



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.

```
import java.util.Scanner;  
  
public class LAB {  
    public static void main(String args[]) {  
        Scanner sc = new Scanner(System.in);  
        int a, b, c;  
        double d, s1, s2;  
        System.out.println("Enter values of  
        a, b, c");  
  
        a = sc.nextInt();  
        b = sc.nextInt();  
        c = sc.nextInt();  
        sc.close();  
  
        d = (double)((b * b) - (4 * a * c));  
        if (a == 0) {  
            System.out.println("Invalid");  
            return;  
        }  
    }  
}
```

PTO

```

if (a*d<0){
    System.out.println("No real solutions");
} else if (d==0){
    s1 = (double)((-b+Math.sqrt(d))/(2*a));
    s2 = (double)((-b-Math.sqrt(d))/(2*a));
    System.out.printf("Roots are Real
and equal: %.4f and %.4f", s1, s2);
} else {
    s1 = (double)((-b+Math.sqrt(d))/(2*a));
    s2 = (double)((-b-Math.sqrt(d))/(2*a));
    System.out.printf("Roots are Real
and distinct: %.4f and %.4f, s1, s2);
}

```

```

srava@LAPTOP-NRIKOIFA ~\Documents\JavaStuff
> cd "c:\Users\srava\Documents\JavaStuff\" ; if ($?) { javac LAB.java } ; if ($?) { java LAB }
Enter values of a,b,c of a quadratic eqn: 1 1 1
No real solutions!
srava@LAPTOP-NRIKOIFA ~\Documents\JavaStuff
> cd "c:\Users\srava\Documents\JavaStuff\" ; if ($?) { javac LAB.java } ; if ($?) { java LAB }
Enter values of a,b,c of a quadratic eqn: 1 2 1
Roots are Real and Equal: -1.0000 and -1.0000
srava@LAPTOP-NRIKOIFA ~\Documents\JavaStuff
> cd "c:\Users\srava\Documents\JavaStuff\" ; if ($?) { javac LAB.java } ; if ($?) { java LAB }
Enter values of a,b,c of a quadratic eqn: 1 -1 -6
Roots are Real and Disinct: 3.0000 and -2.0000

```

Lab Program 2:

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

```
import java.util.Scanner;

class Student {
    Scanner sc = new Scanner(System.in);
    String USN;
    String Name;
    int credits[] = new int[5];
    int marks[] = new int[5];
    int SGPA;
    int totalCredits=0;

    void getDetails(){
        System.out.println("Enter student USN");
        USN = sc.nextLine();
        System.out.println("Enter name:");
        Name = sc.nextLine();
        for(int i=0; i<5; i++){
            System.out.print("Enter credits for subject " + (i+1) + ": ");
            credits[i] = sc.nextInt();
            totalCredits += credits[i];
            System.out.print("Enter marks for subject " + (i+1) + ": ");
            marks[i] = sc.nextInt();
        }
    }

    void displayDetails(){
        System.out.println("USN: " + USN);
        System.out.println("Name: " + Name);
        System.out.println("Total Credits: " + totalCredits);
        System.out.println("SGPA: " + SGPA);
    }
}
```

```
marks[i] = sc.nextFloat();
```

```
}
```

```
}
```

```
Void ShowDetails(){
```

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

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

```
for(i=0; i<5; i++) {
```

```
System.out.println("Subject " +(i+1) + "-
```

```
credits: " + credits[i] + "- Marks: "
```

```
+ Marks[i]);
```

```
}
```

```
}
```

```
Void CalcSGPA(){
```

