

B.M.S. COLLEGE OF ENGINEERING
Basavanagudi, Bengaluru- 560019
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



LAB REPORT

On

**Object Oriented Java Programming
(23CS3PCOOJ)**

Submitted By:
ADITHYA RAVIKEERTHI
1BM22CS020

In partial fulfilment of
BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE AND ENGINEERING
2023-24

Faculty-In-Charge
Swathi Sridharan
Assistant Professor
Department of Computer Science and Engineering

LAB-1: QUADRATIC EQUATION

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.

Step 6 : if $d < 0$
 print (" Roots are imaginary")
 $\alpha_1 = -b / (c * a)$
 $\alpha_2 = \sqrt{abs(d)} / (c * a);$
 print ($\alpha_1 + " + i " + \alpha_2$)
 print ($\alpha_1 + " - i " + \alpha_2$)
 go to step 8

Step 7 : if $d = 0$
 print (" Roots are real equal")
 $\alpha_1 = -b / (c * a);$
 print (α_1);

Step 8 = Step

Java code

```
import java.util.Scanner;
import static java.lang.Math.sqrt;
import static java.lang.Math.abs;

public class New {
    public static void main (String[] args) {
        Scanner in = new Scanner (System.in);
        System.out.println ("Enter coefficients of
            a quadratic equation");
        int a = in.nextInt();
        int b = in.nextInt();
        int c = in.nextInt();
        if (a == 0) {
            System.out.println ("Enter invalid input");
        } else {
            int d = b * b - 4 * a * c;
            if (d > 0) {
                System.out.println (" Roots are real");
                float alpha1 = (float) (-b + sqrt(d)) / (c * a);
                float alpha2 = (float) (-b - sqrt(d)) / (c * a);
                System.out.println (alpha1);
                System.out.println (alpha2);
            }
        }
    }
}
```

```

else if(d < 0) {
    system.out.println("Roots are
                        imaginary");
    float s1 = (float) -b / (c2 * a);
    float s12 = (float) sqrt(abs(d)) / (c2 * a);
    system.out.println(s1 + " + " + i + s12);
    system.out.println(s1 + " - " + i + s12);
}

```

```
} else {  
    System.out.println("Roots are equal");  
    float d = (float) -b / (c * a);  
    System.out.println(d);  
}
```

3
3

Output:

→ Enter the coefficients of a quadratic equation

164

Roots were read

-0.76398205

-5.236068

→ Enter the coefficient of a quadratic equation

121

Roots are real and equal

-1- 0

→ Enter the coefficients of a quadratic equation

712

Roots are imaginary

$$-0.0714285 + i \cancel{0.52972}$$

$$-0.0714285 - 10.52972$$

→ Enter the coefficients of a quadratic equation.

0 7 8

Invalid input.

OUTPUT :

```
adithya@adi82:~/adi$ javac quad.java
adithya@adi82:~/adi$ java quad
Enter num 1 :
21
Enter num 2 :
-2
Enter num 3 :
1
Given equation does not have any real solutions
Adithya Ravikeerthi 1BM22CS020
```

LAB-2: STUDENT SGPA CALCULATOR

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.

