

PROGRAM 7: 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.

PROGRAM :

```
import java.util.*;  
public class Quadratic  
{  
    public static void main(String args[])  
    {  
        Scanner Sc = new Scanner(System.in);  
        System.out.println("Enter the coefficient of  $(x^2)$ ");  
        int a = Sc.nextInt();  
        System.out.println("Enter the coefficient of  $(x)$ ");  
        int b = Sc.nextInt();  
        System.out.println("Enter the constant");  
        int c = Sc.nextInt();  
        double des = (b*b) - (4*a*c);  
        if (des >= 0)  
        {  
            double r1 = (-b + Math.sqrt(des)) / (2*a);  
            double r2 = (-b - Math.sqrt(des)) / (2*a);  
            System.out.println("The roots of the equation are: " + r1 + ", " + r2);  
        }  
        else  
        {  
            System.out.println("There are no real solutions!");  
        }  
    }  
}
```

ALGORITHM:

Step 1 : Start

Step 2 : Read the value of coefficient of  $x^2$ ,  $x$  and constant.

Step 3 :  $dis = (b \times b) - (4 \times a \times c)$

Step 4 : If ( $dis > 0$ )

$$r_1 = (-b + \sqrt{dis}) / 2 \times a$$

$$r_2 = (-b - \sqrt{dis}) / 2 \times a$$

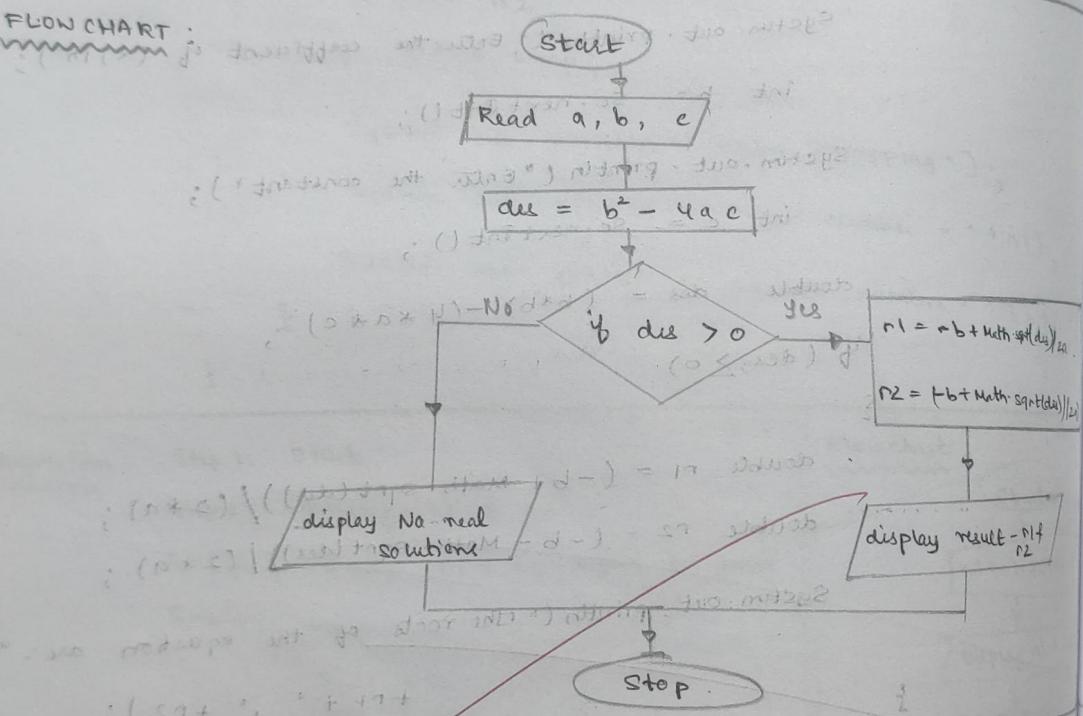
Step 5 : display the roots.

Step 6 : If ( $dis < 0$ )

display no real solutions

Step 7 : Stop.

FLOW CHART:



OUTPUT:

1 Enter the coefficient of  $(x^2)$

2 Enter the coefficient of  $(x)$

3 Enter the constant

There are no real solutions!