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

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

```
if(Marks[i] < 0) {
```

```
System.out.println("Error: Marks  
below 0");
```

```
return;
```

```
}
```

```
PTO
```

```
{elseif (marks[i]<40){  
    points[i]=0;  
}  
{else if (marks[i]>40 && marks[i]<=50){  
    points[i]=4;  
}  
{else if (marks[i]>51 && marks[i]<=60){  
    points[i]=5;  
}  
{else if (marks[i]>61 && marks[i]<=70){  
    points[i]=7;  
}  
{else if (marks[i]>71 && marks[i]<=80){  
    points[i]=8;  
}  
{else if (marks[i]>81 && marks[i]<=90){  
    points[i]=9;  
}  
{else if (marks[i]>91 && marks[i]<=100){  
    points[i]=10;  
}  
{else {  
    System.out.println("Error marks  
alone 100");  
}}}
```

```
    SGPA += (marks[i] * credits[i]);  
}  
  
System.out.println("SGPA of " + Name +  
    " is: " + (float)(SGPA / totalCredits));  
}  
}
```

```
public class Lab2 {  
    public static void main(String args[]) {  
        Student stu1 = new Student();  
        stu1.getDetails();  
        stu1.showDetails();  
        stu1.calcSGPA;  
    }  
}
```

```
Enter Credits for Subject 5:  
3  
Enter Marks for Subject 5:  
75  
Student USN: 1BM19CS068  
Enter student name: Kalle Venata Sravan Dhira  
Subject 1 - Credits: 5 - Marks: 90.0  
Subject 2 - Credits: 4 - Marks: 78.0  
Subject 3 - Credits: 4 - Marks: 83.0  
Subject 4 - Credits: 4 - Marks: 81.0  
Subject 5 - Credits: 3 - Marks: 75.0  
SGPA of Kalle Venata Sravan Dhira is: 8.9
```

Week 5 Lab prog

Lab Program - 3

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

```
import java.util.Scanner  
class Book {  
    String name, author;  
    double price;  
    int num_pages;  
    Scanner sc = new Scanner(System.in);  
    Book() {}  
    Book(String name, String author, double price, int pages) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.num_pages = pages  
    }  
    void setDetails() {  
        System.out.println("Enter the name of the book");  
        this.name = sc.nextLine();  
        System.out.println("Enter author of the book");  
        this.author = author;  
        System.out.println("Enter price: ");  
        this.price = sc.nextDouble();  
    }  
    String toString() {  
        return "Book{" + "name=" + name + ", author=" + author + ", price=" + price + ", num_pages=" + num_pages + '}';  
    }  
}
```

```
System.out.print("Enter number of pages:");
    } thunum-pages = sc.nextInt();
void getDetails(){
    System.out.println("Name:" + Name);
    System.out.println("Author:" + Author);
    System.out.println("Price:" + price);
    System.out.println("Pages:" + pages);
}
```

```
public String toString(){
    return ("Name: " + name + "\n Author:" + author
        + "\n Price:" + price + "\n pages:" +
        pages);
}
```

```
Public class Labprog 3{
    public static void main(String args[]){
        int n;
        Book books[];
        Book myBook = new Book("Chamber of
            secrets", "J.K Rowling", 35.55, 867);
        Scanner sc = new Scanner(System.in);
```

```
System.out.print("Enter number of books");
```

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

```
books = new Book[n];
```

```
for (int i=0; i<books.length; i++) {
```

```
System.out.print(" Details of book" + i + ")");
```

```
books[i] = new Book();
```

```
books[i].setDetails();
```

```
}
```

```
for (i=0; i<books.length; i++) {
```

```
System.out.print(" Details of book" +  
(i+1) + " : ");
```

```
books[i].getDetails();
```

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

```
}
```

```
System.out.print(n(my Book));
```

```
}
```

```
Enter name of the book: Goblet Of Fire
```

```
Enter the author of the book: J.K Rowling
```

```
Enter the price: 35.55
```

```
Enter the number of pages: 354
```

```
Details of book 1:
```

```
Name: Order Of Phoenix
```

```
Author: J.K Rowling
```

```
Price: 35.55
```

```
Number of pages: 402
```

```
Details of book 2:
```

```
Name: Goblet Of Fire
```

```
Author: J.K Rowling
```

```
Price: 35.55
```

```
Number of pages: 354
```

```
Name: Chamber Of Secrets
```

```
Author: J.K Rowling
```

```
Price: 35.55
```

```
Number of pages: 367
```

Week 8

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

1) abstract class Shape{
 int b,h;
 Shape (int b, int h){
 this.b = b;
 this.h = h;
 }
 void printArea();
}

class Rectangle extends Shape{
 Rectangle (int b, int h){
 super(b,h);
 }
 void printArea(){
 System.out.println("Area of the rectangle is "+
 (b*h));
 }
}

class Triangle extends Shape{
 Triangle (int b, int h){
 super(b,h);
 }
 void printArea(){
 System.out.println("Area of
 the triangle is " + ((b*h)/2));
 }
}

Class Circle Extends Shape {

Circle (int r) {

super(r, r);

}

Void print Area() {

System.out.println ("Area of the circle is " +

Math.PI * r * r);

}

}

public class Prog1 {

public static void main (String args[]) {

Rectangle rect = new Rectangle (10, 5);