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

```
public class Student {  
    String usn;  
    String name;  
    int i = 0;  
    private int[] credit = {4, 4, 3, 3, 1, 1, 1};  
    public Student() {  
        this.usn = "18ME017";  
        this.name = "Marty";  
    }
```

```
    public double avg(int[] arr) {  
        double marks = 0, CGPA = 0;  
        for (i = 0; i < arr.length; i++) {  
            marks = (credit[i] * (int) arr[i]) / (i + 1);  
        }  
        CGPA = marks / 70;  
        return CGPA;  
    }
```

```
}  
  
import java.util.Scanner;  
public class Student {  
    String usn;  
    String name;  
    int i = 0;  
    private static int[] credit = {4, 4, 3, 3, 1, 1, 1};  
    Scanner in = new Scanner(System.in);  
    public void first(){  
        System.out.print("Enter your UGN: ");  
        usn = in.next();  
        System.out.print("Enter name: ");  
        name = in.next();  
    }
```

```
public double avg(int[] arr){  
    double sgpa=0, marks;  
    for(i=0; i<arr.length; i++){  
        if (arr[i] >= 100){  
            arr[i] = arr[i]-10;  
        }  
        else if (arr[i] < 40){  
            arr[i] = 0;  
        }  
        marks += credit[i] * (arr[i]/10)+1;  
    }  
    sgpa = marks/20;  
    return sgpa;  
}  
  
public void display(double result){  
    System.out.println("SGPA = " + result);  
}  
  
public class Main{  
    import java.util.Scanner;  
    public class Main{  
        public static void main(String[] args){  
            Scanner in = new Scanner(System.in);  
            int[] arr = new int[8];  
            Student s1 = new Student;  
            s1.first();  
            System.out.println("Enter marks:");  
            for(int i=0; i<8; i++){  
                arr[i] = in.nextInt();  
            }  
            double result = s1.avg(arr);  
            s1.display(result);  
        }  
    }
```

```
adithya@adi82:~/adi$ javac SGPAProgram.java
```

```
adithya@adi82:~/adi$ java SGPAProgram
```

Enter your USN: 20

Enter your name: adithya

Enter your marks for subject 1: 80

Enter your marks for subject 2: 75

Enter your marks for subject 3: 96

Enter your marks for subject 4: 62

Enter your marks for subject 5: 87

Enter your marks for subject 6: 90

Enter your marks for subject 7: 76

Enter your marks for subject 8: 88

Student Details:

USN: 20

Name: adithya

Marks for subject 1: 80

Marks for subject 2: 75

Marks for subject 3: 96

Marks for subject 4: 62

Marks for subject 5: 87

Marks for subject 6: 90

Marks for subject 7: 76

Marks for subject 8: 88

SGPA: 8.65

LAB-3: BOOK DETAILS

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.

Details of a Book (Input & display of the details of book using toString())

```

import java.util.Scanner;
class Books {
    String name;
    String author;
    int price;
    int num_pages;
    public void set(int i) {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter details of book");
        + (i+1) + " " + name, author, price, num_pages
        order");
        name = in.next();
        author = in.next();
        price = in.nextInt();
        num_pages = in.nextInt();
    }
    public String toString() {
        return "Details of Book " + (i+1) + "In"
        "Name : " + name + "In" +
        "Author : " + author + "In" +
        "Price : " + price + "In" +
        "No. of pages : " + num_pages;
    }
    public void get(int i) {
        String s = toString(i);
        System.out.println(s);
    }
}
class Main {
    public static void main(String[] args) {
        int n;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter number of books");
        n = in.nextInt();
        Books b[] = new Books[n];
        for (int i=0; i<n; i++) {
            b[i] = new Books();
            b[i].set(i);
        }
        System.out.println();
    }
}

```

```
for (int i=0; i<n; i++) {  
    system.out.println(b[i].getAuthor(i));  
}
```

Algorithm:

Step 1 : Start

Step 2 : Initialize variable under structure
struct Book{

```
    String name;  
    String author;  
    int price;  
    int num-pages;  
};
```

Step 3 : ~~for (i=0; i<n; i++)~~ Read n.

Step 4 : ~~for (i=0; i<n; i++) {~~

Print "Enter details of the book"

 name = in.next() Read b.name[i]

 b.author[i], b.price[i], b.num-pages[i]

 } continue until it breaks the loop.

