public SavAcct(String customerName, int accountNumber, double balance, double interestRate) {
            super(customerName, accountNumber, balance);
            this.interestRate = interestRate;
        }
        public void computeAndDepositInterest() {
            double interest = balance * (interestRate / 100);
            balance += interest;
            System.out.println("Interest added: " + interest);
        }
        public void withdraw(double amount) {
            if (amount <= balance) {
                balance -= amount;
                System.out.println("Withdrawn: " + amount);
            } else {
                System.out.println("Insufficient balance for withdrawal");
            }
        }
    }
} class CurAcct extends Account

{ private double
minimumBalance; private
double serviceCharge;

    public CurAcct(String customerName, int accountNumber, double balance, double
minimumBalance, double serviceCharge) {
        super(customerName, accountNumber, balance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }
}

```

```

}

public void withdraw(double amount) {
    if (amount <= balance) {
        balance -= amount;
        System.out.println("Withdrawn: " + amount);
        if (balance < minimumBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge imposed: " + serviceCharge);
        }
    } else {
        System.out.println("Insufficient balance for withdrawal");
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        SavAcct savAcc = new SavAcct("Aaron", 12345, 10000, 5);
        CurAcct curAcc = new CurAcct("Aaron", 12345, 10000, 5000, 500);
        System.out.println("Choose Account Type:\n1. Savings Account\n2. Current Account");
        int choice = sc.nextInt();
        switch (choice) {
            case 1:
                System.out.println("Savings Account Selected");
                savAcc.deposit(700);
                savAcc.computeAndDepositInterest();
                savAcc.withdraw(500);
                savAcc.displayBalance();
                break;
            case 2:
                System.out.println("Current Account Selected");
                curAcc.deposit(800);
                curAcc.withdraw(200);
                curAcc.displayBalance();
                break;
            default:
                System.out.println("Invalid choice");
        }
        sc.close();
    }
}

```

Output:

```
Choose Account Type:  
1. Savings Account  
2. Current Account  
1  
Savings Account Selected  
Deposited: 700.0  
Interest added: 535.0  
Withdrawn: 500.0  
Balance: 10735.0  
  
C:\Users\aryan\Downloads>java Bank  
Choose Account Type:  
1. Savings Account  
2. Current Account  
2  
Current Account Selected  
Deposited: 800.0  
Withdrawn: 200.0  
Balance: 10600.0
```

Program 6

Create a package CIE which has two classes - Personal and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Personal. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

Algorithm:

Lab-06
Experiment Result

Create a package CIE with two classes - Personal and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores internal marks of five courses of current semester. Create another package SEE which has class External which is derived from class Personal. This class has an array that stores SEE marks scored in five courses of current semester of the student. Import two packages in a file that declare final marks of n students.

```
package CIE;
public class Personal
{
    protected String usn;
    protected String name;
    protected int sem;
    public Personal (String usn, String name, int sem)
    {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
    public void displayPersonalInfo()
    {
        System.out.println("USN: " + usn);
    }
}
```

```
System.out.println("Name: " + name);
System.out.println("Semester: " + sem);
}
}
package CIE;
import java.util.Scanner;
public class Internals extends Student
{
    protected int marks [] = new int [5];
    public void inputCIEmarks()
    {
        Scanner sc = new Scanner (System.in);
        System.out.println("Enter internal marks for 5 courses");
        for (int i=0; i<5; i++)
        {
            System.out.print ("Courses: " + (i+1) + " ");
            marks [i] = sc.nextInt();
        }
    }
    public void displayCIEmarks()
    {
        System.out.println ("Internal marks: ");
        for (int i=0; i<5; i++)
        {
            System.out.print ("Courses: " + (i+1) + " " + marks[i]);
        }
    }
}
package CIE;
import CIE.internals;
import java.util.Scanner;
public class External extends Internals
{
    public void calculateFinalMarks()
    {
        System.out.println ("Final marks = " + ((marks[0]*3+marks[1]*3+marks[2]*3+marks[3]*3+marks[4]*3)/5));
    }
}
```

