

Lab Program-L (Quadratic eqⁿ)

```
import java.util.Scanner;
public class quadratic
{
    public static void main (String [] args)
    {
        float a,b,c,r1,r2,d;
        Scanner sc = new Scanner (System.in);
        System.out.println ("enter coefficient of n2");
        a = sc.nextFloat ();
        System.out.println ("Enter coeff. of n:");
        b = sc.nextFloat ();
        System.out.println ("Enter constant:");
        c = sc.nextFloat ();
        if (a == 0)
        {
            System.out.println ("Invalid Input");
        }
        else
        {
            float det = (float) Math.pow (b, 2) - 4 * a * c;
            if (det > 0)
            {
                r1 = (float) (-b + Math.sqrt (det)) / (2 * d);
                r2 = (float) (-b - Math.sqrt (det)) / (2 * d);
                System.out.println ("The roots are:");
                System.out.println (" " + r1 + " and " + r2);
            }
            else if (det == 0)
            {
                r1 = (float) -b / (2 * a);
                System.out.println ("root is = " + r1);
            }
        }
    }
}
```

```
    }  
else {  
    System.out.println ("No real sol");  
}  
}  
}
```

Output:

Enter coeff of x^2 :

9

Enter coeff of x :

6

Enter constant:

2

The root are 0, -0.5 and -1.0

Enter coeff of x^2 :

2

Enter coeff of x :

1

Enter constant:

1

No real sol

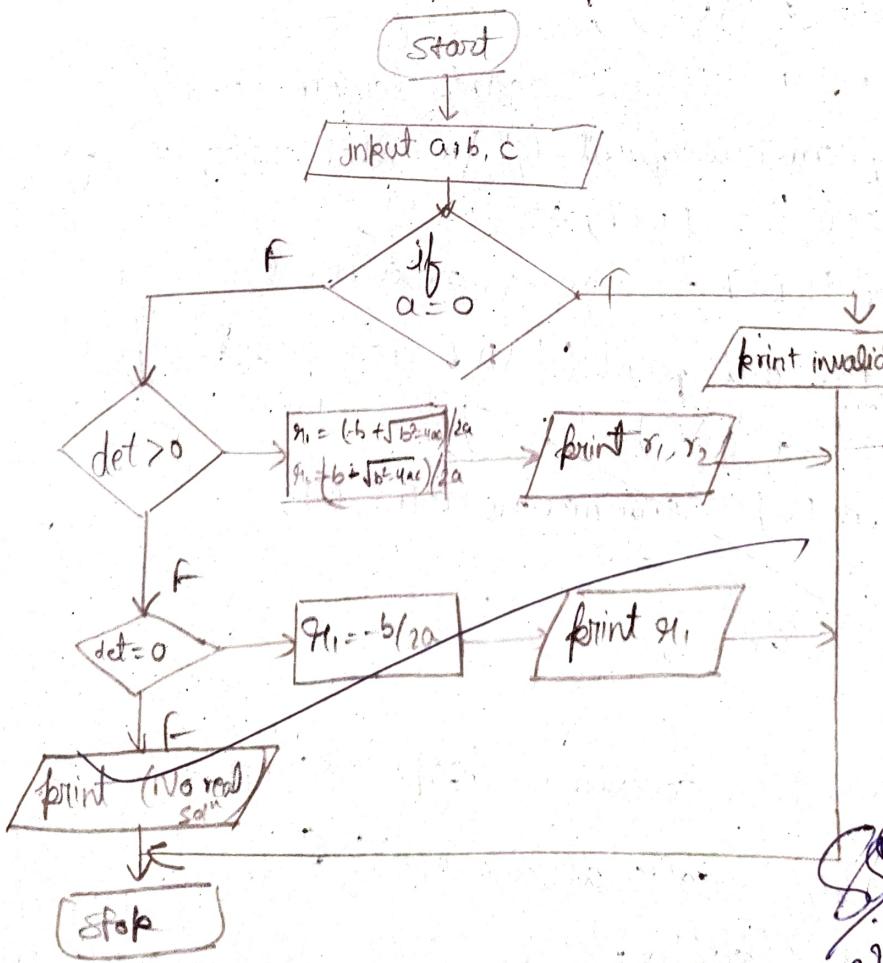
Algorithm:

- Step 1: Start
- Step 2: input a, b, c
- Step 3: If $a=0$; print "invalid"
- Step 4: else $\det = b^2 - 4ac$
- Step 5: if $\det > 0$
 $r_1 = \frac{(-b + \sqrt{b^2 - 4ac})}{2a}$
 $r_2 = \frac{(-b - \sqrt{b^2 - 4ac})}{2a}$
- Step 6: else if $\det = 0$
 $r_1 = -\frac{b}{2a}$

Step 7: else: no real sol

Step 8: Stop

flowchart:



22/12/23

Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\tripa>cd onedrive/desktop/java

C:\Users\tripa\OneDrive\Desktop\java>javac progl.java

C:\Users\tripa\OneDrive\Desktop\java>java progl
enter the coefficients

1

-5

6

r1=3.0

r2=2.0

Lab Program - 2

```
import java.util.Scanner;  
class Student {  
    String USN;  
    String name;  
    int[] credits = new int[8];  
    int[] marks = new int[8];  
    // Method to accept student details  
    public void acceptDetails() {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter USN: ");  
        USN = scanner.nextLine();  
        System.out.print("Enter Name: ");  
        name = scanner.nextLine();  
        System.out.println("Enter details for each  
        subject : \n");  
        for (int i = 0; i < credits.length; i++) {  
            System.out.print("\nEnter credits for  
            subject " + (i + 1) + ": ");  
            credits[i] = scanner.nextInt();  
            System.out.print("\nEnter marks for  
            subject " + (i + 1) + ": ");  
            marks[i] = scanner.nextInt();  
        }  
        scanner.close();  
    }  
}
```

// Method to calculate SGPA

```
public double calculateSGPA() {
```

```
    int totalCredits = 0;
```

```
    int weightedSum = 0;
```

```
    double ans;
```

```
for (int i=0; i< credits.length; i++) {  
    totalCredits += credits[i];  
    int gradePoints;  
    gradePoints = (marks[i]/10) + 1;  
    if (gradePoints == 11) {  
        gradePoints = 10;  
    }  
    else if (gradePoints <= 4) {  
        gradePoints = 4;  
    }  
    weightedSum += gradePoints * credits[i];  
}  
ans = (double) weightedSum / (double) totalCredits;  
return ans;  
}
```

```
public class SGPA {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        // Create a student object  
        Student student = new Student();  
        // Accept and display details  
        student.acceptDetails();  
        System.out.println("In Student Details :");  
        System.out.println("USN : " + student.usn);  
        System.out.println("Name : " + student.name);  
  
        // calculate and display SGPA  
        double sgpa = student.calculateSGPA();  
        System.out.println("In SGPA : " + sgpa);  
        scanner.close();  
    }  
}
```

Output

Enter your details below to calculate your SGPA:

Enter your USN: 1BM22CS049

Enter your name: Arijady Trifathi

Enter your marks: 90 90 89 83 85 95 96 92

9.

Name: Arijady Trifathi

USN: 1BM22CS049

SGPA: 9.25

Algorithm

Step 1: start

Step 2: Initialize the variables USN, marks, name, sgpa

Step 3: Calling class of student in which calling function first();

Step 4: Input: "Enter USN" + USN

Step 5: Input: "Enter name" + name

Step 6: print "Enter marks"

Step 7: for (i=0; i<8; i++) {

(arr[i] = 100 in.nextInt();)

Step 8: Read array credits as { 4,4,3,3,3,1,1,1 }

Step 9: for (i=0; i<arr.length; i++) {

if (arr[i] >= 100) {

arr[i] = arr[i] - 10;

}

elseif (arr[i] < 40) {

arr[i] = 0;

marks += credit[i] * (mark[i]/10 + 1)

