

LAB - 1

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

S1 - Start

S2 - Read  $a, b, c$

S3 - Initialize  $d \leftarrow (b * b) - (4 * a * c)$

S4 - Initialize  $r \leftarrow b / 2 * a$

S5 - if  $d > 0$  go to S6 else go to S8

S6 -  $r_1 = r + (\sqrt{d}) / 2 * a$  and  $r_2 = r - (\sqrt{d}) / 2 * a$

S7 - Print roots are real and distinct, first root  $r_1$  and next  $r_2$

S8 - If  $d = 0$  go to ~~S7~~ S9 else go to S10

S9 - print roots are real and equal,  $-r$

S10 -  $d = -d$

S11 -  $iim = \sqrt{-d} / 2 * a$

S12 - print roots are imaginary, first root is  $r + iim$

and the second root is  $r - iim$

S13 - Stop.

Flowchart

Start

Input a,b,c

$$D \leftarrow \sqrt{(b*b) - 4*a*c}$$

$$r_1 \leftarrow (-b+d) / (2*a)$$

$$r_2 \leftarrow (-b-d) / (2*a)$$

Print r<sub>1</sub> and r<sub>2</sub>

Stop

Code

```

import java.util.Scanner;
public class Quadratic {
    public static void main (String [] args) {
        Scanner scanner = new Scanner (System.in);
        System.out.println ("x: Enter coefficients a, b, c:");
        double a = scanner.nextDouble ();
        double b = scanner.nextDouble ();
        double c = scanner.nextDouble ();
        double discriminant = b * b - 4 * a * c;
        if (discriminant < 0) {
            System.out.println ("x: " "no real solution.");
        }
        else {
            double sqrtDiscriminant = Math.sqrt (discriminant);
            double root1 = (-b + sqrtDiscriminant) / (2 * a);
            double root2 = (-b - sqrtDiscriminant) / (2 * a);
            System.out.println ("The solutions are " + root1 +
                " and " + root2);
        }
        scanner.close ();
    }
}

```

Output

Enter coefficients a, b, c:

1 5 6

The solutions are -2.0 and -3.0,

The screenshot shows a terminal window titled "1: Java Debug Console". The window has tabs for PROBLEMS (17), OUTPUT, TERMINAL (which is selected), and more. The terminal content is as follows:

```
/Contents/Home/bin/java -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:56235 --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -Dfile.encoding=UTF-8 -cp "/Users/adyaprakasha/Library/Application Support/Code/User/workspaceStorage/cfbc9c7f24e20b2c0e1ac38bf24bd34d/redhat.jdt_ws/java_895310d3/bin" quadratic
Enter the Co-Effcient a
2
Enter the Co-Effcient b
3
Enter the Co-Effcient c
4
THANK YOU FOR ENTERRING THE CO-EFFCIENTS
ROOTS ARE IMAGINARY
ROOTS ARE -0.75+i1.1989578808281798
ROOTS ARE -0.75-i1.1989578808281798
bash-5.0$
```

LAB - 2

- ② Develop a Java program to create a class Student with members id, name, an array credits and an array marks. Include methods calculate SGPA of a student.

Algorithm

Define the Student Class:

Create a class named Student

Define the members:

id, name, an array of credits and an array of marks.

Include methods to accept and display details

Include a method to calculate the SGPA of a student

Create Java Program:

Prompt the user to enter the details of the student (id, name, credits and marks).

Initialize a variable to store the total credit and another variable to store the total grade points

Calculate the grade points for each subject using the provided formula.

Calculate the SGPA of the student using the provided formula.

Display the details of the student and the calculated SGPA

Flowchart

Start

Define the student class.

Create a class named Student

Define the members :

Use name, an array of credits and an array for marks

Include methods to calculate and display details.

Include a method to calculate CGPA of a student

Create Java Program :

Prompt the user to enter the details of the student (name, name, credits and marks).