Step 5 : Print "Display details of books"

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

 Print "Name " + b.name[i]

 Print "Author " + b.author[i]

 Print "Price : " + b.price[i]

 Print "No. of pages " + b.num-pages[i]

Step 6 : Stop

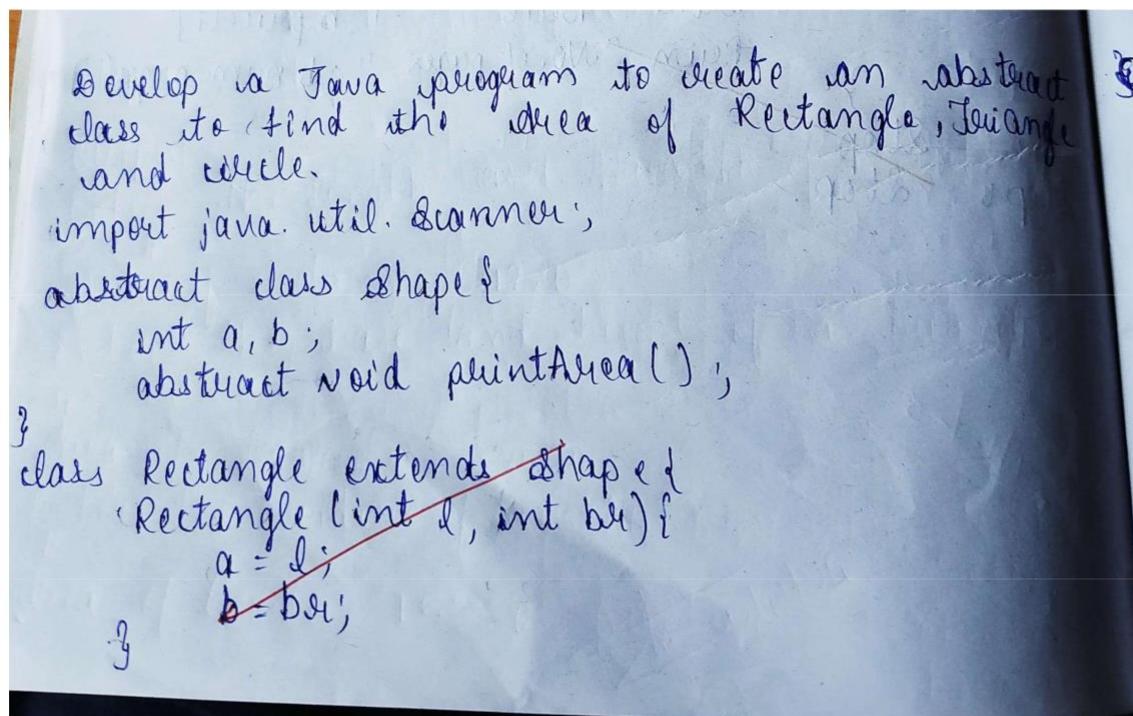
Step 6 : Stop

OUTPUT :

```
adithya@adi82:~/adi$ javac Main.java
adithya@adi82:~/adi$ java Main
Book name=GOT, author=George R. R. Martin, price=1000, num_of_pages=2000
Book name=Crime and Punishment, author=Fyodor Dostoevsky, price=2500, num_of_pages=1500
Adithya Ravikeerthi 1BM22CS020
```


LAB-4: AREA CALCULATION

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.



```

public void paintArea(){
    System.out.println("Area of Rectangle = "+l*b);
}

class Triangle extends Shape{
    Triangle(int ba, int h){
        a = ba;
        b = h;
    }

    public void paintArea(){
        System.out.println("Area of triangle = ");
        double area = 0.5 * a * b;
        System.out.println("Area of triangle = " +
                           area);
    }
}

class Circle extends Shape{
    Circle(int r){
        a = r;
    }

    public void paintArea(){
        double area = 3.14 * a * a;
        System.out.println("Area of the circle = " +
                           area);
    }
}

class Main{
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the base and height of a rectangle");
        Rectangle rec = new Rectangle(
            in.nextInt(), in.nextInt());
        rec.paintArea();
        System.out.println("Enter the base and height of a triangle");
        Triangle tri = new Triangle(
            in.nextInt(), in.nextInt());
        tri.paintArea();
    }
}