Triangle tri = new Triangle (10, 5);

Circle c = new Circle (10);

rect.print Area();

tri.print Area();

c.print Area();

}

```
cd "c:\Users\sjava\Documents\JavaStuff\Webs"
a Prog1
Area of the rectangle is 50
Area of the triangle is 25
Area of the circle is 314.1592653589793
```

}

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

- Accept deposit from customer 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

```
2) import java.util.Scanner;  
class abstract class Account {  
    String cName, accNum, acctype;  
    Scanner sc = new Scanner(System.in)  
    Account (String name, String accNo,  
             String acctype) {  
        this.cName = name;  
        this.accNum = accNo;  
        this.acctype = acctype;  
    }  
    Account () {}  
}  
  
class Current Acc extends Account {  
    private double balance = 5000, rate = 0.05;  
    private boolean conWithdraw = false;  
    Current Acc (String name, String accNo, String acctype) {  
        super(name, accNo, acctype);  
        System.out.println("Welcome "+cName);  
    }  
    void getBalance () {  
        System.out.format("Your balance : %.2f",  
                         balance);  
    }  
}
```

```
void deposit(double amount){  
    char choice;  
    System.out.println("Deposit Account holder :");  
    System.out.println("Name + " Amount: " + amount);  
    System.out.println("Approve Deposit (Y/N):");  
    choice = sc.next().charAt(0);  
    if (choice == 'y' || choice == 'Y') {  
        balance += amount;  
        System.out.println("Deposit approved. Updated  
balance: " + balance);  
    } else {  
        System.out.println("Deposit not approved.");  
    }  
}  
void withdraw(double amount){  
    System.out.println("This account cannot  
withdraw only funds");  
}  
void checkMinAmount(){  
    if (balance < 3000) {  
        balance -= 500;  
        System.out.print("Balance is below level ");  
        System.out.print("Penalty imposed");  
    }  
}
```

class Savings Acc extends Account {

private double balance = 5000, rate = 0.06;

Savings Acc (String name, String accNo, String
accType) {

super(name, accNo, accType);

}
System.out.println("Welcome " + name);

void getBalance () {

System.out.format("Your Balance : %.2f\n",
balance);

void deposit (double amount) {

char choice ;

System.out.println ("Deposit Options
Deposit : Account holder : " + name + " Amount
+ amount);

System.out.println ("Approve Deposit ? (Y/N) ! ");

choice = sc.next().charAt(0);

if (choice == ~~sc.next().charAt(0)~~; 'Y' || choice == 'y') {
balance += amount;

System.out.println (" Deposit approved .
Updated Balance : " + balance);

```
calc Interest();  
check Min amount();  
}  
else {  
    System.out.println(ANSI "Deposit not  
void calc Interest(){  
    double CI;  
    CI = balance => * (Math.pow((tirate/100), 2));  
    balance += CI;  
    System.out.println("Interest Added.");  
void withdraw(double Amount){  
    char choice;  
    if (balance < amount){  
        System.out.println("Account Balance  
is lower than Amount to be Withdrawn");  
    }  
    return;  
}  
System.out.println("Approve " + cName + " Is  
request for withdrawal?(Y/N) ");  
choice = sc.next().charAt(0);  
if (choice == 'Y' || choice == 'y') {  
    balance +- amount
```

System.out.println ("Withdrawal approved.

calc Interest(); Updated Balance: "+balance);
check min Amount();

} else {

System.out.println ("withdrawal not
approved");

}

}

void check min Amount(){

int min amount = 3000, penalty = 500;
if (balance < min amount){
balance -= penalty;

}

System.out.println ("Bal is under minimum
amount to be maintained");

System.out.println ("Penalty imposed");

}

}

```
public class Bank {  
    public static void main (String [] args) {  
        int c;  
        double temp;  
        String name, accNo, accType;  
        Scanner sc = new Scanner (System.in);  
        System.out.println ("Enter Name: ");  
        name = sc.nextLine ();  
        System.out.println ("Enter Account number: ");  
        accNo = sc.nextLine ();  
        System.out.println ("Enter type: ");  
        accType = sc.nextLine ();  
        if (accType.charAt (0) == 'C') {  
            CurrentAcc a = new CurrentAcc (  
                name, accNo, accType);  
            while (true) {  
                System.out.println ("1. deposit\n" +  
                    "2. Withdraw Money\n" +  
                    "3. Display Money\n" +  
                    "4. Exit");  
                c = sc.nextInt();  
                if (c == 1)  
                    a.deposit ();  
                else if (c == 2)  
                    a.withdraw ();  
                else if (c == 3)  
                    a.display ();  
                else if (c == 4)  
                    break;  
            }  
        }  
    }  
}
```

switch(c) {

case 1: {

System.out.println ("Enter amount
to be deposited : ");

temp = sc.nextDouble();

a. deposit (temp);
break;