Calculate the CGPA of the student using the provided method

Display the details of the student and calculate CGPA

End

## Code

```
import java.util.Scanner;  
public class student  
{  
    private String USN;  
    private String name;  
    private int n;  
    private double SPPA=0;  
    private int totalCredits = 0;  
    private double marks[];  
    private double credits[];  
    Scanner ss = new Scanner (System.in);
```

## void Details ()

{

System.out.println ("x: "Enter USN of the student");

USN = ss.nextLine ();

System.out.println ("x: "Enter Name of the student");

~~name = ss.nextLine ();~~

System.out.println ("x: "Enter no. of students");

n = ss.nextInt ();

ss.nextLine ();

credits = new int [n];

marks = new double [n];

System.out.println ("Enter details of the subject: ");

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

System.out.println ("Enter credits allotted to the subject "+  
(i+1));

credits [i] = ss.nextInt ();

System.out.println ("Enter marks in the subject "+ (i+1));  
marks [i] = ss.nextDouble ();

ss.nextLine ();

Calculate (credits [i], marks [i], i);

y

void calculate (int credits, double mark, int j)

{

totalCredits = totalCredits + credit;

if (mark >= 90 && mark <= 100)

SGPA = SGPA + (10 \* credit);

else if (mark >= 80 && mark <= 89)

SGPA = SGPA + (9 \* credit);

else if (mark >= 70 && mark <= 79)

SGPA = SGPA + (8 \* credit);

else if (mark >= 60 && mark <= 59)

SGPA = SGPA + (7 \* credit);

else if (mark >= 50 && mark <= 49)

SGPA = SGPA + (6 \* credit);

else if (mark >= 70 & mark <= 40)

SGPA = SGPA + (5 \* credit);

else

{

System.out.println ("Failed in subject " + (j+1));

SGPA = SGPA / 10;

}

}

void Display()

{

System.out.println (x: "Details of the student");

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

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

System.out.println ("SGPA of student : " + (SGPA / total credit));

}

}

Class Main

{

public static void main (String [] args)

{

Student s1 = new Student ();

s1.Details ();

s1.Display();

}

}

Output:

Enter USN of the student :

IRML8CS078

Enter Name of student

Rahul

Enter no. of subjects

2

\* Enter details of the subjects : \*

Enter credits allotted to the subject 1

3

Enter the marks in the subject 1

85

Enter credits allotted to the subject 2

3

Enter the marks in the subject 2

78

Details of the Student

USN: IRML8CS078

Name: Rahul

SGPA of student : 8.5

```
PROBLEMS ⚡ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ CodeLens (Launch) - Main
bash-5.0$ /Library/Java/JavaVirtualMachines/jdk-15.jdk/Contents/Home/bin/java -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:56291 --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -Dfile.encoding=UTF-8 -cp "/Users/adityaprakasha/Library/Application Support/Code/User/workspaceStorage/cfbc9c7f24e20b2c0e1ac38bf24bd34d/redhat.java/jdt_ws/java_895310d3/bin"
Main
Enter USN of the student
1BM17CS088
Enter Name of the student
james bond
Enter no of subjects
3
*Enter details of the subjects:*
Enter credits allotted to the subject 1
3
Enter marks in the subject 1
86
Enter credits allotted to the subject 2
4
Enter marks in the subject 2
88
Enter credits allotted to the subject 3
3
Enter marks in the subject 3
90
Details of the Student
USN: 1BM17CS088
Name :james bond
SGPA of Student 9.3
bash-5.0$
```

LAB-3 => I

- 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 ~~String~~ a toString() method that could display the complete details of the book. Develop a Java program to create ~~an~~ book object.

Algorithm

Define the Book Class:

Create a class named Book.

Define the attributes:

name, author, price, num-pages.

Include a constructor to initialize the attributes

Include methods to set and get the details of the book object.

Include a toString() method to display the complete details of the book.

~~Prompt the user like enter the number of book objects, n.~~

~~Initialize a counter, i to 1.~~

Loop Start:

~~Inside the loop, create a new Book object.~~

~~Set the details of the Book object using user input or predefined values.~~

Display the details of the Book object using the toString() method:

| Increment the counter, i

| Loop end.

## Flowchart

Start

Define the Book class

Define the attribute - name, author, price, num-page.

Create a constructor to initialize the attributes

Create methods to set and get details of the book objects

Create a toString() method to display the complete details of the book

Create N Book object

Initialize a counter (i) to 1

loop start

Create a new Book Obj

set the details of the Book Obj

Display the details of the Book Obj

Increment the counter (i)

Loop end

end

Code.