```



```
    system.out.println("Enter the radius  
of a circle");  
    circle.cir = new Circle(in.nextInt());  
    cir.printArea();
```

{

3

Algorithm:

Step 1 : Start

Step 2 : Init create abstract class Shape
in which initialize variable a & b.Step 3 : Call for printArea() function in abstract
classStep 4 : Enter length & breadth of a rectangle
(l, b) under class Rectangle extends
Shape
Print "Area of rectangle" + l * b.Step 5 : Read base(ba) and height(h) of a triangle,
under class Triangle extends Shape

$$a = ba \quad b = h$$

Print "Area of triangle" + (0.5 * b * h)

Step 6 : Read radius(r) of a circle
under class Circle of extends Shape

$$a = r$$

Print "Area of circle" + (3.14 * r * r)

Step 7 : Stop.

Output :

Enter the length and breadth of a rectangle:

3 4

Area of rectangle : 12.0

Enter the base and height of a triangle:

4 5.0

Area of triangle : 20.0

Enter the radius of circle.

Area of circle : 153.86

12/01/24

OUTPUT :

```
adithya@adi82:~/adi$ javac Main.java
adithya@adi82:~/adi$ java Main
Rectangle: The area is : 50.0
Triangle: The area is : 16.0
Circle: The area is : 113.03999999999999
Adithya Ravikeerthi 1BM22CS020
```


LAB-5: BANK ACCOUNT DETAILS

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.

→ Bank - current & savings account.

Develop a Java program to create a class Bank
that maintains 2 kinds of account

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

```
class Account {
```

```
    String cust_name;
```

```
    int aeno;
```

```
    String acc_type;
```

```
    double balance;
```

```
    public Account (String cust_name, int  
        aeno, String acc_type) {
```

```
        this.cust_name = cust_name;
```

```
        this.aeno = aeno;
```

```
        this.acc_type = acc_type;
```

```
        this.balance = 0;
```

```
}
```

```
public void displayBal () {
```

```
    System.out.println ("Account number: " + aeno);
```

```
    System.out.println ("Customer name: " + cust_name);
```

```
    System.out.println ("Account type: " + acc_type);
```

```
    System.out.println ("Balance: " + balance);
```

```
}
```

```
class Current extends Account {
```

```
    double min_balance, service_charge;
```

```
    Current (String cust_name, int aeno) {
```

```
        super (cust_name, aeno, "Current");
```

```
        this.min_balance = 500;
```

```
        this.service_charge = 20;
```

```
}
```

```
public void withdraw (double amt) {
```

```
    if (balance - amt) >= min_balance {
```

```
        balance -= amt;
```

```
        System.out.println ("Withdrawal
```

```
successful! Current Balance: " + balance)
```

```
}
```

```
else {
```

```
    balance -= service_charge;
```

```
}
```

```

class Savings extends Account {
    double interest_rate;
    Savings (String cust_name, int acc_no) {
        super(cust_name, acc_no, "Savings");
        this.interest_rate = 0.5;
    }
    public void DepositInterest() {
        balance += balance * interest_rate;
        System.out.println("Interest Deposited");
        currentBalance = "" + balance;
    }
    public void compoundInterest(double initial,
                                  int time) {
        double ci = initial * Math.pow((1+interest_rate), time);
        balance += ci;
        System.out.println("Compound Interest is applied. Current Balance: " + balance);
    }
}

public class Bank {
    public static void main (String [] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Choose account type");
        System.out.println("1. Savings 2. Current");
        System.out.println("Enter choice 1 or 2");
        int choice = in.nextInt();
        System.out.println("Enter customer name");
        String cust_name = in.next();
        System.out.println("Enter account number");
        int accno = in.nextInt();
        System.out.println("Enter initial balance");
        if (choice == 1) {
            Savings savAcc = new Savings(cust_name, accno);
        }
    }
}