}

case 2: { System.out.println ("Enter
amount to be withdrawn : ");

temp = sc.nextDouble();

a. withdraw (temp);
break;

}

case 3: {

a. getBalance ();

break;

}

case 4: {

System.exit (0);

break;

default: System.out.println ("Enter correct option");

}

}

else if (acctype.chatat(0) == 'S') {

Savings Acc a = new SavingsAcc (name, accNo,
acctype);

while (true) {

c = sc.nextInt();

switch (c) {

case 1 : {

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

temp = sc.nextDouble();

a.deposit (temp);

break;

}

case 2 : {

System.out.println ("Enter amount
to be withdrawn : ");

temp = sc.nextDouble();

a.withdraw (temp);

break;

3

case 3: {

a. getBalance();
break;

}

case 4: {

System.exit(0);
break;

}

default: System.out.println("Enter the
correct options");

}

}

} else {

System.out.println("Enter valid type... Exiting");

}

}

```
4. Exit  
1  
Enter amount to be deposited:  
2000
```

```
Deposit. Account holder: Sravan Amount: 2000.0  
Approve Deposit?(Y/S):  
Y
```

```
Deposit approved. Updated balance: 7000.0
```

```
Interest added. Updated balance: 14008.40252
```

- 1. Deposit money
- 2. Withdraw money
- 3. Display money
- 4. Exit

```
2
```

```
Enter amount to be withdrawn:
```

```
15000
```

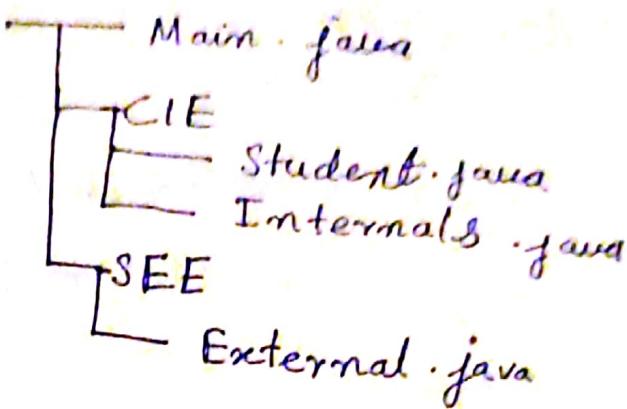
```
Account balance is lower than amount to be withdrawn
```

- 1. Deposit money
- 2. Withdraw money
- 3. Display money
- 4. Exit

Week - 9

File structure →

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



→ Student.java

```
package CIE;  
class Student {  
    public String name, usn;  
    public int sem;  
}
```

→ Internals.java

```
package CIE;  
import java.util.Scanner;  
public class Internals {  
    double internalMarks[] = new double [5];  
    Scanner sc = sc.nextInt();  
    public void setMarks() {  
        new Scanner(System.in);  
        for (int i = 0; i < internalMarks.length; i++) {  
            System.out.println("Enter Internal Marks (0) of  
subject " + (i + 1) + ": ");  
            internalMarks[i] = sc.nextDouble();  
        }  
    }  
}
```

```
    }  
    }  
    public void get Marks () {
```

```
        System.out.println("Intervals Marks out of 50");  
        for (int i = 0; i < intervalMarks.length; i++) {  
            System.out.println(" Subject " + (i + 1) +  
                " : " + intervalMarks[i]);  
        }  
    }
```

```
    }  
    public double get Marks by Index (int index) {  
        return intervalmarks [index];  
    }  
}
```

→ External External.java

```
package SEE;
```

```
import CIE.Student;
```

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

```
public class External extends Student {
```

```
    double  
    double [] externalMarks = new double [5];
```