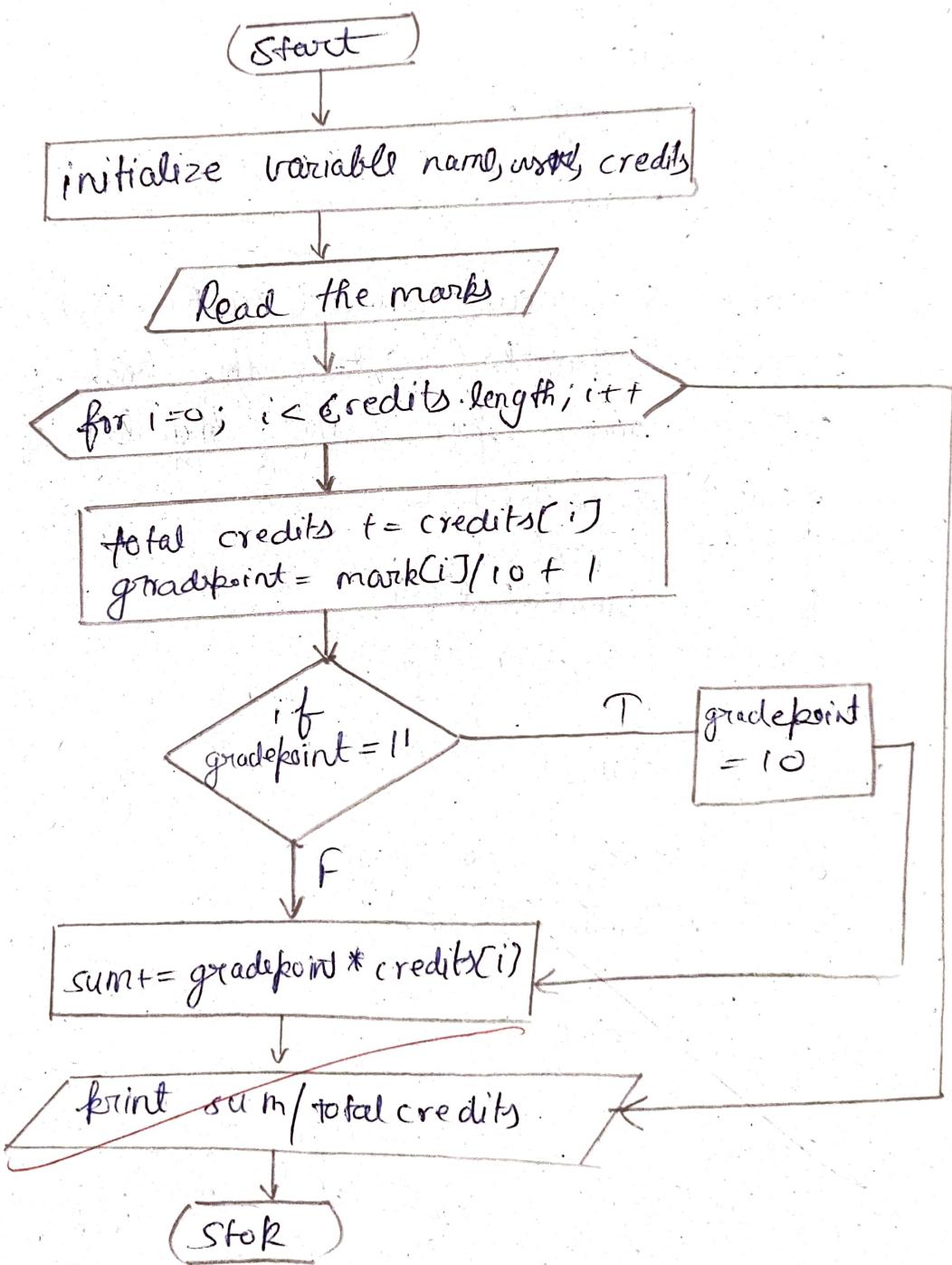
}

Step 10 : Sgpa = marks / 100

Step 11 : print "sgpa = " + sgpa.