```

```
double initial_balance = in.nextDouble();
savAcc.balance = initial_balance;
System.out.println("Enter withdrawal amount");
double withdrawl = in.nextDouble();
savAcc.balance = withdrawl;
System.out.println("Withdrawal successful. Current balance = " +
    savAcc.balance);
System.out.println("Enter time to calculate compound interest:");
int time = in.nextInt();
savAcc.compoundInterest(initial_balance,
    time);
savAcc.displayBal();
```

}

~~else if (choice == 2) {~~

```
Current curAcc = new Current(acc_name,
    accno);
System.out.println("Enter initial balance");
double initial_balance = in.nextDouble();
curAcc.balance = initial_balance;
System.out.println("Enter withdrawal amount");
double amt = in.nextDouble();
curAcc.withdrawl(amt);
curAcc.levyWithChrg();
curAcc.displayBal();
```

~~else {~~

```
System.out.println("Invalid choice");
```

}

}

}

Flowchart
 Step 1: Start.
 Step 2: Initialize variable cust_name, acc_no,
 acc_type, balance.
 Step 3: Input = "Enter customer name "+cust_name
 Step 4: Input = "Enter account number "+acc_no
 Step 5: Input = "Enter account type "+acc_type
 Step 6: Input = "Enter balance "+balance
 Step 7: Print "Enter discount type =
 1. Savings 2. Current"
 Step 8: Input = "Enter choice 1 or 2"
 Step 9: If (choice == 2)
 Step 10: Read choice
 Step 11: If (choice == 1){
 Input = "Enter initial balance"
 initial_balance = input
 Input = "Enter withdrawal amount"
 withdrawal = input
 Input = "Initial Input" Enter minimum
 balance "+ min.
 If (initial_balance - withdrawal >= min){
 balance -= withdrawal;
 }
 Print "Current Balance "+balance
 Input = "Enter interest_rate "+rate
 Input = "Enter time "+time
 ci = initial_balance * power((1 + interest_rate / 100), time)
 - initial_balance;
 Print "At compound Interest : "+ci
 Print "Deposit with Interest "+
 (balance + balance * interest_rate / 100)

Step 11: Else if (choice == 2){
 Print "Input" Enter initial_balance
 balance = initial_balance;
 Input = "Enter withdrawal amount"
 withdrawal = input
 If (balance - withdrawal >= min){
 balance -= withdrawal;
 }
 Print "Current Balance "+balance

Step 12: Else
 Print "Invalid choice"

Step 13: Stop

Output:
 Choose account type
 1. Savings 2. Current
 Enter choice 1 or 2
 Enter customer_name
 ab
 Enter account number
 00
 Enter initial balance
 10000
 Enter withdrawal amount
 1000
 Current balance : 9000.0
 Enter time(in years) for compound interest
 calculation.
 3
 Compound interest applied : 32750.0

```
adithya@adi82:~/adi$ javac AccountTest.java
```

```
adithya@adi82:~/adi$ java AccountTest
```

Current Account Details:

Account Balance: \$5000.0

Deposit of \$2000.0 successful.

Account Balance: \$7000.0

Interest of \$140.0 deposited.

Account Balance: \$7140.0

Withdrawal of \$3000.0 successful.

Account Balance: \$4140.0

Account Balance: \$4140.0

Savings Account Details:

Account Balance: \$1500.0

Deposit of \$500.0 successful.

Account Balance: \$2000.0

Interest of \$50.0 deposited.

Account Balance: \$2050.0

Withdrawal of \$1000.0 successful.

Account Balance: \$1050.0

Account Balance: \$1050.0

Adithya Ravikeerthi 1BM22CS020

LAB-6: CALCULATION OF MARKS

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.