```

2 protected int externalMarks[] = new int[5];
protected int finalMarks[] = new int[5];
public External() {
    externalMarks = new int[5];
    finalMarks = new int[5];
}

y public void input SEEmarks() {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter external marks");
    for (int i = 0; i < 3; i++) {
        System.out.println("Course " + (i + 1) + ": Final");
    }
}

y import SEE External;
import java.util.*;
public class Main {
    public static void main (String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter no. of students");
        int n = sc.nextInt();
        External[] student = new External[n];
        for (int i = 0; i < n; i++) {
            student[i] = new External();
            student[i].input StudentDetails();
            student[i].input (IE marks());
            student[i].input SEE marks();
            student[i].calculate FinalMarks();
        }
    }
}

```

```

for (i=0; i<n; i++)
    Students[i].displayFinalMarks();
}

y
y
y

Enter no. of students: 2
Enter details of 1
Enter usn: 123
Enter name: abc
Enter sem: 3
Enter 5 internal marks: 45 46 47 48 49
Enter 5 SEE marks: 96 98 97 95 94

Enter details of 2
Enter usn: 236
Enter name: fgh
Enter sem: 3
Enter 5 internal marks: 43 44 47 48 50
Enter 5 SEE marks: 98 93 92 99 100

Final marks of Students:
USN: 123
Name: abc
Internal marks: 45 46 47 48 49
SEE marks: 98 97 96 95 94
Final marks: 94 94 93 95 96

USN: 236
name: fgh
sem: 3
internal marks: 43 44 47 48 50
SEE marks: 98 93 92 99 100
Final marks: 94 90 93 97 100

```

Code:

//Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

```
package cie;
import java.util.*;
class Personal
{
    private String usn;
    private String name;
    private int sem;
    Personal(String usn, String name, int sem)
    {
        this.name=name;
        this.usn=usn;
        this.sem=sem;
    }
    void input()
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter name: ");
        name=sc.nextLine();
        System.out.println("Enter usn: ");
        usn=sc.nextLine();
        System.out.println("Enter sem: ");
        sem=sc.nextInt();
    }
    void display()
    {
        System.out.println("Name: "+name);
        System.out.println("usn: "+usn);
        System.out.println("Semester: "+sem);
    }
}
package cie;
import java.util.*;

class Internals extends Personal
{
    protected int[] marks=new int[5];
    public Internals(String usn, String name, int sem) {
        super(usn, name, sem);
    }
    public void inputcie(){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter cie marks: ");
        for(int i=0;i<5;i++){
            System.out.println("Course"+(i+1)+": " );
        }
    }
}
```

```

        marks[i]=sc.nextInt();
    }
}
public void displaycie(){
    System.out.println("cie marks: ");
    for(int i=0;i<5;i++){
        System.out.println("Course"+(i+1)+":"+marks[i] );
    }
}
package SEE;
import cie.Internals;
import java.util.*;
class Externals extends Internals{
    protected int[]seemks=new seemks[5];
    protected int[]finalmks=new finalmks[5];
    public Externals(String usn, String name, int sem) {
        super(usn, name, sem);
    }
    public void inputsee(){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter see marks: ");
        for(int i=0;i<5;i++){
            System.out.println("Course"+(i+1)+": " );
            seemks[i]=sc.nextInt();
        }
    }
    public void calculatefinals(){
        for(int i=0;i<5;i++)
        {
            finalmks[i]=seemks[i]+marks[i];
        }
    }
    public void displayfinal(){
        display();
        displaycie();
        System.out.println("final marks: ");
        for(int i=0;i<5;i++){
            System.out.println("Course"+(i+1)+":"+finalmks[i] );
        }
    }
}

import SEE.Externals;
import java.util.Scanner;
public class Main{
    public static void main (String args[]){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter no of students");
        int n=sc.nextInt();
        Externals[] student=new Externals[n];
        for(int i=0;i<n;i++){

```