12 The roots of the equation are 4.1

```
Command Prompt  
E:\mana\3rd sem\oopj\Lab_prog>javac quadratic.java
```

```
E:\mana\3rd sem\oopj\Lab_prog>java Quadratic
```

```
NAME: MANASVINI DEEPAK
```

```
USN: 1BM22CS336
```

```
Enter the coefficient of (x^2)
```

```
1
```

```
Enter the coefficient (x)
```

```
1
```

```
Enter the constant
```

```
1
```

```
There are no real solutions!
```

```
E:\mana\3rd sem\oopj\Lab_prog>java Quadratic
```

```
NAME: MANASVINI DEEPAK
```

```
USN: 1BM22CS336
```

```
Enter the coefficient of (x^2)
```

```
2
```

```
Enter the coefficient (x)
```

```
-11
```

```
Enter the constant
```

```
12
```

```
The roots of the equation are: 4.0 , 1.5
```

```
E:\mana\3rd sem\oopj\Lab_prog>
```

PROGRAM 2: Develop a Java program to create a class Student with members USN, name, an array of credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

Algorithm:

- Step 1: Start, create a class Student
- Step 2: Declare member USN, name, array of credits and marks
- Step 3: Create a method input to accept student information
- Step 4: Read USN, name, credits and marks
- Step 5: Create a method display() to display student details.
- Step 6: Create a method to calculate SGPA: sgpa()
- Step 7: Check marks assign grade points.
- Step 8: 
$$\text{SGPA} = \frac{\sum_{i=0}^n \text{marks}_i * \text{credits}_i}{\sum_{i=0}^n \text{credits}_i}$$
- Step 9: Display SGPA
- Step 10: Create class run code
- Step 11: Create object of class Student.
- Step 12: Invoke input(), display() & sgpa() methods.
- Step 13: Stop.

Program:

```
class Student
```

```
{
```

```
    String USN = new String();  
    String name = new String();  
    int[] credits = new int[8];  
    int[] marks = new int[8];  
    void input()
```

```
{
```

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

```
    System.out.println("Enter the name of the student : ");
```

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

```
    System.out.println("Enter the USN : ");
```

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

```
System.out.println("Enter the credits in order");
for (int i = 0; i < 8; i++) {
    credits[i] = sc.nextInt();
}
System.out.println("Enter the marks of student in order:");
for (int i = 0; i < 8; i++) {
    marks[i] = sc.nextInt();
}
```

```
void display()
```

```
System.out.println("* * * * STUDENT DETAILS * * * *");
System.out.print("Name: " + name);
System.out.print("USN: " + USN);
System.out.print("The marks obtained by 2 credits for each subject are ");
for (int i = 0; i < 8; i++) {
    System.out.print(subject[i] + credits[i]);
    System.out.print(" " + subject[i] + " credits = " + credits[i]);
}
```

```
void sgpa()
```

```
double sum = 0, res = 0;
for (int i = 0; i < 8; i++)
```

```
double GP = 0;
if (90 < marks[i] && marks[i] <= 90) {
    GP = 10;
}
```

```
else if (20 < marks[i] && marks[i] <= 80) {
    GP = 9;
```

else if (marks[i] > 70 + marks[i] <= 80)

GP = 8;

else if (marks[i] > 60 + marks[i] <= 70)

GP = 7;

else if (marks[i] > 50 + marks[i] <= 60)

GP = 6;

else if (marks[i] > 40 + marks[i] <= 50)

GP = 5;

else

GP = 4;

res = res + GP \* credit[i];

sum = sum + credit[i];

} when n is taken for student info

double sgpa = res / sum;

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

}

}

class runcode

{

public static void main (String args [])

{

Scanner sc = new Scanner (System.in);

System.out.println("Enter the number of students");

int n = sc.nextInt();

for (int i = 1, j = 1; i <= n, j++) {

{

Student s = new Student();

s.input();

s.display();

s.sgpa();

}

}

Output : Enter the number of students

3

Enter the name of the student:  
Anjana

Enter the USN:

IBM 22CS 003

Enter the credits of the student in order:

4

4

3

3

3

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

</

subject 7 94

subject 7 media 1

subject 8 97

subject 8 media 1

SQPA 965

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

19/11/20

```
E:\mana\3rd sem\oop\Lab_prog>javac cgpa.java
E:\mana\3rd sem\oop\Lab_prog>java runcode
NAME: MANASVINI DEEPAK
USN: 1BM22CS336
Enter the number of students
1
Enter the name of the student:
Anjana
Enter the USN:
1BM22CS003
Enter the credits in order:
4
4
3
3
3
1
1
1
Enter the marks of student in order
91
91
88
90
82
85
94
97
****STUDENT DETAILS****
Name: Anjana
USN: 1BM22CS003
The marks obtained and credits for each subject are
Subject 1 91
Subject 1 +credits: 4
Subject 2 91
Subject 2 +credits: 4
Subject 3 88
Subject 3 +credits: 3
Subject 4 90
Subject 4 +credits: 3
Subject 5 82
Subject 5 +credits: 3
Subject 6 85
Subject 6 +credits: 1
Subject 7 94
Subject 7 +credits: 1
Subject 8 97
Subject 8 +credits: 1
SGPA: 9.5
```

PROGRAM 9: Create a class Book which contains four members name, author, twice, num of pages. Include a constructor to set the value of the members. Include methods to get the details of the object. Include a `toString()` method, which display the complete details of the book. Develop a Java program to create n book objects.

ALGORITHM:

Step 1: Create class Book.

Step 2: Declare the member variables.

Step 3: Write a constructor to assign values.

Step 4: Write methods:

void	get_Name (String Name);
void	set_Name (String Name);
void	get_Author (String Author);
void	set_Author (String Author);
void	get_Pages (int Pages);
void	set_Pages (int Pages);
void	get_Price (double Price);
void	set_Price (double Price);

Step 5: Create method `toString()` to return details in string format.

Step 6: Create another class NumberBooks.

Step 7: Read no. of books for which details are to be read.

Step 8: Declare array of Book type with n elements.

Step 9: Call the method set details.

Step 10: Invoke `toString()` to print details.

```
CODE :  
import java.util.*;  
  
class Book  
  
{  
    private String name;  
    private String author;  
    private double price;  
    private int numPages;  
  
    public Book()  
    {  
        this.name = "Unknown";  
        this.author = "Unknown";  
        this.price = 0.0;  
        this.numPages = 0;  
    }  
  
    public Book(String name, String author, int pages, double price)  
    {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.numPages = pages;  
    }  
  
    public void setName(String name)  
    {  
        this.name = name;  
    }  
  
    public void setAuthor(String author)  
    {  
        this.author = author;  
    }  
  
    public void setPage(int pages)  
    {  
        this.numPages = pages;  
    }  
  
    public void setPrice(double price)  
    {  
        this.price = price;  
    }  
  
    public String getName()  
    {  
        return name;  
    }  
  
    public void getAuthor()  
    {  
        return author;  
    }  
  
    public void getPages()  
    {  
        return numPages;  
    }  
}
```

public void getPrice()

{ return price; }

}

public String toString()

{

return "Book Name : " + name + " Author : " +  
author + " Price : " + price + " Pages : " + numPages;

}

}

public class NumberBooks

{

public static void main (String args[])

{

Scanner sc = new Scanner (System.in);

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

int n = sc.nextInt();

BOOK [] books = new Book [n];

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

{

books [i] = new Book();

System.out.print ("Enter book Name");

String name = sc.next();

System.out.print ("Enter book author");

String author = sc.next();

System.out.print ("Enter price");

double price = sc.nextDouble();

System.out.print ("Enter number of pages");

int numPages = sc.nextInt();

books [i].setName (name);

books [i].setAuthor (author);

books [i].setPrice (price);

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

books [i].setPages (numPages);

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

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

OUTPUT : Enter no. of Books : 2  
Enter Book Name : Harry Potter  
Enter Author : JK Rowling  
Enter Price : 250  
Enter number of Pages : 500  
Enter Book Name : Famous Five  
Enter author : Enid Blyton  
Enter Price : 200  
Enter number of Pages : 300  
Book Name : Harry Potter Author Name : JK Rowling Price: 250  
Pages: 500  
Book Name : Famous Five Author Name : Enid Blyton Price: 200  
Pages: 300

```
E:\mana\3rd sem\oop\Lab_prog>javac NumberBook.java

E:\mana\3rd sem\oop\Lab_prog>java NumberBook
NAME: MANASVINI DEEPAK
USN: 1BM22CS336
Enter number of books:
2
Enter Book Name:
Harry Potter
Enter Author:
JK Rowling
Enter Price:
250
Enter Number of pages:
600
Enter Book Name:
Famous Five
Enter Author:
Enid Blyton
Enter Price:
200
Enter Number of pages:
300
Book Name: Harry Potter Author: JK Rowling Price: 250.0 Pages: 600
Book Name: Famous Five Author: Enid Blyton Price: 200.0 Pages: 300
```