Sri
19/01/24

- 1) Create package (IE, SEE & import the package on main.java file.

```
package IE;
import java.util.Scanner;
public class student {
    public int sem;
    public String usn;
    public String name;
    public void accept()
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter usn = ");
        usn = in.nextLine();
        System.out.print("Enter name = ");
        name = in.nextLine();
        System.out.print("Enter sem = ");
        sem = in.nextInt();
    }
}
```

↳

```
package IE;
public class Internal {
    public int marks[] = new int[5];
}
```

↳

```
package SEE;
import IE.student;
public class External extends student {
    public int ext = new int[5];
}
```

```

Main.java
import java.util.*;
import BEE.*;
import CIE.*;
public class Main{
    public static void main(String args[]){
        int final[] = new int[5];
        Scanner in = new Scanner(System.in);
        System.out.println("Enter n = ");
        int n = in.nextInt();
        BEE.External end[] = new BEE.External[n];
        CIE.External in[] = new CIE.External[n];
        for(int i=0;i<n;i++){
            end[i] = new BEE.External();
            in[i] = new CIE.Internal();
            System.out.println("Enter details of "
                + (i+1));
            end[i].accept();
            for(int j=0;j<5;j++){
                System.out.println("Course "+(j+1));
                in[i].marks[j] = in.nextInt();
                end[i].ext[j] = in.nextInt();
                final[j] = in[i].marks[j] +
                    end[i].ext[j];
            }
            System.out.println("Final marks
                of "+end[i].name);
        }
        for(int k=0;k<5;k++){
            System.out.println("Course "
                + (k+1) + " = " + final[k]);
        }
    }
}

```

Algorithm:

```
Step 1: #include <iostream.h>
Step 2: Initialize variable name, usn, sem,
        marks[5], ext[5], final[5]
Step 3: Function: void accept()
        {
            Input: "Enter usn ", usn
            Input: "Enter name ", name
            Input: "Enter sem ", sem
        }
Step 4: Print "Enter no. of students"
Step 5: Read n
Step 6: Create structure of name, usn, sem
& name it as st
Step 7: for(i=0; i<n; i++)
        st[i].accept()
        {
            Print "Enter internal & external
            marks respectively"
            for(j=0; j<5; j++)
            {
                Read st[i].marks[j]
                Read st[i].ext[j]
                st[i].final[j] = st[i].marks[j]
                + st[i].ext[j];
            }
        }
Step 8: print "Final marks"
        for(i=0; i<n; i++)
        {
            print st[i].name
            for(j=0; j<5; j++)
            {
                print "Final course " + (j+1)
                print st[i].final[j]
            }
        }
Step 9: #endif
```

Output:

```
Enter n=1
Enter details of 1
Enter usn : 15
Enter name : jswrf
Enter sem : 2
Enter internal external marks
Course 1
23 89
Course 2
23 9
Course 3
45 78
Course 4
34 90
Course 5
25 90
Final marks
Course 1 = 56
Course 2 = 16
Course 3 = 61
Course 4 = 62
Course 5 = 62
```

8
020284

OUTPUT :

```
adithya@adi82:~/java$ javac FinalMarks.java
^[[Aadithya@adi82:~/java$ java FinalMarks
Enter n:
1
Enter details 1
Enter the details:
Adithya
20
3
Enter im and sm of sub 1
50
50
Enter im and sm of sub 2
5050
50
Enter im and sm of sub 3
50
50
Enter im and sm of sub 4
50
50
Enter im and sm of sub 5
50
50
Final marks of 20
Course 1 = 100
Course 2 = 5100
Course 3 = 100
Course 4 = 100
Course 5 = 100
Adithya Ravikeerthi 1BM22CS020
```

LAB-7: EXCEPTION HANDLING

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 cases both father and son’s age and throws an exception if son’s age is >=father’s age.

Program:

Write a java program that demonstrates
exceptional handling of inheritance tree.
class "father" & derived class "son".