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

```
class book
```

```
{
```

```
String booktitle;
```

```
String author;
```

```
int no-of-pages;
```

```
double price;
```

```
Scanner sc = new Scanner (System.in);
```

```
book()
```

```
{
```

```
System.out.print ("S: Enter book title :");
```

```
booktitle = sc.nextLine();
```

```
System.out.print ("S: Enter the author name :");
```

```
author = sc.nextLine();
```

```
System.out.print ("S: Enter the price :");
```

```
price = sc.nextDouble();
```

```
System.out.print ("S: Enter the pages :");
```

```
no-of-page = sc.nextInt();
```

~~public String toString()~~

```
{
```

```
return ("Book name = " + booktitle + " Author = " + author + "
```

```
price = " + price + " Pages = " + no-of-pages);
```

```
}
```

```
y
```

Class Books

{

public static void main (String [] args)

{

int n, i;

Scanner in = new Scanner (System.in);

System.out.println ("Enter number of books : ");

n = in.nextInt ();

book [] b = new book [n];

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

{

System.out.println ("Enter details of Book " + (i+1));

b [i] = new book ();

}

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

{

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

}

}

Output:

Enter no. of books : 2

Enter details of Book : 1

Enter book title : jubar

Enter the author name : kumar

Enter the price : 250

Enter the pages : 55

Enter details of Book : 2

Enter the book title : gunar

Enter the author name : Kumar

Enter the price : 650

Enter the pages : 95

~~Book name = jubar Author = kumar Price = 250.0 Page = 55~~

~~Book name = gunar Author = Kumar Price = 650.0 Page = 95~~

```
bash-5.0$ /Library/Java/JavaVirtualMachines/jdk-15.jdk/Contents/Home/bin/java -agentlib:jdwp=tra  
nsport=dt_socket,server=n,suspend=y,address=localhost:52179 --enable-preview -XX:+ShowCodeDetails  
InExceptionMessages -Dfile.encoding=UTF-8 -cp "/Users/adityaprakasha/Library/Application Support/  
Code/User/workspaceStorage/cec6510638e2882570735580372d9ec6/redhat.java/jdt_ws/jdt.ls-java-projec  
t/bin" Books  
Enter number of books: 2  
Enter details of Book: 1  
Enter book title: invisible man  
Enter the author name: raplh ellison  
Enter the price: 300  
Enter the pages: 400  
Enter details of Book: 2  
Enter book title: native son  
Enter the author name: richard wright  
Enter the price: 350  
Enter the pages: 500  
Book name = invisible man Author = raplh ellison Price = 300.0 Pages = 400  
Book name = native son Author = richard wright Price = 350.0 Pages = 500  
bash-5.0$ []
```

(ii) 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 contains the method printArea() that prints the area of the given shape.

### Algorithm

Create an abstract class Shape with the following:

Two integers

An empty printArea()

Create a class Rectangle that extends Shape and implements printArea():

Define the printArea() method to calculate and print the area of a rectangle.

Create a class Triangle that extends Shape and implements printArea():

Define the printArea() method to calculate and print the area of a triangle.

Create a class Circle that extends Shape and implements printArea():

Define the printArea() method to calculate and print the area of circle.

## Flowchart

Start

Create abstract class Shape

- two integers
- empty method printArea()

Create class Rectangle

- extends Shape
- implements printArea()

Create class Triangle

- extends Shape
- implement printArea()

Create class Circle

- extends Shape
- implement printArea()

End

Code

```
import java.util.*;
import java.lang.Math.*;
abstract class shape
{
    public int a;
    public int b;
    abstract public void printArea();
    Scanner s = new Scanner (System.in);
}
```

class rectangle extends shape {

```
    public void printArea()
        System.out.print (s: "Please enter length and breadth of
rectangle : ");

```

```
float a = s.nextFloat ();

```

```
float b = s.nextFloat ();

```

```
float area = a * b;

```

~~```
System.out.println ("Area = " + area + " sq. units");
```~~

```
}
```

class triangle extends shape {

```
    public void printArea()

```

~~```
        System.out.print (s: "Please enter three sides of the
triangle : ");