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

PTO

```
public void setDetails() {
    System.out.println("Enter name: ");
    name = sc.nextLine();
    System.out.println("Enter USN: ");
    USN = sc.nextLine();
    System.out.println("Enter sem: ");
    sem = sc.nextInt();
}
```

```
public void setMarks() {
    for (int i = 0; i < externalMarks.length; i++) {
        System.out.println("Enter External Marks (100) of Subject " + (i + 1) + ": ");
        externalMarks[i] = sc.nextDouble();
    }
}
```

```
public void getDetails() {
    System.out.println("Name: " + name + "USN: " + USN
        + "Semester: " + sem);
}

public void PTO
```

```
public void getMarks () {  
    System.out.println ("External marks  
out of (50): ");  
  
    for (int i = 0; i < externalMarks.length; i++)  
        System.out.println ("Subject " + (i + 1) +  
                           + externalMarks[i]);  
}
```

{

{

→ Main.java

```
import java.util.Scanner;  
import CIE.Intervals;  
import SEE.Externals;
```

```
public class Main {
```

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

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

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

`n = sc.nextInt();`

`Internals[] cie = new Internals[n];`

`Externals[] see = new Externals[n];`

`for (int i = 0; i < see.length; i++) {`

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

`see[i] = new Externals();`

`see[i].getDetails();`

`System.out.println("Enter Internal Marks "+(i+1)+
":");`

`cie[i] = new Internals();`

`cie[i].setMarks();`

`System.out.println("Enter external marks of
student "+(i+1)+" : ");`

`see[i].setMarks();`

`}`

```
for (int i=0; i< see.length; i++) {
```

```
    System.out.println("Details of student");
```

```
    see[i].getDetails();
```

```
    System.out.println("Internal Marks of Student");
```

```
    see[i].cie.getMarks();
```

```
    System.out.println("External Marks : ");
```

```
    see.getMarks();
```

```
    System.out.println("Final Marks );
```

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

```
    System.out.println(" Subject " + (j+1) + " >> " +
```

```
    (cie[i].getMarksByIndex(j) + see[i].get  
    MarksByIndex(j));
```

```
}
```

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

```
}
```

```
sc.close();
```

```
}
```

```
Enter External Marks(100) of Subject 3:  
90  
Enter External Marks(100) of Subject 4:  
90  
Enter External Marks(100) of Subject 5:  
90  
Details of Student 1:  
Name: Sravan  
USN: 1BM19CS068  
Semester: 3  
Internal Marks of Student 1:  
Internals Marks out of 50:  
Subject 1 >> 36.0  
Subject 2 >> 36.0  
Subject 3 >> 36.0  
Subject 4 >> 36.0  
Subject 5 >> 36.0  
External Marks of Student 1:  
External Marks out of 50:  
Subject 1 >> 40.0  
Subject 2 >> 40.0  
Subject 3 >> 40.0  
Subject 4 >> 40.0  
Subject 5 >> 40.0  
Final Marks out of 100 of Student 1:  
Subject 1 >> 76.0  
Subject 2 >> 76.0  
Subject 3 >> 76.0  
Subject 4 >> 76.0  
Subject 5 >> 76.0
```

Lab program 7

Write a program to demonstrate generics with multiple object parameters.

```
class myGen <A,B> {
```

```
    A obj1;
```

```
    B obj2;
```

```
    my Gen(A obj1, B obj2) {
```

```
        this. obj1 = obj1;
```

```
        this. obj2 = obj2;
```

```
}
```

```
    public void print() {
```

```
        System.out.println(this.obj1);
```

```
        System.out.println(this.obj2);
```

```
}
```

```
}
```

```
public class labprog7 {
```

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

```
        myGen<String, Integer> myobj1 = new myGen<String, Integer>();
```

```
        myGen<String, Double> myobj2 = new myGen<String, Double>("Sravan", 19);
```

```
        myobj1.print();
```

```
        myobj2.print();
```

```
> cd "c:\Users\srava\Documents"
```

Sravan
19
F
69.0

Lecture program 8

import
java.util.Scanner

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

class WrongAge extends Exception {

 public WrongAge (String s) {

 super (s);

}

}

class Father ~~throws~~ WrongAge {

 int fatherAge;

 Father (int fAge, int sAge) throws WrongAge {

 if (fAge <= sAge) {

 throw new WrongAge ("Fathers age is less than or equal to Sons' age");

 } else {

 fatherAge = fAge;

}

}

```

class Son extends Father {
    int sonAge;
    Son (int fAge, int sAge) throws WrongAge {
        super(fAge, sAge);
        sonAge = sAge;
    }
    void print() {
        System.out.println("Father's age: " + fatherAge);
        System.out.println("Son's age: " + sonAge);
    }
}