PROGRAM 10: Develop a Java Program to create an abstract class named Shape that contain two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle, such that one of the class extends the class Shape. Both the class contains only the method printArea() that prints the area of the given shape.

#### ALGORITHM :-

Step 1: Create an abstract class Shape.

Step 2: Declare members int a, int b

Step 3: Create abstract method printArea() with no body.

Step 4: Create class Rectangle that extends Shape.

Step 5: Declare members int a, int b.

Step 6: Create method which prints area

$$\text{area} = a * b;$$

Step 7: Create class Triangle. Declare the members and print area.

Step 8: Create class Circle extends Shape and implement printArea method.

Step 9: Create a public class Area.

Step 10: Create objects of Rectangle, Triangle and Circle class and invoke printArea().

CODE:

abstract class Shape

{  
private int a, b;  
shape () {  
}

abstract void printArea();

class Rectangle extends Shape

{  
private int a, b;

Rectangle (int a, int b)

{  
this.a = a;  
this.b = b;

void printArea()

{  
System.out.println ("Area of Rectangle: " + (a\*b));

class Triangle extends Shape

{  
private int a, b;

Triangle (int a, int b)

{  
this.a = a;  
this.b = b;

void printArea()

{  
System.out.println ("Area of triangle: " + (0.5 \* a \* b));

class Circle extends Shape

{  
private int a;

circle (int a)

{  
this.a = a;

{  
void printArea()

{  
System.out.println ("Area of circle: " + (3.14 \* a \* a));

public class Area

public static void main (String args [])

Shape r = new Rectangle(5,4);

shape t = new Triangle(4, 0);

shape c = new Circle(3);

r.printArea();

t.printArea();

c.printArea();

Output:

Area of Rectangle : 20

Area of triangle : 10.0

Area of Circle : 28.26

```
C:\Users\laptop>cd /d E:\mana\3rd sem\oopj\Lab_prog  
E:\mana\3rd sem\oopj\Lab_prog>javac Area.java  
E:\mana\3rd sem\oopj\Lab_prog>java Area  
NAME: MANASVINI DEEPAK  
USN: 1BM22CS336  
Area of Rectangle: 20  
Area of Triangle: 12.0  
Area of Circle: 28.25999999999998
```

PROGRAM 10:- Develop a Java Program to create a class Bank that maintains two kinds of account for its customers. one called Saving 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 holder should maintain a minimum balance and if the balance falls below the level, a service charge is imposed. Create a class Account that stores account details and achieve the following: accept deposit & update display balance, compute and deposit interest withdraw and update balance.

#### ALGORITHM:

Step 1: Write a class Account

Step 2: Initialize variables ~~customer-name, Accno, acctype, balance~~

Step 3: Input : Enter customer name, accno, balance account type as savings or current from user.

Step 4: Enter choice of savings or current account.

Step 5: Read the choice if choice is savings Step 6 for current step 4.

Step 6: Enter initial balance, withdrawal amount & min balance from user check condition for min

balance ie initial - balance >= withdrawal should be  $\geq \text{min}$  ie  $\geq 800$  then new Balance = balance - withdrawal. along the message

Then print current balance, enter interest rate & time to calculate CI.

$$CI = \text{balance} * \text{power}(1 + \text{interest rate}, \text{time}) - \text{initial\_balance}$$

point of deposit with interest.