```~~

```
float a = s.nextFloat ();

```

```
float b = s.nextInt();
```

```
float c = s.nextInt();
```

```
float d = (a+b+c)/2;
```

```
double area = Math.sqrt(d*(d-a)*(d-b)*(d-c));
```

```
System.out.println ("Area = " + area + " Sq. units");
```

```
}
```

```
}
```

class circle extends shape

```
public void printArea () {
```

```
System.out.print ("S: " Please enter radius of circle : ");
```

```
float r = s.nextInt();
```

```
float area = 22/7 * r * r;
```

```
System.out.println ("Area = " + area + " Sq. units");
```

```
}
```

```
y
```

class ShapeDemo {

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

```
Shape r = new rectangle ();
```

```
Shape t = new triangle ();
```

~~Shape c = new circle ();~~

~~for (int i=0; i<100; i++)~~

~~System.out.println (x: "In 1) Triangle In 2) Rectangle In~~

~~3) Circle In ")~~

System.out.println (x: "Enter your choice : ");

Scanner s = new Scanner (System.in);

```
int ch = sc.nextInt();
```

```
switch(ch)
```

```
{
```

```
    Case 1: t.printArea();
```

```
    break;
```

```
    Case 2: t.printArea();
```

```
    break;
```

```
    Case 3: c.printArea();
```

```
    break;
```

```
    default:
```

```
        System.out.println("u: \"Invalid choice\"");
```

```
y
```

```
3
```

```
y
```

```
b
```

### Output:

1) Triangle

2) Rectangle

3) Circle

Select your choice:

1

Please enter 3 sides of Triangle : 3

4

6

Area = 5.332682251925386 sq. units.

Enter your choice:

2

Please enter length and breadth of rectangle : 4

9

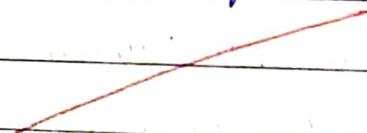
Area = 36.0 Sq. units.

Enter your choice:

3

Please enter radius of circle : 6

Area = 108.0 Sq. units.



```
bash-5.0$ /Library/Java/JavaVirtualMachines/jdk-15.jdk/Contents/Home/bin/java -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:61892 --enable-preview -XX:+ShowCodeDetails InExceptionMessages -Dfile.encoding=UTF-8 -cp "/Users/adityaprakasha/Library/Application Support/Code/User/workspaceStorage/cfbc9c7f24e20b2c0e1ac38bf24bd34d/redhat.java/jdt_ws/java_895310d3/bin" Shapedemo

1)Triangle
2)Rectangle
3)Circle

Enter your choice:
1
Please enter three sides of triangle: 3
4
5
Area=6.0sq.units

1)Triangle
2)Rectangle
3)Circle

Enter your choice:
2
Please enter length and breadth of rectangle: 2
3
Area=6.0sq.units

1)Triangle
2)Rectangle
3)Circle

Enter your choice:
4
Invalid choice

1)Triangle
2)Rectangle
3)Circle

Enter your choice:
3
Please enter radius of circle: 4
Area=48.0sq.units

1)Triangle
2)Rectangle
3)Circle

Enter your choice:

```

LAB-4

(4)

Develop a Java program to create a class Bank that maintains two kind of accounts for its customers, one called savings and the other current account. The saving 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. If 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 Curr-accnt. and Sav-accnt to make them more specific to their requirements. include the necessary methods in order to achieve the following tasks:

- Accept deposits from customers and update the balance
- Display the balance
- Compute and deposit interest
- Permit withdrawal and update the balance
- Check for the minimum balance impose penalty if necessary and update the balance.

Algorithm

Start

Create a class Account with customer name, account number and type of account.

Derive the class Curr-accnt and Sav-accnt from Account to make them more specific to their requirements.

In Curr-accnt add cheque book facility, minimum balance +

and charge

In Saw-eet add compound interest and withdrawal facilities

Create a class Bank with the following methods :

- add amount
- deposit
- display balance
- compute interest
- withdraw
- check minimum balance and impose penalty if necessary

Implement the add amount method to create a new amount and add it to the bank's list of amounts.

Implement the deposit method to account a deposit from a customer and update the balance of the corresponding account.