→ class wrongAge extends Exception {
 public wrongAge()
 super("age cannot be negative");
 }
 class input extends Exception {
 public input()
 super("wrong input");
 }
 class Father {
 public int age;
 Father(int age) throws wrongAge {
 if (age > 0)
 throw new wrongAge();
 this.age = age;
 }
 class Son extends Father {
 int s-age;
 Son(int f-age, int s-age) throws wrongAge,
 input {
 super(f-age);
 if (f-age < 0 & s-age < 0)
 throw new wrongAge();
 else if (f-age <= s-age) {
 throw new input();
 }
 this.s-age = s-age;
 }
 }
 }
 public class T {
 public static void main(String[] args) {
 // code
 }
 }
}

```

try {
    Father f = new Father(40);
    System.out.println("Father's age: " + f.age);
    Son s = new Son(10, 50);
    System.out.println("Son's age: " + s.age);
} catch (WrongAge e) {
    System.out.println(e.toString());
}
catch (Input ae) {
    System.out.println(ae.toString());
}
}

```

Algorithm:

Step 1: Start

Step 2: Initialize variable f-age, s-age

Step 3: Create user defined exception

class WrongAge extends Exception

public WrongAge()

{ super("Age cannot be negative") }

Step 4: Create another user defined exception

class Input extends Exception

public Input()

{ super("Wrong input") }

Step 5: Create class Father

public int age;

Father(int age) throws WrongAge {

if (age > 0) {

throw new exception WrongAge();

}

}

This.age = age;

Step 6: Create class Son extends Father {

int s-age

Son(int f-age, int s-age) throws WrongAge, Input {

super(f-age);

```
adithya@adi82:~/java$ javac J.java
```

```
adithya@adi82:~/java$ java J
```

Father's age: 40

Input: Wrong input

Adithya Ravikeerthi 1BM22CS020

```
adithya@adi82:~/java$ javac J.java
```

```
adithya@adi82:~/java$ java J
```

Father's age: 40

Son's age: 30

```
adithya@adi82:~/java$ javac J.java
```

```
adithya@adi82:~/java$ java J
```

Father's age: 40

WrongAge: Age cannot be negative

LAB-8: MULTITHREADING

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.

Write a program which creates 2 threads where one is executed every 10 second & another for 2 second.

class One extends Thread {

```
public void run() {
    int i=0;
    while(i<2) {
        i++;
        try {
            System.out.println("BMS College of
Engineering");
            Thread.sleep(1000);
        } catch(Exception e) {
            System.out.println(e.toString());
        }
    }
}
```

class Two extends Thread {

```
public void run() {
    int i=0;
    while(i<2) {
        i++;
        try {
            System.out.println("CSE");
            Thread.sleep(2000);
        } catch(Exception e) {
            System.out.println(e.toString());
        }
    }
}
```

```
public class I {
    public static void main(String[] args) {
        One t1 = new One();
        Two t2 = new Two();
        t1.start();
        t2.start();
    }
}
```

Algorithm:

Step 1: start

Step 2: Initialize variable $i = 0$

Step 3: Construct a class One & Two which extends Thread.

Step 4: Under class One create method run(),
under which, while ($i < 5$) {
 $i++$
 try {

```
        System.out.println("BMS College of Engineering");
        Keep Thread in sleep(1000)
```

```
} catch (Exception e) {
```

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

Step 5: Under class Two create method run()
under which, while ($i < 5$) {
 $i++$
 try {

```
        System.out.println("GE")
```

```
} catch (Exception e) {
```

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

Step 6: Create a main class

```
    New class One t1 = new One();
```

```
    New class Two t2 = new Two();
```

```
    t1.start()
```

```
    t2.start()
```

```
adithya@adi82:~/java$ javac Main.java
```

```
adithya@adi82:~/java$ java Main
```

BMS COLLEGE OF ENGINEERING

Adithya Ravikeerthi 1BM22CS020

CSE

CSE

CSE

CSE

CSE

BMS COLLEGE OF ENGINEERING

BMS COLLEGE OF ENGINEERING

BMS COLLEGE OF ENGINEERING

BMS COLLEGE OF ENGINEERING

LAB-9:

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 Arithmetic Exception Display the exception in a message dialog box.