Step 4: else if (choice == 2)

create object of Saving Account

initial balance, withdrawal amount. Initialize

Step 8: stop further process

CODE:

```
import java.util.*;  
  
class Account {  
    String customerName;  
    long accno;  
    String accountType;  
    double balance;  
  
    public Account (String customerName, long accno, String accountType) {  
        this.customerName = customerName;  
        this.accno = accno;  
        this.accountType = accountType;  
        this.balance = 0.0;  
    }  
  
    public void displayBalance () {  
        System.out.println ("Account Number : " + accno);  
        System.out.println ("Customer Name : " + customerName);  
        System.out.println ("Account Type : " + accountType);  
        System.out.println ("Balance $ " + balance);  
    }  
}
```

class CurrentAccount extends Account

{ double minBalance;

double serviceCharge;

public CurrentAccount (String customerName, long accNo)

{ super (customerName, accNo, "Current");

this.minBalance = 500.0;

this.serviceCharge = 50.0;

}

public void withdraw (double amount)

{ if (balance - amount >= minBalance)

{

balance -= amount;

}

System.out.println ("Withdrawal successful.

current Balance : \$" + balance);

}

{

else

{

System.out.println ("Insufficient funds !

Withdrawal not allowed!");

public void imposeServiceCharge ()

{ if (balance < minBalance)

{ balance -= serviceCharge;

System.out.println ("Service charge imposed.

current Balance : \$" + balance);

{

else System.out.println ("Insufficient funds !

Service charge imposed.");

class SavAcct extends Account

double interestRate;

public SavAcct (String customerName, long accno)

{ super (customerName, accno, "Savings");

this . interestRate = 0.05;

public void compoundInterest (double initialAmount, int term)

double compoundInterest = initialAmount \*

Math . pow (1 + interestRate,

(double term = < term - term) / initial amount;

Balance += compoundInterest;

System . out . println ("compoundInterest deposited -

Current Balance after \$ " + balance);

public class Bank

public static void main (String args [] )

Scanner sc = new Scanner (System . in );

System . out . println ("choose Account Type : ");

System . out . println (> "1 current /n 2 : savings ");

int choice = sc . nextInt ();

System . out . println ("Enter customer Name ");

String customerName = sc . nextLine();

System . out . println ("Enter account number ");

long accno = sc . nextLong ();

if choice == 1)

1) Create new current account

curr Account currAccount = new currAccount (customerName, accNo);

System.out.println ("Enter initial balance : \$");

double initialBalance = sc.nextDouble();

curr Account balance = initialBalance;

System.out.print ("Enter withdraw amount : \$");

double withdrawalAmount = sc.nextDouble();

curr Account . withdraw (withdrawalAmount);

curr Account . imposeServiceCharge ();

curr Account . displayBalance ();

else if choice == 2)

Sav Account savAccount = new SavAccount (customerName,

accNo);

System.out.println ("Enter initial balance \$");

double initialBalance = sc.nextDouble();

Sav Account . balance = initialBalance;

System.out.print ("Enter the withdrawal

amount \$");

double withdrawalAmount = sc.nextDouble();

SavAccount . balance = withdrawalAmount;

System.out.println ("Withdrawal successful in

current Balance : \$" + savAccount . Balance);

System.out.print ("Enter interest rate : ");

double interestRate = sc.nextDouble();

Sav Account . interestRate = interestRate;

SAV Account - display Balance () ;

System.out.println("Enter term (in years) for CI calculation");  
int term = sc.nextInt();

SAV Account - compound interest (initial Balance, term);

SAV Account - display (Balance());

else

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

}

Output:

Choose Account Type:

1) Current

2) Savings.

Enter customer Name: Aditi

Enter Account Number: 0029789579.