Implement the display balance method to show the current balance of a given account.

Implement the compute interest method to calculate and deposit interest to the savings account.

Implement the withdraw method to allow a customer to withdraw money from their account and update the balance accordingly.

Implement the check minimum balance method to check if the balance of a current account falls below the minimum balance and impose a penalty if necessary.

End.

Flowchart

Start

## Create class Account

- customer name
- amount number
- type of account

## Create class Current-Acc

- cheque book facility
- minimum balance
- Service charge

## Create class Sav-Acc

- Compound interest
- with drawal facilities

## Create class Bank

- add amount
- deposit
- display balance
- Compute interest
- withdraw
- check minimum balance
- impose penalty

End.

Code

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

```
class Account
```

```
{
```

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

```
String custName, accType;
```

```
long accNumber;
```

```
double balance = 9876.5;
```

```
void Accept ()
```

```
{
```

```
System.out.println ("Enter name");
```

```
custName = in.nextLine();
```

```
System.out.println ("Enter Account number");
```

```
accNumber = in.nextInt();
```

```
}
```

```
void deposit ()
```

```
{
```

~~int dep;~~

```
System.out.println ("Enter the amount to be deposited")
```

```
dep = in.nextInt();
```

```
balance += dep;
```

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

```
}
```

```
void withdraw ()
```

```
{
```

int withdraw

System.out.println ("x: " + "Enter the amount you want to withdraw");

withdraw = in.nextInt();

balance = withdraw;

System.out.println ("Balance = " + balance);

}

} class

class CurrentAccount extends Account

{

void penalty ()

{

if (balance < 2000)

{

balance = 400

System.out.println ("x: " + "400 penalty for maintaining less than minimum balance.");

System.out.println ("Balance = " + balance);

}

}

~~class SavingsAccount extends Account~~

{

void interest ()

{

double i;

$i = balance * 0.05;$

$balance += i;$

`System.out.println ("Interest = " + i);`

`System.out.println ("Total Balance = " + balance)`

y

class Bank

{

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

{

`Scanner sc = new Scanner (System.in);`

`System.out.println ("1: Enter your choice 1 n1. Savings  
Account in 2. Current Account");`

`int choice = sc.nextInt();`

`Current C = new Current ();`

`SavAct S = new SavAct ();`

`if (choice == 2)`

{

~~C.Accept();~~

~~System.out.println ("1: Enter your choice 1 n1. Deposits  
in 2. withdraw");~~

~~int n = sc.nextInt();~~

~~switch (n)~~

{

~~case 1 :~~

~~C.deposit();~~

~~break;~~

}

case 2 : {

C. withdrawal();

C. penalty();

break;

}

~~default:~~ System.out.println("x: " "Wrong choice");

y

y

if (choice == 1){

S. Accept();

System.out.println("x: " "Enter your choice b/w.

Deposit, M2, Withdraw ");

int n = sc.nextInt();

switch (n) {

case 1 : {

S. deposit();

S. interest();

break;

}

case 2 : {

S. withdrawal();

break;

}

default: System.out.println("x: " "Wrong choice");

}

} } }

Output

Enter your choice

1) Savings Account

2) Current Account

1

Enter name

Balaji

Enter account number

154646455

Enter your choice

1. Deposit

2. Withdraw

1

Enter the amount to be deposited

590000

Balance = 599876.5

Balance = 29993.825

Total Balance = 629870.325

Cdt

Dr

2

PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL 2: Java Debug Console + □ ×

```
bash-5.0$ /Library/Java/JavaVirtualMachines/jdk-15.jdk/Contents/Home/bin/java -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:61844 --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -Dfile.encoding=UTF-8 -cp "/Users/adityaprakasha/Library/Application Support/Code/User/workspaceStorage/cfbc9c7f24e20b2c0e1ac38bf24bd34d/redhat.java/jdt_ws/java_895310d3/bin"
Bank
Enter your choice
1. Savings Account
2.Current Account
1
Enter name
rahul
Enter Account number
758595858
Enter your choice
1. Deposit
2. Withdraw
1
Enter the amount to be deposited
1000000
Balance = 1009876.5
Interest = 50493.825000000004
Total Balance = 1060370.325
bash-5.0$
```

Lab-5

## (b) Package program

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 derived class of Student.