9. Write a program that creates a user interface to perform integer divisions. The user enter 2 no.s in the text field, Num1 & Num2. The division of Num1 & Num2 is displayed in the result field when Divide button is clicked.

this
age)
JFrame: a window that has decorations such as border, title & supports button

FlowLayout(): arranges components in a left-to-right flow, much like lines of text in para.

Code:

```
import java.awt.*;  
import java.awt.event.*;  
public class swingDemo1 {  
    swingDemo1() {  
        JFrame jfrm = new JFrame("Divide App");  
        jfrm.setSize(275, 150);  
        jfrm.setLayout(new FlowLayout());  
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
  
        JLabel lab = new JLabel("Enter divisor &  
        dividend");  
        JTextField ablabtf = new JTextField(8);  
        JTextField btjf = new JTextField(8);  
        JButton button = new JButton("Calculate");  
  
        JLabel ans = new JLabel();  
        JLabel alab = new JLabel();  
        JLabel blab = new JLabel();  
        JLabel anslab = new JLabel();
```

```

jfrm.add(ev);
jfrm.add(jlab);
jfrm.add(btjf);
jfrm.add(btjf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(alslab);

ActionListener I = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        System.out.println("Action event from " + evt.getActionCommand());
    }
};

btjf.addActionListener(I);
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(btjf.getText());
            int b = Integer.parseInt(btjf.getText());
            int ans = a / b;
            alab.setText("A=" + a);
            blab.setText("B=" + b);
            alslab.setText("Ans=" + ans);
        } catch (NumberFormatException e) {
            alab.setText("Enter only integers!");
            btjf.setText("");
            alslab.setText("");
            evr.setText("Enter only integers!");
        }
    }
});

```

```

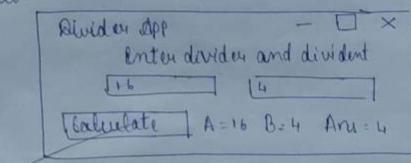
catch (ArithmaticException e) {
    alab.setText("0");
    blab.setText("0");
    alslab.setText("0");
    evr.setText("B should be non-zero!");
}
}

jfrm.setVisible(true);
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new swingDemo();
        }
    });
}

```

Output:



Functions:

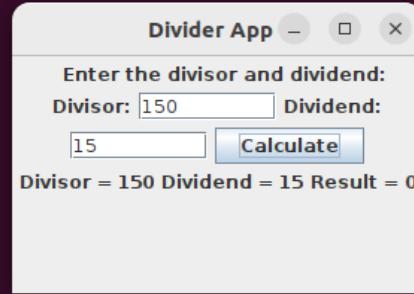
JLabel: object that can display either text or an image.

ActionListener: The listener interface for receiving action events.

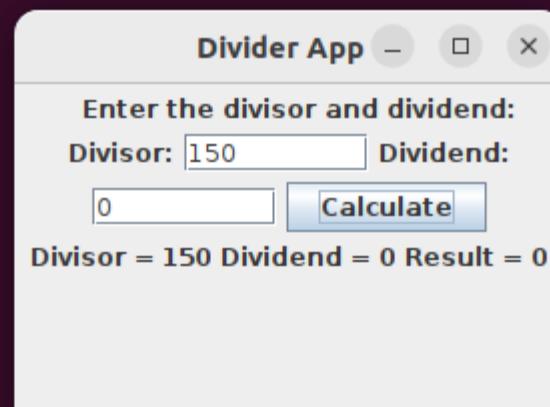
ActionEvent: A semantic event which indicates that a component defined action occurred.

Output:

```
adithya@adi82:~/java$ javac SwingDemo.java  
adithya@adi82:~/java$ java SwingDemo
```



```
adithya@adi82:~/java$ java SwingDemo
```



```
adithya@adi82:~/java$ java SwingDemo
```