Step 12 : stop

flowchart



```
C:\Users\tripa\OneDrive\Desktop\java>javac Prog2.java
```

```
C:\Users\tripa\OneDrive\Desktop\java>java Prog2
```

Enter your name:

aradhy

Enter your usn:

1BM22CS049

Enter the number of subject:

3

Enter the marks of subject 1 :

95

Enter credits of subject 1 :

4

Enter the marks of subject 2 :

96

Enter credits of subject 2 :

4

Enter the marks of subject 3 :

92

Enter credits of subject 3 :

3

Name: aradhy USN: 1BM22CS049

The marks of a subject 1 : 95

The credits of the subject : 4

The marks of a subject 2 : 96

The credits of the subject : 4

The marks of a subject 3 : 92

The credits of the subject : 3

The SGPA of USN: 1BM22CS049 Name: aradhy is : 10.0

Program-3
Create a class Book which contains four members
name, author, price, num-pages. Include a
constructor.

```
import java.util.Scanner;  
class book {  
    String name;  
    String author;  
    float price;  
    int num-page;  
  
    void set-details()  
{  
    Scanner s = new Scanner(System.in);  
    System.out.println("Enter the book name,  
author name, price and number of pages");  
    name = s.next();  
    price = s.next();  
    price = s.nextFloat();  
    num-pages = s.nextInt();  
}  
  
void get-details()  
{  
    String details = toString();  
    System.out.println(details);  
}  
  
String toString()  
{  
    return "The book " + name + " is written by "  
    + author + " consists of " + num-pages + " pages"  
    and cost around " + price;  
}
```

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

```
{  
    Scanner S = new Scanner (System.in);  
    System.out.println ("Enter the number of books");  
    int n = S.nextInt ();  
    book b [] = new book [n];  
    for (int i=0; i<n; i++)  
    {  
        b [i] = new book ();  
        b [i].set = details ();  
    }  
    for (int i=0; i<n; i++)  
    {  
        b [:].getDetails ();  
    }  
}
```

Output:

```
Enter the number of books
```

2

Enter the book name, author name, price and
number of pages

ABC

markhaward

500

320

Enter the book name, author name, price and number of pages

Xyz

David page

755

250

Output:

The book ABC was written by mark howard
consists of 500 pages and costs around 320
The book XYZ was written by David psyc
consists of 755 pages and cost around 250

Algorithm:

Step1: Start

Step2: Initialize the variables name, author, price,
num-pages

Step3: Calling class book.

Step4: Input: "Enter bookname, author, pages, prices"

Step5: Read the data

Step6: Input: "Enter the no of books you
want to generate";