```

student[i]=new Externals();
student[i].input();
student[i].inputcie();
student[i].inputsee();
student[i].calculatefinals();

}

for(int i=0;i<n;i++){
    student[i].displayfinal();
}
}
}

```

output:

```

Total Students:
2
Enter USN for student 1:
001
Enter Name for student 1:
ABC
Enter USN:
002
Enter Name:
XYZ
Enter Semester:
3
USN: 002
Name: XYZ
Semester: 3
Enter marks of subject 1:
50
Enter marks of subject 2:
50
Enter marks of subject 3:
50
Enter marks of subject 4:
50
Enter marks of subject 5:
50
Enter marks of external exam for subject 1:
50
Enter marks of external exam for subject 2:
50
Enter marks of external exam for subject 3:
50
Enter marks of external exam for subject 4:
50
Enter marks of external exam for subject 5:
50
USN: 002
Name: XYZ
Semester: 3
Final marks for each subject:
Subject 1: 100
Subject 2: 100
Subject 3: 100
Subject 4: 100
Subject 5: 100
Enter USN for student 2:

```

Program 7

We have created an interface named **Polygon**. It includes a default method **getPerimeter()** and an abstract method **getArea()**. We can calculate the perimeter of all polygons in the same manner so we implemented the body of **getPerimeter()** in **Polygon**. Now, all polygons that implement **Polygon** can use **getPerimeter()** to calculate perimeter. However, the rule for calculating the area is different for different polygons. Hence, **getArea()** is included without implementation. Any class that implements **Polygon** must provide an implementation of **getArea()**.

Algorithm:

```
create Polygon which includes default getPerimeter()
and abstract method getArea() (calculate
perimeter & area of all polygons. Any class
implements polygon must provide implementation
of getArea().)

import java.util.*;
interface Polygon{
    public void getPerimeter();
    public abstract void getArea();
}

class Square implements Polygon{
    Scanner sc=new Scanner(System.in);
    public void getPerimeter(){
        System.out.println("Enter no. of sides");
        int n=sc.nextInt();
        int p=0;
        for (int i=0;i<n;i++){
            System.out.println("Enter side");
            int s=sc.nextInt();
            p+=s;
        }
        System.out.println("Perimeter: "+p);
    }

    public void getArea(){
        System.out.println("Enter side length");
        int a=sc.nextInt();
        int area=a*a;
        System.out.println("Area: "+area);
    }
}
```

```
class Triangle implements Polygon{
    Scanner sc=new Scanner(System.in);
    public void getPerimeter(){
        System.out.println("Enter no. of sides");
        int n=sc.nextInt();
        for (int i=0;i<n;i++)
            System.out.println("Enter side");
        int s=sc.nextInt();
        p+=s;
        System.out.println("Perimeter: "+p);
    }

    public void getArea(){
        System.out.println("Enter base and height");
        int b=sc.nextInt();
        int h=sc.nextInt();
        double area = 0.5*b*h;
        System.out.println("Area: "+area);
    }
}

public class Main {
    public static void main (String args[]){
        Scanner sc=new Scanner(System.in);
        Polygon sq=new Square();
    }
}
```

```

18. getPerimeter();
19. getArea();
Polygon t = new Triangle();
t.getPerimeter();
t.getArea();
y
y
Output
Enter no. of sides : 4
Enter side : 5
Perimeter : 20
Enter side length : 5
Area : 25.00
Enter no. of sides : 3
Enter side : 3
Enter side : 2
Enter side : 1
Perimeter : 6
Enter base and height : 3
Area : 6

```

Code:

```
import java.util.*;
```

```
interface Polygon {
    void getPerimeter();
    void getArea();
}
```

```
class Square implements Polygon {
    private int side;

    public Square() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the side length of the square: ");
        this.side = sc.nextInt();
    }
}
```

@Override