The class has an array that stores the SEE marks scored in five courses of the current semester of the student.

Import the two packages via file that declares the final marks of n students in all five courses.

## Algorithm

Create the CIE Package:

- Define the 'Student' class in the CIE package with members like 'usn'; name and 'sem'.
- Define the 'Internals' class in the CIE package with an array to store the internal marks scored in five courses of the current semester for the student.

### Create the SEE Package:

- Define the 'External' class in the SEE package, which is a derived class of the 'Student' class.
- The 'external' class should have an array to store the SEE marks scored in five courses of the current semester for the student.

### Export the package:

- \* Export the CIE and SEE package in a file that declares the final marks of all students via all five courses.

### Code

#### Package CIE

#### STUDENT

```
package CIE;  
import java.util.*;  
public class student {  
    public String name;  
    public String roll;  
    public int sem;  
    public void display(){
```

```
        Scanner sc = new Scanner (System.in);
```

```
        System.out.println ("x: " "Name: ");
```

name = sc.nextLine();

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

nm = sc.nextLine();

System.out.println("y: " + nm);

nm = sc.nextLine();

}

y

### Internals

package CIE;

import java.util.\*;

public class Student

public double ~~clm~~ clm[];

public void display()

clm = new double[5];

Scanner c = new Scanner(System.in);

System.out.println("Enter clm marks out of 50:");

for (int i=0; i<5; i++)

clm[i] = c.nextDouble();

y  
3  
3

PACKAGE SEEExternal

package SEE;

import CIE.\*;

import java.util.\*;

public class External extends CIE.Student {

    public double resm[];

    public void display () {

        resm = new double [5];

        Scanner s = new Scanner (System.in);

        System.out.println ("See marks of 5 subjects  
        out of 100:");

        for (int i=0; i<5; i++) {

            resm[i] = s.nextDouble();

        }

    }

}

Main

import CIE;

import SEE;

import java.util.\*;

public class Main {

    public static void main (String [] args) {

        int n;

        Scanner sc = new Scanner (System.in);

PACKAGE SEEExternal

Package SEE;

import CIE.\*;

import java.util.\*;

public class External extends CIE.Student {

    public double reem[];

    public void display(){

        reem = new double[5];

        Scanner s = new Scanner(System.in);

        System.out.println("See marks of 5 subjects  
        out of 100:");

        for (int i=0; i<5; i++) {

            reem[i] = s.nextDouble();

        }

    }

}

Main

import CIE;

import SEE;

import java.util.\*;

public class Main {

    public static void main (String [] args) {

        int n;

        Scanner sc = new Scanner (System.in);

System.out.println ("x: " + total no. of students");

n = sc.nextInt();

CE.Student st [ ] = new CE.Student [n];

CE.~~Student~~ int [ ] = new CE.Batches [n];

SEE.External em [ ] = new SEE.External [n];

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

st [i] = new CE.Student ();

in [i] = new CE.Questions ();

em [i] = new SEE.External ();

st [i].display ();

in [i].display ();

em [i].display ();

System.out.println ("Totalmarks of " + st [i].

Name + "\n");

for (int j = 0; j < 5; j++)

System.out.println (in [i].cine [j] + em [i].

reem [j] / 2);

3

4

5

```
Enter no. of students:  
2  
Name:  
A  
USN:  
1BM17CS005  
Sem:  
2  
Eneter cie marks out of 50:  
48  
47  
46  
48  
44  
SEE marks for 5 subjects out of 100:  
84  
88  
86  
82  
80  
Total Marks of A  
90.0  
91.0  
89.0  
89.0  
84.0  
Name:  
B  
USN:  
1BM19CS192  
Sem:  
5  
Eneter cie marks out of 50:  
43  
41  
39  
37  
45  
SEE marks for 5 subjects out of 100:  
78  
80  
84  
86  
82  
Total Marks of B  
82.0  
81.0  
81.0  
80.0  
86.0
```

Lab - b

Q

Write a program that demonstrate handling of exception 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 takes both father and son's age and throws an exception if son's age is  $\geq$  father's age.