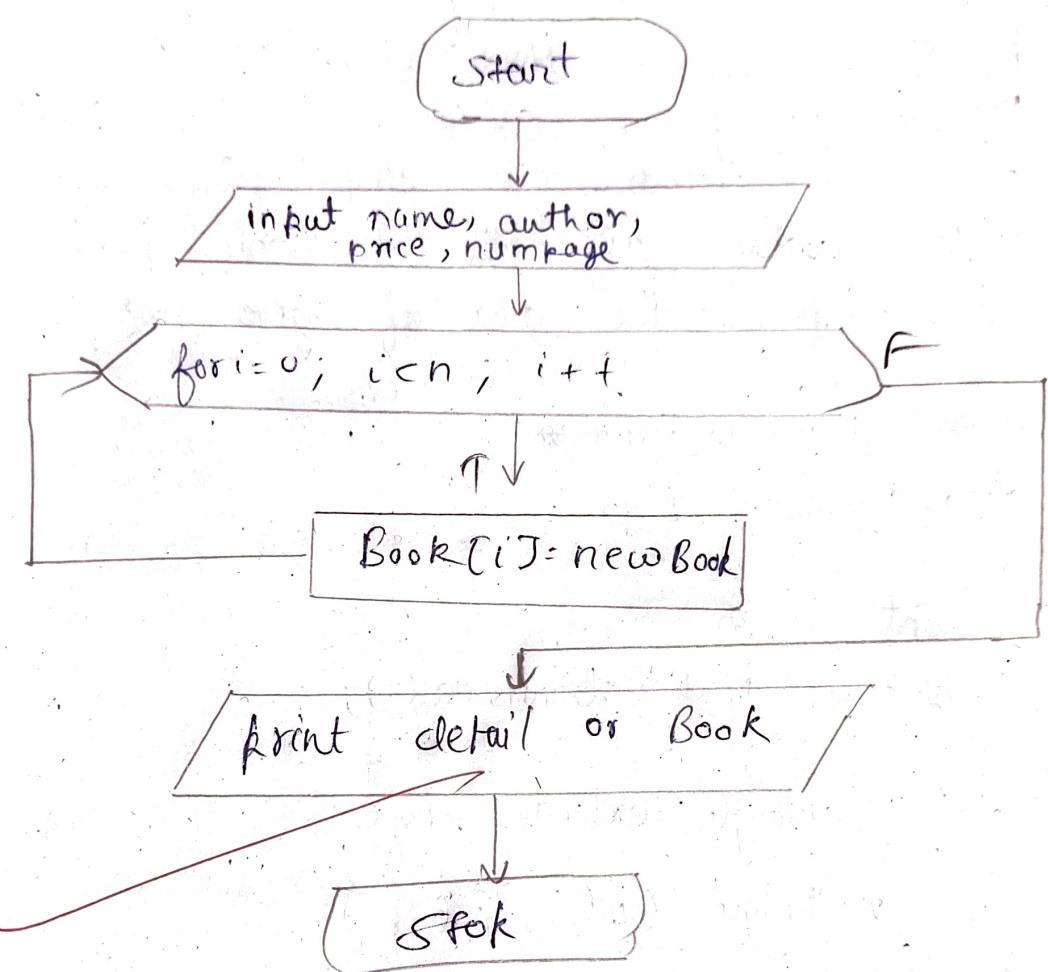
Step7: for (int i=0; i<n; i++) {
 b[i] = new book();
 b[i].set_details();
}

Step8: Input: "Book details"

Step9: for (int i=0; i<n; i++)
{
 b[i].get_details();
}

Step10: Stop

flowchart



Enter number of books:

2

Enter the name of book:

A

Enter the author of book:

B

Enter the price of the book:

254

Enter the number of pages:

568

Book: A Author: B details entered.

Enter the name of book:

C

Enter the author of book:

D

Enter the price of the book:

456

Enter the number of pages:

590

Book: C Author: D details entered.

Book Name: A Author Name: B Price: 254.0 Pages: 568

Book Name: C Author Name: D Price: 456.0 Pages: 590

Prog 4.

- Q. 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 & Circle such that each of these extends class shape. Each of these contain only the method. printArea() that prints the area of given shape.

```
import java.util.Scanner;
abstract class shape
{
    int a, b;
    abstract void printArea();
}
class rectangle extends shape
{
    rectangle (int l, int b)
    {
        a = l;
        this.b = b;
    }
    void printArea()
    {
        System.out.println ("Area of rectangle : " + a * b);
    }
}
```

```

class triangle extends shape
{
    triangle (int ba, int h)
    {
        a = ba;
        b = h;
    }
    void printArea()
    {
        System.out.println (" Area of triangle "
            + " " + 0.5*a*b);
    }
}

class circle extends shape
{
    circle (int r)
    {
        area = pi * r * r;
    }
    void printArea()
    {
        System.out.println (" Area of circle " + 3.14*r*r);
    }
}

class main
{
    public static void main (String [] args)
    {
        Scanner s = new Scanner (System.in);
        System.out.println (" enter the length "
            " and breadth of rectangle ");
        rectangle rec = new rectangle
            (s.nextInt (), s.nextInt ());
        rec. h
    }
}

```

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

```
{
```

```
    Shape ob = Rectangle (2,4);
```

```
    ob. printarea();
```

```
    Shape ob1 = New Rect Triangle (4,4);
```

```
    ob1. printarea();
```

```
}
```

```
}
```

Algorithm :

Step 1: Start

Step 2: Declare abstract class shape.