Enter initial balance: 10000

Enter withdrawal amount: 5500

Withdrawal successful. Current Balance: 4500

Enter interest rate 0.05

Account Number: 0029789579

Customer Name: Aditi

Account Type: Savings

Balance: 4500

Enter term in years for CI calculation: 2

Balance: 5012.5

```
E:\mana\3rd sem\oop\Lab_prog>javac Bank.java
E:\mana\3rd sem\oop\Lab_prog>java Bank
NAME: MANASVINI DEEPAK
USN: 1BM22CS336
ENTER CHOICE
ENTER 1 TO CREATE SAVINGS ACCOUNT
ENTER 2 TO CREATE CURRENT ACCOUNT
1
Enter Customer Name:
Aditi
Enter Account Number:
00297789597
Enter Balance:
10000
Enter Interest:
0.05
ENTER CHOICE FOR SAVINGS ACC
ENTER 1 TO DEPOSIT AMT
ENTER 2 TO COMPUTE INTEREST
ENTER 3 TO WITHDRAW
ENTER 4 TO EXIT
1
Enter amount to deposit:
2000
Deposit successful. Updated balance: 12000.0
Current balance: 12000.0
ENTER CHOICE
1
Enter amount to deposit:
200
Deposit successful. Updated balance: 12200.0
Current balance: 12200.0
ENTER CHOICE
2
Interest computed and deposited. Updated balance: 12206.1
ENTER CHOICE
3
Enter amount to withdraw:
350
Withdrawal successful. Updated balance: 11856.1
Current balance: 11856.1
ENTER CHOICE
4
```

PROGRAM 12: Create a package CIE which has two classes - Student and Internals. The class Personel 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 5 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:

Step 1: Start

Step 2: Create a package CIE

Step 3: Define a public class Student with members USN, name, sem

Step 4: A method input() to accept value for member variables

Step 5: Create another class Internals in package CIE.

Has an array storing marks in 5 subjects

Step 6: Create package SEE.

Step 7: import Student class from package CIE.

Step 8: Create an External class which extends Student.

Step 9: Consist of array storing SEE marks in 5 subjects

Step 10: Create a new class and import packages CIE & SEE  
Create array storing total marks.

Step 11: Read input for number of students

Step 12: Instantiate objects External & Internal

Step 13: For each course take input of external + internal marks

Step 14: Display total marks. Step 15: Stop.

CODE:-

```
package CIE;
import java.util.*;
public class Student
{
    public String USN;
    public String name;
    public int sem;
    public void input()
    {
        Scanner sc = new Scanner (System.in);
        System.out.println("Enter USN, name, sem.");
        USN = sc.nextLine();
        name = sc.nextLine();
        sem = sc.nextInt();
    }
}
```

Package CIE;

Public class Internals

```
{}
Public int internals[] = new int [5];
```

Package SEE.

```
import CIE.Student;
```

Public class Extends extends Student

```
{}
public int see[] = new int [5];
```

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

```
import SEE.*;
```

```
import CIE.*;
```

Public class Marks

```
{}
public static void main (String args[])
{
    int m = new int [5];
    Scanner sc = new Scanner (System.in);
}
```

```
int m = new int [5];
```

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

```
System.out.println("Enter the number of students: ");
```

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

SEE External  $se[] = \text{new SEE.External}[n];$

Internal  $in[] = \text{new CIE}[n];$

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

```
{
```

```
    se[i] = new SEE.External();
```

```
    in[i] = new CIE();
```

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

```
se[i].input();
```

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

```
{
```

```
    System.out.println("Enter internal & External marks  
of " + (j+1) + " subject");
```

```
    se[i].internal[j] = sc.nextInt();
```

```
    in[i].see[j] = sc.nextInt();
```

```
    m[j] = se[i].internal[j] + in[i].see[j];
```

```
{
```

```
System.out.println("Final marks of " + se[i].name);
```

```
for (int k=0; k<5; k++)
```

```
{
```

```
    System.out.println("Course: " + (k+1) +  
    "In Marks: " + m[k]);
```

```
{
```

OUTPUT: Enter the number of students

1

Enter Details

Manaswini

18M2205336

3

Enter internal & external marks of 1

45

45

46

46

```
E:\mana\3rd sem\oop\Lab_prog>javac Marks.java

E:\mana\3rd sem\oop\Lab_prog>java Marks
NAME: MANASVINI DEEPAK
USN: 1BM22CS336
Enter Student name:
Disha
Enter USN:
1BM22CS286
Enter Semester:
3
Enter SEE Marks:
Enter marks:
90
Enter marks:
78
Enter marks:
84
Enter marks:
85
Enter marks:
93
Enter CIE Marks:
Enter marks:
45
Enter marks:
43
Enter marks:
49
Enter marks:
50
Enter marks:
41
Name: Disha
USN: 1BM22CS286
Semester: 3
SEE Marks: [90, 78, 84, 85, 93]
CIE Marks: [45, 43, 49, 50, 41]
```

PROGRAM 13: 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 takes both father and son's age and throws an exception if son's age  $\geq$  father's age.