Algorithm

- Define a class called 'Father' :
  - \* Implement a constructor that takes the father's age as input
  - \* Check if the age is less than 0.
  - \* If the age is less than 0, raise a 'WrongAge' exception
  - \* If the age is valid, store it in a class variable.
- Define a class called 'Son' that inherits from 'Father' :
  - \* Implement a constructor that takes the father's age and Son's age as input.
  - \* Call the constructor of the base class ('Father') using 'super()'.

• Start with your own interests, hobbies, passions  
and interests.

• You can also visualize what you expect from them.  
• Now it's time to imagine.

• If you want to succeed with people, imagine

Deliberately creating situations that reflect what you  
would like to receive.

• Another reason why to imagine

• Imagine the person you want to attract, the person you  
want to meet. It's the same idea with visualization  
technique.

• Personal development

• If you want to meet the situation or the person you

• Prefer, then make sure you create situations

• So, deliberately imagine a meeting where other

• People - people of your choice that you would like  
to know. Imagine meeting them, talking to them,  
etc.

- \* Check if the son's age is greater than or equal to the father's age.
  - \* If the son's age is greater than or equal to the father's age raise a 'Wrong Age' exception.
  - \* If the son's age is invalid, store it in a class variable.
- Define a custom exception class called 'WrongAge' to handle valid age inputs.
- In the main part of program:
- \* Prompt the user to input the father's age and son's age.
  - \* Create an instance of the 'Son' class with the provided ages.
  - \* Handle exception:
    - \* If the input cannot be converted to an integer, catch a 'ValueError' and print an error message.
    - \* If a 'WrongAge' exception is raised during object creation, catch it and print the error message.
    - \* If no exception occurs, print the father's age and son's age.

Code

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

```
class WrongAge extends Exception {
```

```
    public WrongAge (String message) {
```

```
        super (message);
```

```
}
```

y.

```
class Father {
```

```
    int fatherAge;
```

```
    public Father (int fatherAge) throws WrongAge {
```

```
        if (fatherAge <= 0) {
```

```
            throw new WrongAge (message: "Father's age  
cannot be negative");
```

```
}
```

    this.fatherAge = fatherAge;

y

y

```
class Son Extends Father {
```

```
    int sonAge;
```

```
    public Son (int fatherAge, int sonAge) throws  
Exception WrongAge {
```

```
        super (fatherAge);
```

```
        if (sonAge >= fatherAge) {
```

            throws new WrongAge (message: "Son's age should  
be less than the father's age");

}

this.sonAge = sonAge;

}

}

public class FatherSon {

public static void main (String [] args) {

Scanner sc = new Scanner (System.in);

System.out.println ("x: Enter the Father's age and  
son's age: ");

int fa = sc.nextInt();

int sa = sc.nextInt();

try {

Son s = new Son (fa, sa);

System.out.println ("Father's age : " + s.fatherAge  
" In Son's age: " + s.sonAge);

}

catch (WrongAge e) {

System.out.println ("Error: " + e.getWrongAge());

}

}

}

Output:

18M18CS078 > LAB - PROGRAM - 7

Enter the Father's age and Son's age :

45 25

Error: Son's age should be less than father's age.

18M18CS078 > LAB - PROGRAM - 7

Enter the Father's age and Son's age :

45 25

Father's age : 45

Son's age : 25.

SSQ

16/12/24

```
[ADITYAs-MacBook-Pro:Lab-7 adityaprakasha$ javac age.java
[ADITYAs-MacBook-Pro:Lab-7 adityaprakasha$ java Main
Enter father's age: 12
Enter son's age: 2
[ADITYAs-MacBook-Pro:Lab-7 adityaprakasha$ java Main
Enter father's age: 2
Enter son's age: 12
Son's age is more than father's age
ADITYAs-MacBook-Pro:Lab-7 adityaprakasha$ ]
```

(8)