Step 3: Initialize objects circle, rectangle, triangle

Step 4: Extend class shape to circle use
method area to print area

Step 5: Extend class shape to rectangle, use
method area to print area.

Step 6: Extend class to triangle, use method
area to print area

Step 7: Stop.

Output

~~Area = 8~~

~~Area = 8~~

The area of rectangle is: 18

The area of triangle is: 12.0

The area of circle is: 78.5

Lab 6

Q2. import java.util.Scanner;

```
class account {  
    String customerName;  
    long accNo;  
    String accountType;  
    double balance;  
    public account (String customerName, long accNo, String accountType) {  
        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);  
    }  
}
```

3
~~class curAcct extends Account~~
~~class cur {~~

~~double minBalance;~~
~~double serviceCharge;~~
~~public curAcct (String customerName,~~
~~long accNo) {~~
~~super (customerName, accNo, "current");~~
~~this.minBalance = 500.0;~~
~~set minimum balance () {~~

```
set service charge ()  
{ this.service charge = 50;  
}  
  
public void withdraw (double amount)  
{  
    if (balance - amount >= minBalance) {  
        balance -= amount;  
        System.out.println ("Withdrawal successfully  
        current balance: $ " + balance);  
    } else {  
        System.out.println ("insufficient funds  
        Withdrawals not allowed");  
    }  
}  
  
public void impose Service Charge () {  
    if (balance < minBalance) {  
        balance -= service charge;  
        System.out.println ("Service charge imposed  
        Current Balance: Rs " + balance);  
    }  
}  
  
class SavAcct extends Account {  
    double interestRate;  
    public SavAcct (String customerName, long accno)  
    {  
        super (customerName, accno, "Savings");  
        this.interestRate = 0.05;  
    }  
}
```

```
public void depositInterest() {
    double interest = balance * interestRate;
    balance += interest;
    System.out.println("Interest deposited.");
    Current Balance: $ " + balance);
}

public void compoundInterest(double initialAmount, int term) {
    double compoundInterest = initialAmount *
        Math.pow(1 + interestRate, term) - initialAmount;
    balance += compoundInterest;
    System.out.println("Compound Interest deposited");
    Current Balance: Rs " + balance);
}

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");
        System.out.println ("2. Savings");
        System.out.println ("Enter choice (1 or 2): ");
        int choice = scanner.nextInt();
        System.out.println ("Enter account number:");
        long accno = scanner.nextLong();
        if (choice == 1) {
            Current currAccount = new Current (CustomerName,
                accno);
            System.out.println ("Enter initial Balance: $ ");
        }
    }
}
```

```
double initialBalance = scanner.nextDouble();
curAccount.Balance = initialBalance;
System.out.println ("Enter withdrawal amount:$");
double withdrawalAmount = scanner.nextDouble();
curAccount.withdrawal (withdrawalAmount);
curAccount.imposeServicecharge ();
curAccount.displayBalance ();
} else if (choice == 2) {
    SavAcct savAccount = new SavAcct (customerName, accno);
    System.out.println ("Enter initial Balance: $");
    double initialBalance = scanner.nextDouble();
    savAccount.Balance = initialBalance;
    System.out.println ("Enter withdrawal Amount:$");
    savAccount.Balance -= withdrawalAmount;
    System.out.println ("Enter interest rate:");
    double interestRate = scanner.nextDouble();
    savAccount.interestRate = interestRate;
    savAccount.displayBalance ();
    System.out.println ("Enter term for compound
    interest calculation:");
    int term = scanner.nextInt();
    savAccount.compoundInterest (initialBalance, term);
    savAccount.displayBalance ();
} else {
    System.out.println ("Invalid choice");
}
}
SSD
19/11/24
```

```
Enter Name:  
Mohan  
Enter Account Number:  
6796352  
1.Savings 2.Current  
Enter Account Type:  
1  
Enter deposit  
3000  
Name: Mohan  
Account Type: Savings  
Account Number: 6796352  
Current Balance: 3000.0  
1.Deposit 2.Withdraw 3.Interest 4.Exit  
Enter your choice:  
1  
Enter amount:  
1000  
Balance: 4000.0  
Enter your choice:  
3  
Enter time period:  
3  
Enter rate:  
7  
Interest: 900.172000000005  
Interest Deposited Amount : 4900.172000000005  
Enter your choice:  
4  
Enter 1 to continue or 0 to exit  
1  
Enter Name:  
Rahul  
Enter Account Number:  
6457786  
1.Savings 2.Current  
Enter Account Type:  
2  
Enter deposit(>2000)  
4500  
Name: Rahul  
Account Type: Current  
Account Number: 6457786  
Current Balance: 4500.0  
1.Deposit 2.Withdraw 3.Exit  
Enter your choice:  
2  
Enter amount:  
2400  
Balance: 2100.0  
Enter your choice:  
2  
Enter amount:  
2400  
Enter insufficient funds.  
Balance: 2100.0  
Enter your choice:  
3  
Invalid Choice  
Enter 1 to continue or 0 to exit  
0
```