```

```

Enter Father's age
35
Enter Sons's age
23
Father's Age: 35
Son's Age: 23
srava@LAPTOP-NRIKOIFA ~\Documents\JavaStuff\Week-10> cd "c:\Users\srava\Documents\JavaStuff\Week-10"; if (?) { javac Lab
Enter Father's age
34
Enter Sons's age
34
Exception WrongAge: Father's Age is equal to or less than Son's Age
srava@LAPTOP-NRIKOIFA ~\Documents\JavaStuff\Week-10>

```

```

public class LabProg8 {
    public static void main (String args[]) {
        int fAge, sAge;
        Scanner sc = new Scanner (System.in);
        System.out.println("Enter Father's age:");
        fAge = sc.nextInt();
        System.out.println("Enter Son's age:");
        sAge = sc.nextInt();
        try {
            Son son = new Son (fAge, sAge);
            son.print();
        } catch (WrongAge err) { System.out.printStackTrace(); }
    }
}

```

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.

class One implements Runnable {

 thread t1;

 String message;

 long time;

 One (String message, long time) {

 this.message = message;

 this.time = time;

 t1 = new Thread (this, "Thread");

 System.out.println ("One starts + t1);

 t1.start();

}

public void run () {

 try {

 for (int i=5; i>0; i--) {

 System.out.println (message)

 t1.sleep (time);

 } catch (InterruptedException e) {

 System.out.println ("One interrupted");

 }

 System.out.println ("Done");

}

}

```
public class PrintStuff {
    public static void main(String args[]) {
        One o = new One ("BMS College of Engineering",
                        10000);
        One t = new One ("CSE", 2000);
    }
}
```

```
One starts Thread[Thread,5,main]
One starts Thread[Thread,5,main]
BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College Of Engineering
One Done
```

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

```
class MyDialog extends Dialog implements  
ActionListener {
```

```
Prog1 p;
```

```
MyDialog (Frame parent, String title) {
```

```
super (parent, title, false);
```

```
setLayout (new FlowLayout());
```

```
setSize (300, 150);
```

```
P = (Prog1) parent;
```

```
Button b = new Button ("OK");
```

```
Label l = new Label ("Error : " + P.error);
```

```
add (l);
```

```
add (b);
```

```
} b.addActionListener (this);
```

```
public void actionPerformed (ActionEvent ae) {
```

```
dispose ();
```

```
}
```

```
}
```

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

```
public class Prog1 extends Frame implements ActionListener  
    Text Field Num1, Num2;  
    Button divide = new Button("Divide");  
    float res = 0;  
    public String error = "";  
    public Prog1(){  
        setLayout(new FlowLayout());  
        Num1 = new TextField(1);  
        Num2 = new TextField(1);  
        Label Num1L = new Label("Num1: ", Label.LEFT);  
        Label Num2L = new Label("Num 2: ", Label.RIGHT);  
        add(Num1L);  
        add(Num1);  
        add(Num2L);  
        add(Num2);  
        add(divide);  
        divide.add ActionListener(this);  
        addWindowListener(new WinAdapter());  
    }  
}
```

```
public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == divide) {
        try {
            int n1 = Integer.parseInt(Num1.getText());
            int n2 = Integer.parseInt(Num2.getText());
            if (n2 <= 0) {
                throw new ArithmeticException("Error");
            }
            res = (float) n1 / n2;
            repaint();
        } catch (NumberFormatException exception) {
            System.out.println(exception);
            res = 0;
            error = "Entered number is not an Integer.";
            repaint();
        } catch (ArithmeticException exception) {
            System.out.println(exception);
            res = 0;
            error = "You tried to divide by zero.";
            repaint();
        }
    }
}
```

```
    if (res == 0) {
        System.out.println("This is error");
    } else {
        MyDialog d = new MyDialog(this);
        d.setVisible(true);
    }
}
```

```
public void paint(Graphics g) {
    g.drawString("Result: " + String.valueOf(res),
                20, 100);
}
```

```
public static void main(String[] args) {
    Prog1 p = new Prog1()
    p.setSize(new Dimension(400, 200));
    p.setTitle("Divide");
    p.setVisible(true);
}
```

```
class WinAdapter extends WindowAdapter {  
    public void windowClosing(WindowEvent we) {  
        System.exit(0);  
    }  
}
```

}