```

public void getPerimeter() {
    int perimeter = 4 * side;
    System.out.println("Perimeter of Square: " + perimeter);
}

@Override
public void getArea() {
    int area = side * side;
    System.out.println("Area of Square: " + area);
}
}

class Triangle implements Polygon {
    private int a, b, c, base, height;

    public Triangle() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the lengths of the three sides of the triangle: ");
        this.a = sc.nextInt();
        this.b = sc.nextInt();
        this.c = sc.nextInt();

        System.out.println("Enter the base of the triangle: ");
        this.base = sc.nextInt();

        System.out.println("Enter the height of the triangle: ");
        this.height = sc.nextInt();
    }

    @Override
    public void getPerimeter() {
        int perimeter = a + b + c;
        System.out.println("Perimeter of Triangle: " + perimeter);
    }

    @Override
    public void getArea() {
        double area = 0.5 * base * height;
        System.out.println("Area of Triangle: " + area);
    }
}

public class Main {
    public static void main(String[] args) {
        // Create Square object
        Polygon square = new Square();
        square.getPerimeter();
    }
}

```

```
square.getArea();  
  
Polygon triangle = new Triangle();  
triangle.getPerimeter();  
triangle.getArea();  
}  
}
```

Output:

```
Rectangle Area: 50.0  
Rectangle Perimeter: 30.0  
Circle Area: 153.93804002589985  
Circle Perimeter: 43.982297150257104  
Triangle Area: 6.0  
Triangle Perimeter: 12.0
```

Program 8

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age<0. In Son class, implement a constructor that uses both father and son's age and throws an exception if son's age is >=father's age.

Algorithm:

14B

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age < 0. In Son class, implement a constructor that uses both father and son's age and throws an exception if son's age is >= father's age.

```
import java.util.*;
class WrongAge extends Exception {
    public class WrongAge (String message) {
        super(message);
    }
}
class Father {
    int fage;
    public Father (int age) throws WrongAge {
        if (age < 0)
            throw new WrongAge ("Father age cannot be negative!");
        this.fage = age;
    }
    System.out.println("Father age : " + age);
}
class Son extends Father {
    int sonAge;
    public Son (int fage, int sonAge) throws WrongAge {
        super(fage);
    }
}
```

if (sonAge < 0)
 throw new WrongAge ("Son age cannot be negative");
if (sonAge >= fage)
 throw new WrongAge ("Son's age cannot be greater or equal to father age");
 this.sonAge = sonAge;
System.out.println("Son age : " + sonAge);

public class InheritanceExceptionDemo {
 public static void main (String args[]) {
 try {
 Father f = new Father (45);
 Son s = new Son (45, 20);
 } catch (WrongAge e) {
 System.out.println("Exception caught : " + e.getMessage());
 }
 try {
 Father f = new Father (-5);
 } catch (WrongAge e) {
 System.out.println("Exception caught : " + e.getMessage());
 }
 try {
 Son s = new Son (40, 50);
 } catch (WrongAge e) {
 System.out.println("Exception caught : " + e.getMessage());
 }
 }
}

```

if (lonAge < 0) {
    throw new WrongAge("Son age cannot be negative")
}

if (lonAge != fAge) {
    throw new WrongAge("Son's age cannot be greater
                        or equal to father age"),
}

    lonAge = lonAge;
    System.out.println("Son age : " + lonAge);