Q.6 Create a package CFE 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 sem 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 package in a file that declares the final marks of all students in all five courses?

Sol:

```
Package CFE;
public class student {
    String USN;
    String name;
    int sem;
    public student (String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}
public class Internals extends student {
    int[] internal_marks;
    public Internals (String USN, String name, int sem, int[] internal_marks) {
        super (usn, name, sem);
        this.internal_marks = internalMarks;
    }
}
```

```
package SEE;
import CIE.Student;
public class External extends Student {
    int[] seeMarks;
    public External (String usn, String name, int sem,
        int[] seeMarks) {
        super (usn, name, sem);
        this.seeMarks = seeMarks;
    }
}
```

```
import CIE.Internal;
import SEE.External;
public class finalMarks (String[] args) {
    int n=5;
    int[][] finalMarks = new int[n][5];
    Internal[] internalArray = new Internal [n];
    External[] externalArray = new External [n];
    for (int i=0; i<n; i++) {
        String USN = "USN" + (i+1);
        String name = "student" + (i+1);
        int sem = i;
        int[] internalMarks = {75, 80, 85, 90, 95};
        int[] seeMarks = {80, 85, 90, 95, 100};
        internalArray[i] = new Internal (USN, name,
            sem, internalMarks);
        externalArray[i] = new External (USN, name,
            sem, seeMarks);
        finalMarks[i][j] = internalArray[i].internalMarks
            [j] + externalArray[i].
            seeexternalMarks [j];
    }
}
```

```
System.out.println ("final marks for n students  
in all five courses :");  
for (int i=0; i<n; i++) {  
    System.out.println ("student " + (i+1) + ":");  
    for (int j=0; j < 5; j++) {  
        System.out.println ("finalMarks [" + i + "][" + j + "]");  
    }  
    System.out.println ();  
}  
}  
}  
}  
}
```

~~System.out.println();~~

~~System.out.println("final marks for n students in five courses");~~

output:

final marks for n students in five courses:

Student 1 :	144	207	181	73	209
Student 2 :	215	116	248	183	156
Student 3 :	70	46	79	70	131
Student 4 :	123	99	40	92	133
Students :	64	173	83	156	83

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

Program 7

wrote a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "son"

```
import java.util.Scanner;
class wrongAge extends Exception {
    public wrongAge (String message) {
        super (message);
    }
}
class Father {
    int fatherAge;
    public Father (int fatherAge) throws wrongAge {
        if (fatherAge < 0) {
            throw new wrongAge ("Age cannot be negative");
        }
        this.fatherAge = fatherAge;
    }
}
class son extends Father {
    int sonAge;
    public son (int fatherAge, int sonAge) throws wrongAge {
        super (fatherAge);
        if (sonAge >= fatherAge) {
            throw new wrongAge ("son's age must be less than Father's age");
        }
    }
}
```

```
this.sonAge = sonAge;
```

```
}
```

```
}
```

```
public class fatherSon {
```