ALGORITHM:

- STEP 1: Create an exception WrongAge which extends exception
- STEP 2: Create base class Father.
- STEP 3: Make a constructor Father(age) which inputs age.
- STEP 4: If age  $< 0$ ; throw exception Wrong Age.
- STEP 5: Create Son class derived from Father.
- STEP 6: Make both father and son's age in constructor.
- STEP 7: If son's age  $\geq$  father's age, throw exception Wrong Age.

OUTPUT: Father and Son's Age

Enter Father's Age

12

Enter Son's Age

24

Son's age greater !!

code: import java.util.\*;

class WrongAge extends Exception

public WrongAge (String s)

super(s);

class Father

int age;

Father (int age) throws WrongAge

if (age < 0)

throw new

WrongAge ("Age is less than zero!");

this.age = age;

public int getAge()

return age;

class Son extends Father

int s-age, f-age;

Son (f-age, s-age) throws WrongAge

super(f-age);

if (s-age > f-age)

throw new WrongAge ("Son's age greater!");

this.s-age = s-age;

PR  
public int getSonAge()

{  
return (s-age);

}

}

(e. print) of Age class

public class Age

{

public static void main (String args [])

{  
System.out.println ("Father and Son's Age");

Scanner sc = new Scanner (System.in);

int f-age, s-age;

System.out.println ("Enter father's age");  
f-age = sc.nextInt();

try

{  
Father f = new Father (f-age);  
System.out.println ("Father's age : " + f.getAge());

catch (WrongAge e)  
{  
System.out.println (e.getMessage());

try

{

System.out.println ("Enter son's age");  
s-age = sc.nextInt();

Son s = new Son (f-age, s-age);

System.out.println ("Son's Age : " + s.getSonAge());

catch (WrongAge e)

{  
System.out.println (e.getMessage());

```
E:\mana\3rd sem\oop\Lab_prog>javac Age.java

E:\mana\3rd sem\oop\Lab_prog>java Age
NAME: MANASVINI DEEPAK
USN: 1BM22CS336
Father Son
Enter father's age
12
Father's age: 12
Enter son's age
24
ERROR:Son's age is greater than father's age

E:\mana\3rd sem\oop\Lab_prog>java Age
NAME: MANASVINI DEEPAK
USN: 1BM22CS336
Father Son
Enter father's age
-12
ERROR:Age less than zero!!
Enter son's age
12
ERROR:Age less than zero!!

E:\mana\3rd sem\oop\Lab_prog>java Age
NAME: MANASVINI DEEPAK
USN: 1BM22CS336
Father Son
Enter father's age
45
Father's age: 45
Enter son's age
23
Age of Son: 23
```

PROGRAM 14: Write a program creates 2 threads 1, & one thread displaying "BMS college of Engineering" once every ten seconds & another displaying "CSE" once every 2 seconds.

ALGORITHM :-

Step 1: Start.

Step 2: Create class Thread1 as extends thread.

Step 3: Initialize  $t_1 = 0$ ; time = 30000 in constructor of Thread1.

Step 4: Create public void run() method.

Step 5: Use for while loop holding true time  $t_1 < 30000$   
Print "BMS college of Engineering".

Step 6: Include a try block and use sleep method for 10000 ms  
In catch block print statement error.  
& increment  $t_1 = t_1 + 10000$ ;

Step 7: Create class Thread 2 that extends thread.

Initialize  $t_2 = 0$ ; time = 30000.

Step 8: In the run(); use while loop to print CSE

Step 9: In a try block use sleep method for 2000 s.  
In the catch block, print error statement &  
and increment  $t_2 = t_2 + 2000$ .

Step 10: Create a class Demo.

Instantiate object of Thread1 & Thread2.

Call start method of both objects.

Stop.