public class InheritanceExceptionDemo {
    public static void main (String args[]) {
        try {
            Father f = new Father(45);
            Son s = new Son(45, 20);
        }
        catch (WrongAge e) {
            System.out.println("Exception caught : " + e.getMessage());
        }
        try {
            Father f = new Father(-5);
        }
        catch (WrongAge e) {
            System.out.println("Exception caught : " + e.getMessage());
        }
        try {
            Son s = new Son(40, 50);
        }
        catch (WrongAge e) {
            System.out.println("Exception caught : " + e.getMessage());
        }
    }
}

```

Output:

Father's age : 45
Father's age : 45
Son's age : 20
Exception caught : Father's age cannot be negative
Father's age : 40
Exception caught : Son's age cannot be greater than or equal to Father's age

Extra Programs:

- 1) Arithmetic Exception \Rightarrow / by zero
- 2) File: test.txt is missing. Please check file name
- 3) Please enter your age - Numeric value
 - 11 you are not authorized
 - 64 you are authorized.
- 4) java.lang.ArithmaticException: / by zero at GFG1.main(GFG1.java:9)
- java.lang.ArithmaticException: / by zero
- java.lang.ArithmaticException: / by zero

Type an integer
23
you typed 23
as trapping exception

Code:

```

import java.util.Scanner;
class WrongAge extends Exception{

```

```

int a;
WrongAge(int a)
{
this.a=a;} public String
toString(){ return a+" is a
invalid Age";
}
class SonAgeExceedsFatherAge extends Exception{
int fa,a;
SonAgeExceedsFatherAge(int fa,int a)
{

this.fa=fa;
this.a=a;
}
public String toString(){ return "father's("+fa+") age cannot be lesser
than that of son("+a+");"
}} class
Father { int
fage;
Father(int a) {
fage=a; }
public void fathervalidage() throws WrongAge{
if(fage<0){
throw new WrongAge(fage);}
}}
class Son extends Father{
int age;
Son(int fa,int a)
{
super(fa); age=a;} public void sonvalidage() throws
SonAgeExceedsFatherAge{ if(fage<age){ throw new
SonAgeExceedsFatherAge(fage,age);
}}
void
display()
{
System.out.println("Father's age:"+fage+"\nSon's age:"+age);
}
}class FatherSon{ public static void
main(String args[]){
Scanner sc=new Scanner(System.in);
System.out.println("BJ KEERTANA \n1BM23CS059");
}

```

```
System.out.println("Enter Father's age:");
int fage=sc.nextInt();
System.out.println("Enter Son's age:");
int age=sc.nextInt();
Son child=new Son(fage,age);
try{
    child.fathervalidage();
    child.sonvalidage();
    System.out.println("Ages are valid");
    child.display();
} catch(WrongAge e){
    System.out.println(e);
} catch(SonAgeExceedsFatherAge e){

    System.out.println(e);
}}
```

Output:

```
Father's age is set to: 45
Father's age is set to: 45
Son's age is set to: 20
Exception caught: Father's age cannot be negative.
Father's age is set to: 40
Exception caught: Son's age cannot be greater than or equal to Father's age.
```

Program 9

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

Algorithm:

Write a program which creates two threads, one thread displaying "BMS College of Engineering" one every two seconds.

```

class Col1 extends Thread {
    public static void main (String args[]) {
        public void run() {
            Thread t1 = new Thread () {
                while (true) {
                    System.out.println ("BMS College of Engineering");
                }
            }
            try {
                Thread.sleep (1000);
            } catch (InterruptedException e) {
                if do something
            }
        }
    }
}

class Col2 extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println ("CSE");
                Thread.sleep (2000);
            }
        } catch (InterruptedException e) {
            if do something
        }
    }
}

```

public class multiThread
{
 public static void main (String args [])
 {
 Thread t1 = new Thread (new Runnable ()
 {
 public void run ()
 {
 System.out.println ("Thread 1 running");
 }
 }, "Thread 1");
 Thread t2 = new Thread (new Runnable ()
 {
 public void run ()
 {
 System.out.println ("Thread 2 running");
 }
 }, "Thread 2");
 t1.start ();
 t2.start ();
 }
}

Output :
Thread 1 running
Thread 2 running

BMS College of Engineering

LSE
LSE
CSE
CSE
CSE
BMS College of Engineering

CSE
CSE
CSE
CSE
CSE
BMS College of Engineering

CSE
CSE
CSE
CSE
CSE
BMS College of Engineering

Year + C

Code:

class BMSCE extends Thread