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

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

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

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

```
        } catch (WrongAge e) {
```

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

```
        }
```

```
    }
```

```
}
```

~~Output:~~ Enter father's age and son's age :

~~48 21~~

~~father's age : 48~~

~~Son's age : 21~~

(28)

```
Enter father's age and son's age:  
45 25  
Father's age: 45  
Son's age: 25  
PS C:\Users\sarth\Downloads\Java lab programs\prog7> java fatherson  
Enter father's age and son's age:  
32 43  
Error: Son's age must be less than Father's age
```

Program-8

- Q. Write a program which creates two threads, one thread displaying "BMS" once every ten seconds and another displaying "CSE" once every two seconds.

```
class A extends Thread  
{  
    int t1=0, time;  
    A() {  
        t1 = 10000;  
        time = 21000;  
    }  
    public void run()  
    {  
        while (t1 <= time)  
        {  
            try {  
                sleep(1000);  
            }  
            catch (Exception e) {  
                System.out.println("error");  
            }  
            t1+=10000;  
        }  
    }  
}
```

~~class B extends Thread {~~
 ~~int t2=0, time;~~
 B() {
 time = 21000;
 t2 = 5000;
 }
}

```
public void run()
{
    while (t2 <= time)
    {
        System.out.println ("CSE");
        try {
            sleep (2000);
        }
        catch (Exception e)
        {
            System.out.println ("error");
        }
        t2 += 2000;
    }
}
```

8 Class Th

```
{ public static void main (String args[])
{
    A a = new A ();
    B b = new B ();
    a.start ();
    b.start ();
}
```

Output: BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS

College

of

Engineering

CSE

CSE

CSE

CSE

CSE

BMS

College

of

Engineering

```
New Thread Thread[#21,CSE,5,main]
New Thread Thread[#22,BMS College Of Engineering,5,main]
CSE : 0
BMS College Of Engineering : 0
CSE : 1
CSE : 2
CSE : 3
CSE : 4
BMS College Of Engineering : 1
CSE : 5
CSE : 6
CSE : 7
CSE : 8
CSE : 9
BMS College Of Engineering : 2
Exiting
BMS College Of Engineering : 3
BMS College Of Engineering : 4
BMS College Of Engineering : 5
BMS College Of Engineering : 6
BMS College Of Engineering : 7
BMS College Of Engineering : 8
BMS College Of Engineering : 9
Exiting
```

Q9 WAP 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 & Num2 is displayed in the result field when the divide button is clicked. If Num1 and Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class swingDemo {
    swing Demo() {
        JFrame jfrm = new JFrame ("Divider App");
        jfrm.setSize (275, 150);
        jfrm.setLayout (new FlowLayout ());
        jfrm.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
        JLabel jlab = new JLabel ("Enter the divisor  
and dividend:");
        JTextField ajtf = new JTextField (8);
        JTextField bjtf = new JTextField (8);
        JButton button = new JButton ("Calculate");
        JLabel err = new JLabel ();
        JLabel alab = new JLabel ();
        JLabel blab = new JLabel ();
        JLabel anslab = new JLabel ();
```

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

```
ActionListener l = new ActionListener() {
    public void actionPerformed (ActionEvent evt) {
        try {
            int a = Integer.parseInt (ajtf.getText ());
            int b = Integer.parseInt (btff.getText ());
            int ans = a/b;
            alab.setText ("\\nA = " + a);
            blab.setText ("\\nB = " + b);
            anslab.setText ("\\nAns = " + ans);
        }
        catch (NumberFormatException e) {
            alab.setText (" ");
            blab.setText (" ");
            anslab.setText (" ");
            err.setText ("Enter only Integer!");
        }
        catch (ArithmaticException e) {
            alab.setText (" ");
            blab.setText (" ");
            anslab.setText (" ");
            err.setText ("B should be non zero!");
        }
    }
};
```

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

output output | Dividend Divisor
 100 5
| Calculate Ans = 20



Divider App

-



Enter the divider and divident:

Calculate

A = 100 B = 5 Ans = 20