### Program 8

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

```
class NewThread implements Runnable {
```

```
String Name;
```

```
Thread t;
```

```
int n;
```

```
NewThread (String threadName, int n) {
```

```
Name = threadName;
```

```
this.n = n;
```

```
t = new Thread (this, Name);
```

```
System.out.println ("New Thread") ++);
```

```
t.start ();
```

```
}
```

```
public void run () {
```

```
try {
```

```
for (int i=0; i<10; i++) {
```

```
System.out.println (Name + ":" + i);
```

```
Thread.sleep (n);
```

```
}
```

```
}
```

```
Catch (InterruptedException e) {  
    System.out.println ("Name : " + e.getMessage());  
}  
}  
}  
  
public class Prog8 {  
    public static void main (String [] args) {  
        NewThread ob1 = new NewThread ("CSE", 2000);  
        NewThread ob2 = new NewThread ("BMS College of  
Engineering", 10000);  
    }  
}
```

### Algorithm

Define class A which extends Thread class:

- Define a class named 'A' that extends Thread
- Initialize instance variables t1 and time to represent time intervals.
- Implement the constructor to set initial values of t1 and time.
- Override the run method to execute the thread logic
- Provide the run method in a while loop to run until t1 reaches time.

- Within the loop print "BMS College Of Engineering" and catch any exception that occur during step.

### Main Class

- Create a class named `th` to contains the main method
- Private the main method , instantiate objects And
- Start both thread using the start method

### Output

BMS College of Engineering

CSE

CSE

CSE

SCE

CSE

BMS College Of Engineering

CSE

CSE

CSE

CSE

CSE

Exiting

```
bash-5.0$ cd /Users/adityaprakasha/Developer ; /Library/Java/JavaVirtualMachines/jdk-11.0.8.jdk/Contents/Home/bin/java -agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:51471 -Dfile.encoding=UTF-8 -cp "/Users/adityaprakasha/Library/Application Support/Code/User/workspaceStorage/cec6510638e2882570735580372d9ec6/redhat.java/jdt_ws/Developer_877bb0be/bin" threadprg
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
Exiting: Thread[cse,5,main]
BMS College of Engineering
|
```

## LAB - 7.

### (a) program 9

WAP that creates a user interface to perform integer division. The user enters two numbers in the text fields num1 and num2. The result of num1 and num2 is displayed in the result field. When the division button is clicked. If num1 is not an integer the program would throw an error NumberFormat exception. If num1 were zero, the program would throw an ArithmeticException. Display the exception in a message.

### Definition

JFrame class - The Java Swing functionalities are the container used for an application is called JFrame.

- It acts like the main window where components like Labels, Buttons, TextField are added to create a GUI.

JFrame.setSize - Resizes this component so that it has width w and height h.

JFrame.setLayout - used to set layout of the container.

Flow Layout - Used to arrange the components in a line default layout of applet or ~~panel~~ panel.

- JFrame.setDefaultCloseOperation - Used to specify one of several options for the close button.
- Table - Create an empty label without any content.
- JText Field - Constructor that creates a new empty text field with the given string and a specified number of columns.

Code -

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

class SwingDemo

{

Swing Demo()

{

```
JFrame jfrm = new JFrame ("Divider App");  
jfrm.setSize (500, 200);  
jfrm.setLayout (new FlowLayout ());  
jfrm.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
```

JLabel jl1 = new JLabel ("Enter the dividend  
JTextField

divident and divisor : ");

JTextField ajtf = new JTextField (8);

JTextField bjtf = new JTextField (8);

JButton button = new JButton ("Calculate");

JLabel l11 = 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);
```

Action Listener : 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(argv[0].getTerm(0));

int b = integer.parseInt(argv[1].getTerm(0));

int ans = a/b;

alab.setTerm("ln A = " + a);

blab.setTerm("ln B = " + b);

anslab.setTerm("ln Ans = " + ans);

}

Catch -(Number Format Exception - E)

{

alab.setTerm(" "));

blab.setTerm(" "));

anslab.setTerm(" "));

}

Catch -(Arithmatic Exception)

{

alab.setTerm(" "));

blab.setTerm(" "));

anslab.setTerm(" "));

err.setTerm("B should be Non zero "));

}

)

);

ifnum.setBoolean(true);

public static void main (String args())

{

Saving • addition • multiply later (new formula)

{  
pari wird nun (S)

{  
heutigen Rest (S);  
}  
});

} . . . . .

### Output

Divider App

Einfere dividend und divisor

15

5

Calculate

Dividend (A) = 15

Divisor (B) = 5

Result = 3