```

{
public void run()
{
for(int i=1; i<=2; i++)
{
System.out.println("BMSCE " + i);

try
{Thread.sleep(10*1000);
}
catch(InterruptedException e)
{
}
}

class CSE extends Thread
{
public void run()
{
for(int i=1; i<=10; i++)
{
System.out.println("CSE " + i);
try
{Thread.sleep(2*1000);
}
catch(InterruptedException e)
{
}
}

public class Main
{
public static void main(String args[])
{
System.out.println("USN: 1BM23CS059\nName:BJ KEERTANA \n");
BMSCE b1 = new BMSCE();
b1.start();
}

```

```
CSE c1 = new CSE();
c1.start();

}
}
```

Output:

```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
|
```

Program 10

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

Algorithm:

```

1.
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class swingDemo
{
    swingDemo()
    {
        JFrame jfrm = new JFrame ("Divide App");
        jfrm.setSize (245,150);
        jfrm.setLayout (new FlowLayout ());
        jfrm.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel ("Enter divisor  
and dividend");
        JTextField aif = new JTextField (8);
        JTextField bif = new JTextField (8);
        JButton button = new JButton ("Calculate");

        JLabel eml = new JLabel ();
        JLabel alab = new JLabel ();
        JLabel blab = new JLabel ();
        JLabel anslab = new JLabel ();

        jfrm.add (eml);
        jfrm.add (jlab);
        jfrm.add (aif);
        jfrm.add (bif);
        jfrm.add (button);
        jfrm.add (alab);
        jfrm.add (blab);
        jfrm.add (anslab);
    }
}

```

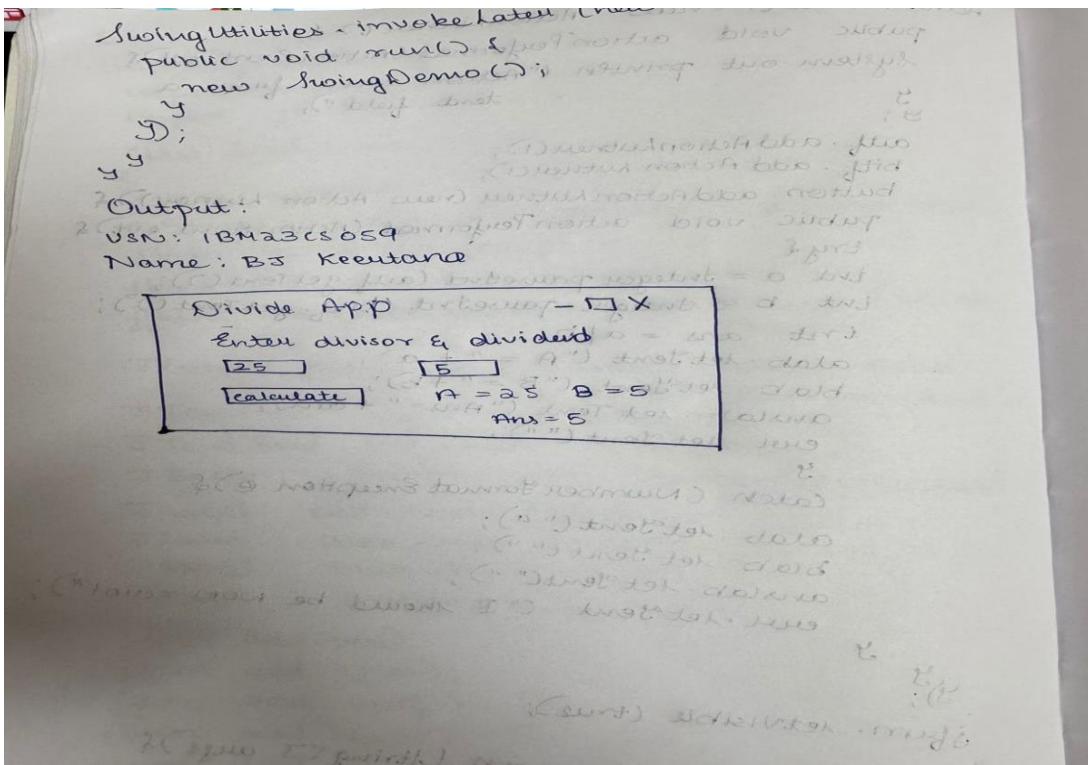
```