CODE: class Thread1 extends Thread

```
{ int t1, time;
    thread1()
    {
        t1 = 0;
        time = 30000;
    }

    public void run()
    {
        while (t1 <= time)
        {
            System.out.println("BMS COLLEGE OF ENGINEERING");
            try
            {
                sleep(10000);
            }
            catch (InterruptedException e)
            {
                System.out.println("Error");
            }
        }
    }
}
```

class Thread2 extends Thread

```
{ int t2, time;
    thread2()
    {
        t2 = 0;
        time = 30000;
    }

    public void run()
    {
        while (t2 <= time)
        {
            System.out.println("CSE");
        }
    }
}
```

try

{

sleep(2000);

}

catch (InterruptedException e)

System.out.println("Error");

}

t2 = 2000;

at take no error record should come

3.

above code is part of function with right logic with +

class Demo

```
public static void main (String args[])
{
    thread1 = new thread();
    thread1.start();
    thread2 = new thread();
    thread2.start();
}
```

at start() method for thread 1 and 2

th1.start();  
th2.start();

}

now we have to start 2 threads at same time

OUTPUT:

BMS COLLEGE OF ENGINEERING

CSE

CSE

CSE

BMS COLLEGE OF ENGINEERING

CSE

```
E:\mana\3rd sem\oop\Lab_prog>javac threadexample.java

E:\mana\3rd sem\oop\Lab_prog>java Threads
NAME: MANASVINI DEEPAK
USN: 1BM22CS336
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
CSE
CSE
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
CSE
CSE
BMS COLLEGE OF ENGINEERING
```

PROGRAM 15:

PF

\* Definitions :-

↳ JFrame class :- In swing functionality, the container used for application is called a JFrame.

- It acts like the main window where components like labels, buttons, textfield are added to create a GUI.

\* jfrm.set size :- Resize this component so that it has width w and height h.

\* jfrm.setLayout (LayoutManager) :- we use to set layout of the container.

"Flow layout":

- used to arrange the components in a line,
- default layout of applet or panel.

\* jfrm.setDefaultCloseOperation : used to specify one of the several options for the close button.

\* JLabel : creates an empty label without any text.

\* JTextField :

- constructor that creates a new empty textField with the given string and a specified number of columns.

\* add :

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 the message dialog box.

```
import javax.swing.*; // provides toolkit class = J window toolkit
import java.awt.*; // provides container toolkit class = Box
import java.awt.event.*;
class SwingDemo
{
    SwingDemo()
    {
        JFrame jfrm = new JFrame("Divide APP");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel jlab = new JLabel("Enter the dividend and divisor:");
        JTextField qjtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        JButton button = new JButton("calculate");
        JLabel res = new JLabel();
        JLabel ares = new JLabel();
        JLabel blab = new JLabel();
        JLabel arstab = new JLabel();
        jfrm.add(jlab);
        jfrm.add(qjtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(res);
        jfrm.add(ares);
        jfrm.add(blab);
        jfrm.add(arstab);
    }
}
```

```
jfrm.add (ur);
```

```
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 wt)
```

```
{
```

```
System.out.println ("Action event from a textfield")
```

```
}
```

```
};
```

```
ajtf.addActionListener (l);
```

```
bjtf.addActionListener (l);
```

```
button.addActionListener (new ActionListener ())
```

```
{
```

```
public void actionPerformed (ActionEvent wt)
```

```
try
```

```
{
```

```
int a = Integer.parseInt (ajtf.getText());
```

```
int b = Integer.parseInt (bjtf.getText());
```

```
int ans = a + b;
```

```
alab.setText ("In A = " + a);
```

```
blab.setText ("In B = " + b);
```

```
anslab.setText ("In Ans = " + ans);
```

```
}
```

```
catch (NumberFormatException e)
```

```
{
```

```
alab.setText ("");
```

```
blab.setText ("");
```

since only integers (+),

```

    catch (ArithmaticException e)
    {
        alab.setText(" ");
        blab.setText(" ");
        onelab.setText(" ");
        er.setText("B should be NON zero+");
    }
}

ifrm.setVisible(true);

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

```

OUTPUT:

Divide APP.

Enter the dividend and divisor:	
<input type="text" value="15"/>	<input type="text" value="5"/>
<input type="button" value="calculate"/>	
Dividend (A) = 15	
Divisor (B) = 5      Result = 3	

8/2/20  
28/2/20

```
E:\mana\3rd sem\oop\Lab_prog>javac SwingDemo.java
```

```
E:\mana\3rd sem\oop\Lab_prog>java SwingDemo
```

```
E:\mana\3rd sem\oop\Lab_prog>javac SwingDemo.java
```

```
E:\mana\3rd sem\oop\Lab_prog>java SwingDemo
```

```
NAME: MANASVINI DEEPAK
```

```
USN: 1BM22CS336
```