ActionListener = new ActionListener()
{
    public void actionPerformed (ActionEvent evt)
    {
        System.out.println ("Action event from a
        text field");

        aif.addActionListener (this);
        bif.addActionListener (this);
        button.addActionListener (new ActionListener());
        public void actionPerformed (ActionEvent evt)
        {
            try
            {
                int a = Integer.parseInt (aif.getText ());
                int b = Integer.parseInt (bif.getText ());
                int ans = a/b;
                alab.setText ("A = " + a);
                blab.setText ("B = " + b);
                anslab.setText ("Ans = " + ans);
                eml.setText ("");
            }
            catch (NumberFormatException e)
            {
                alab.setText ("");
                blab.setText ("");
                anslab.setText ("");
                eml.setText ("B should be NON zero!");
            }
        }
    }
};

public static void main (String [] args)
{
    System.out.println ("USN: IBMSQ3 CS059 \n
                        Name: BJ Keerthana");
}

```



Code:

```

import javax.swing.*;  

import java.awt.*;  

import java.awt.event.*;  

class SwingDemo {  
  

  SwingDemo() {  

    JFrame jfrm = new JFrame("Divider App");  

    jfrm.setSize(275, 150); jfrm.setLayout(new FlowLayout());  

    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  

    ;  
  

    JLabel jlab = new JLabel("Enter the divisor and dividend:");  

    JTextField ajtf = new JTextField(8);  

    JTextField bjtf = new JTextField(8);  

    JButton button = new JButton("Calculate");  
  

    JLabel err = new JLabel();  

    JLabel alab = new JLabel();  

    JLabel blab = new JLabel();  

    JLabel anslab = new JLabel();
  }
}
  
```

```

jfrm.add(err);
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

ActionListener l = new ActionListener() { public
    void actionPerformed(ActionEvent evt) {
        System.out.println("Action event from a text field");
    }
};

ajtf.addActionListener(l);
bjtf.addActionListener(l);
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(ajtf.getText());
            int b =
                Integer.parseInt(bjtf.getText()); int
            ans = a / b; alab.setText("A = " + a);
            blab.setText("B = " + b);
            anslab.setText("Ans = " + ans);
            err.setText("");
        } catch (NumberFormatException e) {
            alab.setText(""); blab.setText("");
            anslab.setText("");
            err.setText("Enter Only Integers!");
        } catch (ArithmaticException e) {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("B should be NON zero!");
        }
    }
});
jfrm.setVisible(true);
}

public static void main(String[] args) {
System.out.println("USN: 1BM23CS0359\nName: BJ KEERTANA\n");

```

```
SwingUtilities.invokeLater(new Runnable() {
    public void run() {
        new SwingDemo();
    }
});
```

Output:

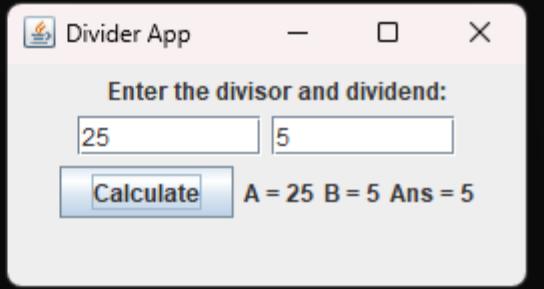
```
D:\ananya>java SwingDemo
```

```
D:\ananya>Javac SwingDemo.java
```

```
D:\ananya>java SwingDemo
```

```
USN: 1BM23CS034
```

```
Name: Ananya N Gowda
```



INDEX OF OBSERVATION: