

**Lab Program:**

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 to find Roots of quadratic equation.

```
import java.util.Scanner;
import java.lang.Math;
class quadratic {
    public static void main (String ss[])
    {
        int a,b,c;
        double root1,root2;
        Scanner numsc = new Scanner (System.in);
        System.out.println("Enter the values of 'a', 'b' and 'c' for the
                           quadratic equation : ");
        a = numsc.nextInt();
        b = numsc.nextInt();
        c = numsc.nextInt();
        System.out.println("The quadratic equation is: " + a + "x^2" + b + "x" + c);
        d = b * b - 4 * a * c;
        if (d < 0) (d > 0)
        {
            root1 = (-b + Math.sqrt(d)) / (2 * a);
            root2 = (-b - Math.sqrt(d)) / (2 * a);
            System.out.println("The roots of the quadratic equation are real.");
            System.out.printf("Root 1 = %.4f", root1);
            System.out.printf("Root 2 = %.4f", root2);
        }
        else if (d == 0)
        {
            System.out.println("The roots of the quadratic equation are not real")
            root1 = (-b + Math.sqrt(d)) / (2 * a);
            root2 = (-b - Math.sqrt(d)) / (2 * a);
            System.out.println("The roots of the quadratic equation are equal
                               and real.");
            System.out.printf("Root 1 = %.4f \n Root 2 = %.4f", root1, root2);
        }
    }
}
```

else

{

System.out.println("The roots of the quadratic equation are  
not real");

}

}

}

```
Enter the values of 'a', 'b' and 'c' of the quadratic equation:
```

```
4 4 -4
```

```
The roots of the quadratic equation are real.
```

```
Root 1 = 0.6180
```

```
Root 2 = -1.6180
```

```
PS D:\Java Programs> cd 'd:\Java Programs'; & 'c:\Users\khush\doptOpenJDK\jdk-11.0.8.10-hotspot\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=y,address=5005' Data\Roaming\Code\User\workspaceStorage\ccf4ece568f6d40f2f82c79
```

```
Enter the values of 'a', 'b' and 'c' of the quadratic equation:
```

```
3 12 12
```

```
The quadratic equation is: 3x2+12x+12
```

```
The roots of the quadratic equation are real and equal.
```

```
Root 1 = -2.0000
```

```
Root 2 = -2.0000
```

```
PS D:\Java Programs> cd 'd:\Java Programs'; & 'c:\Users\khush\doptOpenJDK\jdk-11.0.8.10-hotspot\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=y,address=5005' Data\Roaming\Code\User\workspaceStorage\ccf4ece568f6d40f2f82c79
```

```
Enter the values of 'a', 'b' and 'c' of the quadratic equation:
```

```
1 1 1
```

```
The quadratic equation is: 1x2+1x+1
```

```
The roots of the quadratic equation are not real.
```

## **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 StudentDetails {
```

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

```
{
```

```
    Student s1 = new Student();
```

~~```
    s1.printDetails();
```~~

```
    s1.getDetails();
```

```
    s1.printDetails();
```

```
    s1.getSGPA();
```

```
}
```

```
}
```

```
class Student {
```

```
    private String USN, name;
```

```
    private int credits[], total_credits=0, gp=0;
```

```
    private double marks[];
```

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

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

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

```
void getDetail()
```

```
System.out.println("Enter the Student Details : ");
System.out.println("In USN : ");
USN = ss.next();
System.out.println("In Name : ");
name = ss.next();
for (int i=0; i<5; i++)
{
    System.out.printf("In Marks and credits of Subject-%d : ", i+1);
    marks[i] = ss.nextDouble();
    credits[i] = ss.nextInt();
}
```

```
void printDetail()
```

```
System.out.printf("The Student Details are : ");
System.out.printf("USN : %s", USN);
System.out.printf("Name : %s", name);
for (int i=0; i<5; i++)
{
    System.out.printf("In Subject %d Marks : %.f", i+1, marks[i]);
    System.out.printf("In Subject %d Credit : %.d", i+1, credits[i]);
}
```

```
void getGPA()
```

```
for (int i=0; i<5; i++)
{
    totalCredits += credits[i];
    if (marks[i] == 100.0)
        gp = (int) marks[i]/10;
    else
        gp = (int) ((marks[i]/10)+1);
    totalGPA += gp * credits[i];
}
```

Page \_\_\_\_\_  
Date \_\_\_\_\_

```
sgpa = total_gp / total_credits;  
System.out.println ("The SGPA of the student is " + sgpa);  
}  
}
```

Enter the Student Details:

USN:

1bm19cs072

Name:

Khushil

Marks and Credits of Subject 1 :80 4

Marks and Credits of Subject 2 :77 3

Marks and Credits of Subject 3 :68 3

Marks and Credits of Subject 4 :83 3

Marks and Credits of Subject 5 :92 2

The Student Details are:

USN: 1bm19cs072

Name: Khushil

Subject 1 Marks: 80.00

Subject 1 Credits: 4

Subject 2 Marks: 77.00

Subject 2 Credits: 3

Subject 3 Marks: 68.00

Subject 3 Credits: 3

Subject 4 Marks: 83.00

Subject 4 Credits: 3

Subject 5 Marks: 92.00

Subject 5 Credits: 2

The SGPA of the student is: 8.53

### **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.

## LAB 3 :- weeks :-

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

```
class Book
```

```
{
```

```
    private String name, author;
```

```
    private double price;
```

```
    private int numPages;
```

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

```
Book()
```

```
{
```

```
    name = "Harry Potter";
```

```
    author = "J.K Rowling";
```

```
    price = 1200.99;
```

```
    numPages = 550;
```

```
Book(String name, String auth, double cost, int np)
```

```
{
```

```
    name = name;
```

```
    author = author;
```

```
    price = cost;
```

```
    numPages = np;
```

```
}
```

```
public void getDetails()
```

```
    System.out.println("Enter the Book Details: ");
    System.out.println("Name: ");
    name = sc.next();
    System.out.println("Author: ");
    author = sc.next();
    System.out.println("Price: ");
    price = sc.nextDouble();
    System.out.println("Number of Pages: ");
    num_page = sc.nextInt();
}
```

```
public String toString()
```

```
{  
    return ("The Book Details are: \n" + "Book Title: " + name + "\n Author: " +  
           author + "\n Price: " + price + "\n Number of pages: " + num_page);  
}
```

```
class BookDetails {
```

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

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

```
    int n, ch;
```

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

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

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

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

```
{
```

```
        System.out.println("1: Enter values \n 2: Print Dettavalues values ");
```

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

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

```
        if (ch == 1)
```

```
{
```

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

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

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

```
}
```

else

{

b[i] = new Book();

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

}

}

5

}

Enter the number of Books:

2

1:Enter Values

2:Print Default Values

Enter the choice:

1

Enter the Book Details:

Name:

OliverTwist

Author:

CharlesDickens

Price:

500

Number of pages:

1500

The Book Details are:

Book Title: OliverTwist

Author: CharlesDickens

Price: 500.0

Number of Pages: 1500

1:Enter Values

2:Print Default Values

Enter the choice:

2

The Book Details are:

Book Title: Harry Potter

Author: J.K Rowling

Price: 1200.99

Number of Pages: 550

## Lab Exercises – 4 and 5 for 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.

## Lab program 4 :- week 8:-

1) abstract class shape  
{

int side1, side2;

double area

abstract void printArea();  
}

class Rectangle extends shape.  
{

Rectangle (int l, int b)  
{

side1 = l;

side2 = b;

}

void printArea()  
{

area = side1 \* side2;

System.out.println("The area of the Rectangle is: " + area);

}

}

class Triangle extends shape

{

Triangle (int h, int b)  
{

side1 = h;

side2 = b;

}

void printArea()  
{

area = 0.5 \* side1 \* side2;

System.out.println("The Area of the Triangle is: " + area);

}

}

class Circle extends Shape

{  
Circle (int r)

{  
side l = r;

}

void printArea ()

{

area = 3.14 \* side l \* side l;

System.out.println ("The Area of the Circle is: " + area);

}

}

class ShapeMain

{

public static void main (String args [])

{

Rectangle rect = new Rectangle (5, 6);

rect.printArea ();

Triangle tri = new Triangle (5, 8);

tri.printArea ();

Circle cir = new Circle (10);

cir.printArea ();

}

}

The area of the Rectagle is: 30.0  
The Area of the triangle is: 20.0  
The Area of the Circle is: 314.0

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

## Lab 5 program :- (week 8) :-

```
import java.util.Scanner;
import java.lang.Math;

class Account
{
    Scanner ss = new Scanner(System.in);
    String acc_name, acc_no;
    int acc_type;
    double balance;

    void CreateAccount()
    {
        System.out.println("Enter the Details of the new account:");
        System.out.printf("Name: ");
        acc_name = ss.next();
        System.out.printf("Ideal Account number: ");
        acc_no = ss.next();
        if (acc_type == 1)
        {
            System.out.printf("Enter the first Deposite Value: ");
            balance = ss.nextDouble();
            System.out.printf("Thank you for creating an Account.");
        }
        else
        {
            System.out.printf("Enter the first Deposite Value (above 1000): ");
            balance = ss.nextDouble();
            System.out.printf("Thank you for creating an Account. In\n"
                            "you will receive your cheque book.");
        }
    }
}
```

```
String get Account No()
{
    return acc-no;
}
```

```
void Display()
```

```
{
```

```
System.out.println("The Account Details are given as follows: ");
```

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

```
System.out.println("Account Number: " + acc-no);
```

```
if (acc-type == 1)
```

```
System.out.println("Account Type: Saving Accounts");
```

```
else
```

```
System.out.println("Account Type: Current Accounts");
```

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

```
}
```

```
}
```

```
class Sav-Acc extends Account
```

```
{
```

```
void withdraw()
```

```
{
```

```
double amount;
```

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

```
amount = ss.nextDouble();
```

```
balance -= amount;
```

```
}
```

```
void deposite()
```

```
{
```

```
double amount;
```

```
System.out.println("Enter the amount to be Deposited: ");
```

```
amount = ss.nextDouble();
```

```
balance += amount;
```

```
}
```

```
void compound_interest()
```

```
{
```

```
byte years_of_dep;
```

```
double interest;
```

```
System.out.println("Enter the number of years of compound interest: ");
```

```
years_of_dep = ss.nextInt();
```

```
interest = (balance * Math.pow(1 + 4.5/100, years_of_dep)) - balance;
```

```
System.out.println("The compound interest is: " + interest);
```

```
}
```

```
class Curr_Acc extends Account
```

```
{
```

```
    void withdraw()
```

```
{
```

```
    double amount;
```

```
    System.out.println("Warning: A minimum of 5000 balance must be maintained in it. If failed, a penalty of Rs 100 will be imposed.");
```

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

```
    amount = ss.nextDouble();
```

```
    balance -= amount;
```

```
    penaltycheck();
```

```
}
```

```
    void deposit()
```

```
{
```

```
    double amount;
```

```
    System.out.println("Enter the amount to be Deposited: ");
```

```
    amount = ss.nextDouble();
```

```
    balance += amount;
```

```
}
```

```
    void penaltycheck()
```

```
{
```

```
    if (balance < 5000)
```

```
{
```

```
        int pen = 100;
```

```
        System.out.println("The balance is less than 5000 a penalty of Rs.100 is imposed.");
```

```
        balance -= pen;
```

```
}
```

```
}
```

```
}
```

class Bank

{ public static void main (String args[])

{ Sav-Accet S-acct [] = new Sav-Accet [10];

Curr-Accet C-acct [] = new Curr-Accet [10];

Scanner ss = new Scanner (System.in);

String acctno;

int ch, i=0, j=0;

while (true)

{

System.out.println ("Welcome to the bank.\n");

System.out.println ("Enter the action to be performed: ");

System.out.println ("1: Create a Savings Account\n2: Create a Current Account.");

System.out.println ("3: Deposit\n4: Withdraw\n5: Display Balance\n6: Check Compound Interest");

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

ch = ss.nextInt();

switch (ch)

{

case 1: S-acct[i] = new Sav-Accet();

S-acct[i].acc-type = 1;

S-acct[i].CreateAccount();

i++;

break;

case 2: C-acct[i] = new Curr-Accet();

C-acct[i].acc-type = 2;

C-acct[i].CreateAccount();

i++;

break;

case 3: System.out.println ("Enter the account number: ");

acctno = ss.nextInt();

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

{

if (acctno.equals (c-acct[k].get Account No(1)))

{

System.out.println ("This Account is a current  
Account.");

c-acct[k].deposit();

}

}

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

{

if (acctno.equals (c-acct[k].get Account No(1)))

{

System.out.println ("This Account is a current Account.");

c-acct[k].deposit();

}

}

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

{

if (acctno.equals (s-acct[k].get Account No(1)))

{

System.out.println ("This Account is a Saving Account.");

s-acct[k].deposit();

}

}

break;

case 4: System.out.println ("Enter the account number: ");

acctno = ss.nextInt();

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

{

if (acctno.equals (c-acct[k].get Account No(1)))

{

System.out.println ("This Account is a current Account.");

c-acct[k].withdraw();

}

}

```

for (int k=0; k<i; k++)
{
    if (acctno.equals (s-acct[k].getAccountNo()))
        System.out.println ("This Account is a saving Account. ");
        s-acct [k]. withdraw();
    }
}

break;
}

case 5 : System.out.println ("Enter the account number");
acctno = ss.next();
for (int k=0; k<j; k++)
{
    if (acctno.equals (c-acct[k].getAccountNo()))
        c-acct [k]. display();
}

for (int k=0; k<i; k++)
{
    if (acctno.equals (s-acct[k].getAccountNo()))
        s-acct [k]. display();
}

break;
}

```

```

case 6 : System.out.println ("Enter the account number: ");
acctno = ss.next();
for (int k=0; k<j; k++)
{
    if (acctno.equals (c-acct[k].getAccountNo()))
        System.out.println ("This is a current account. In This account
does not provide interest");
}

for (int k=0; k<i; k++)
{
    if (acctno.equals (s-acct[k].getAccountNo()))
        s-acct [k]. compound_interest();
}

break;
}

```

Welcome to the bank.

Enter the action to be performed:

1: Create a Savings Account

2: Create a Current Account

3: Deposite

4: Withdraw

5:Display Balance

6: Check Compound Interest

Enter your choice: 1

Enter the Details of the new account:

Name: Raju

Ideal Account number: 12345

Enter the first Deposite Value: 2000

Thank you for creating an Account.

Welcome to the bank.

Enter the action to be performed:

1: Create a Savings Account

2: Create a Current Account

3: Deposite

4: Withdraw

5:Display Balance

6: Check Compound Interest

Enter your choice: 2

Enter the Details of the new account:

Name: Babu

Ideal Account number: 67890

Enter the first Deposite Value(above 5000):

6000

Thank you for creating an Account.

You will shortly receive your Cheque Book.

Welcome to the bank.

Enter the action to be performed:

1: Create a Savings Account

2: Create a Current Account

3: Deposite

4: Withdraw

5:Display Balance

6: Check Compound Interest

Enter your choice: 4

Enter the account number:

67890

This Account is a Current Account.

Warning: A minimum of 5000 balance must be maintained

If failed, a penalty of Rs.100 will be imposed.

Enter the Amount to be withdrawn:

2000

The balance is less than 5000 a penalty of Rs.100 is imposed.

Welcome to the bank.

Enter the action to be performed:

1: Create a Savings Account

2: Create a Current Account

3: Deposite

4: Withdraw

5:Display Balance

6: Check Compound Interest

Enter your choice: 5

Enter the account number:

12345

The Account Details are given as follows:

Name: Raju

Account Number: 12345

Account Type: Savings Account

Balance: 2000.0

---

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.

## Lab program 6 (week 9):-

② Student.java :- (..\\packs\\CIE\\Student.java)

package CIE;

import java.util.Scanner;

public class Student

    public ~~int~~ String usn, name;

    public int sem;

    public Scanner ss=new Scanner(System.in);

    public void getData()

}

        System.out.println("Name: ");

        name=ss.next();

        System.out.println("USN: ");

        usn=ss.next();

        System.out.println("Semester: ");

        sem=ss.nextInt();

}

}

Internals.java :- (..\\packs\\CIE\\Internals.java)

package CIE;

import java.util.Scanner;

public class Internals extends Student

{

    public int ciemarks[] = new int[5];

    public void getmarks()

{

        getData();

        System.out.print("Enter the CIE marks of the student: ");

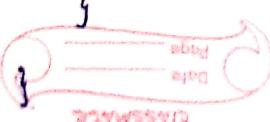
        for (int i=0; i<5; i++)

{

            System.out.print("Subject "+(i+1)+" : ");

            ciemarks[i] = ss.nextInt();

}



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

External.java:-(..\\pack<sup>3</sup>\\SEE\\External.java)

```
package SEE;
import java.util.Scanner;
public class External extends CIE.Student
{
    public int semarks[] = new int[5];
    Scanner ss = new Scanner(System.in);
    public void getmarks()
    {
        System.out.println("Enter the SEE marks of the student:");
        for(int int i=0; i<5; i++)
        {
            System.out.print("Subject " + (i+1) + ": ");
            semarks[i] = ss.nextInt();
        }
    }
}
```

Main.java:- (..\\packs\\ Main.java)

import java.util.Scanner;

import CIE.\*;

import SEE.\*;

class Main

{

public static void main (String args[])

Scanner ss=new Scanner (System.in);

int n;

int finalmarks []= new int [5];

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

n=ss.nextInt();

Internals studcie []= new Internals [n];

Externals studsee []= new Externals [n];

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

{

System.out.println ("\n x x x x x x x x x x x x \n");

studcie [i]= new Internals();

studsee [i]= new Externals();

studcie [i].getmarks();

studsee [i].getmarks();

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

finalmarks [j]= studcie [i].ciemarks [j] +  
(studsee [i].seemarks [j]/2);

}

System.out.println ("\n x x x x x x x x x x x x \n");

System.out.println ("The Student and their final marks are: \n");

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

{

studcie [i].display();

~~Final~~



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

```
    System.out.println("Subject " + (j+1) + ":" + finalmarks[j]);
```

```
    System.out.println("\nxx x x x x x x x x x\n");
```

```
}
```

**Enter the number of Students:**

2

\*\*\*\*\*

**Name:**

Khushil

**USN:**

1BM19CS072

**Semester:**

3

**Enter the CIE marks of the Student:**

Subject 1 :34

Subject 2 :50

Subject 3 :48

Subject 4 :46

Subject 5 :40

**Enter the SEE marks of the Student:**

Subject 1 :89

Subject 2 :

90

Subject 3 :92

Subject 4 :76

Subject 5 :78

\*\*\*\*\*

**Name:**

Sharat

**USN:**

1BM19CS072

**Semester:**

3

**Enter the CIE marks of the Student:**

Subject 1 :50

Subject 2 :48

Subject 3 :45

Subject 4 :43

Subject 5 :35

**Enter the SEE marks of the Student:**

Subject 1 :76

Subject 2 :78

Subject 3 :98

Subject 4 :80

Subject 5 :90

\*\*\*\*\*

The Students and their finals marks are:

Name: Khushil  
USN: 1BM19CS072  
Semester: 3  
Subject 1 :78  
Subject 2 :95  
Subject 3 :94  
Subject 4 :84  
Subject 5 :79

\*\*\*\*\*

Name: Sharat  
USN: 1BM19CS072  
Semester: 3  
Subject 1 :88  
Subject 2 :87  
Subject 3 :94  
Subject 4 :83  
Subject 5 :80

\*\*\*\*\*

**Write a program to demonstrate generics with multiple object parameters.**

## Lab program 7 :-

```
class famous<T, S>
{
    T mem1;
    S mem2;
    famous(T ob1, S ob2)
    {

```

```
        this.mem1 = ob1;
```

```
        this.mem2 = ob2;
    }
```

```
    public void displayItems()
    {
```

```
        System.out.println("The value of " + this.mem1 + " is : " + this.mem2);
    }
```

```
}
```

```
class FamousMain
```

```
{
```

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

```
        famous<String, String> std1 = new famous<String, String>("Name", "Elon musk");
```

```
        famous<String, Double> std2 = new famous<String, Double>
```

```
( "Networth in Billions ", 127.9 );
```

```
        famous<String, Integer> std3 = new famous<String, Integer>("Age", 49);
```

```
        std1.displayItems();
```

```
        std2.displayItems();
```

```
        std3.displayItems();
```

```
}
```

```
}
```

```
The value of Name is: Elon Musk  
The value of Net Worth in Billions is: 127.9  
The value of Age is: 49  
PS D:\Java Programs\Week 10> 
```

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 Wrong Age( ) 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.

## Lab program 8 :-

```
import java.util.*;  
class AgeException extends Exception  
{  
    public String toString()  
    {  
        return "Age of the Father cannot be lesser than the age of the  
        Son";  
    }  
}
```

~~class~~  
class Father

```
{  
    public Scanner ss = new Scanner (System.in);  
    int f-age;
```

```
    father()  
    {
```

```
        System.out.println ("Enter the age of the Father: ");
```

```
        f-age = ss.nextInt();
```

```
}
```

```
}
```

class Son extends Father

```
{
```

```
    int s-age;
```

```
    son()  
    {
```

```
        System.out.println ("Enter the Age of the Son : ");
```

```
        s-age = ss.nextInt();
```

```
}
```

```
}
```

class FamilyMain

{

    static void WrongAge (int fage, int sage) throws AgeException

{

        if (fage < sage)

{

            throw new AgeException();

}

    System.out.println ("There ~~were~~ were no other problems  
        arised during execution");

    System.out.println ("The Age of the Father is: " + fage);

} System.out.println ("The Age of the Son is: " + sage);

public static void main (String args [3])

{

    Son sn = new Son();

    try

    {

        WrongAge (sn.fage, sn.sage);

}

    catch (AgeException e)

{

        System.out.println ("An ~~Age~~ exception popped up: " + e);

}

}

}

```
Enter the Age of the Father:  
34  
Enter the Age of the Son:  
12  
There were no other problems arised during execution  
The age of the Father is: 34  
The age of the Son is: 12  
PS D:\Java Programs\Week 10> cd 'd:\Java Programs\Week 10'; & 'c:\Users\khush\.vscode\exten  
JDK\jdk-11.0.8.10-hotspot\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,susp  
Storage\2b4e8dbe371b3dd5a4fa074dc92557af\redhat.java\jdt_ws\Week 10_d36af370\bin' 'Family  
Enter the Age of the Father:  
12  
Enter the Age of the Son:  
34  
An Exception popped up: Age of the Father Cannot be lesser than the age of the Son  
PS D:\Java Programs\Week 10> []
```

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.

Week 11 :-

Lab program 9 :-

class NewThread1 implements Runnable

{

    Thread t;

    String ~~sth~~ name;

    long time;

    NewThread1(String stname, long thtime)

{

        time = thtime;

        name = stname;

        t = new Thread(this, name);

        t.start();

}

    public void run()

{

        try

{

            for (int n=10; n>0; n--)

{

                System.out.println(t.getName());

                Thread.sleep(time);

}

}

    catch (InterruptedException ie)

{

        System.out.println("Child Thread Interrupted");

}

        System.out.println("Child Thread Quitting....");

}

}

```
class ThreadMain
{
    public static void main(String args[])
    {
        NewThreads1 n1 = new NewThreads1("BMS College of Engineering", 1000);
        NewThreads1 n2 = new NewThreads1("CSE", 2000);
    }
}
```

BMS College of Engineering.

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering.

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering.

Child Thread Quitting.....

BMS College of Engineering.

Child Thread Quitting.....

---

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.

## Lab program 10 :- (Week 12)

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

class NumException extends Exception {
    public String toString() {
        return "There is an Arithmetic Exception.";
    }
}
```

```
class FormatException extends Exception {
    public String toString() {
        return "There is an Format Exception.";
    }
}
```

```
public class dividenum extends frame implements ActionListener {
```

```
    JTextField num1, num2;
```

```
    Button div;
```

```
    double result;
```

```
    String msg = "The result is: 0.0";
```

```
    public dividenum()
```

```
{
```

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

```
    Label num1n = new Label("Numerator: ", Label.RIGHT);
```

```
    Label num2n = new Label("Denominator: ", Label.RIGHT);
```

```
    Button div = new Button("Divide");
```

```
    num1 = new JTextField(5);
```

```
    num2 = new JTextField(5);
```

```
    add(num1n);
```

```
    add(num1);
```

```
    add(num2n);
```

```
    add(num2);
```

```
    add(div);
```

```
num1.addActionListener(this);
num2.addActionListener(this);
div.addActionListener(this);
```

```
addWindowListener(new WindowAdapter()
{
```

```
public void windowClosing(WindowEvent we)
{
```

```
    System.exit(0);
}
```

```
} );
```

```
public boolean isDouble (double num)
{
```

```
    double dec;
```

```
    dec = num - (int) num;
```

```
    if (dec == 0.0)
```

```
        return true;
```

```
    else
```

```
        return false;
    }
```

```
public double divide (double a, double b) throws -
```

```
- NumException , FormatException
```

```
{
```

```
if (b == 0.0)
```

```
{
```

```
    throw new NumException();
```

```
}
```

```
else if ( !isDouble(a) || !isDouble(b) )
```

```
{
```

```
    throw new FormatException();
```

```
}
```

```
return (double) a/b;
```

```
}
```

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

```
{
```

```
    double a, b;
```

```
    a = Double.parseDouble (num1.getText ());
```

```
    b = Double.parseDouble (num2.getText ());
```

```
    try {
```

```
        result = divide (a, b);
```

```
        msg = ("The result is: " + result);
```

```
}
```

```
    catch (NumberFormatException ne)
```

```
{
```

```
        msg = ne.toString ();
```

```
}
```

```
    catch (FormatException fe)
```

```
{
```

```
        msg = fe.toString ();
```

```
}
```

```
    repaint ();
```

```
}
```

```
public void paint (Graphics g)
```

```
{ ResultDialog d = new ResultDialog (this, "Result")
```

~~```
d.setVisible (true);
```~~

```
}
```

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

```
{
```

```
    dividenum appwin = new dividenum ();
```

```
    appwin.setSize (new Dimension (350, 300));
```

```
    appwin.setTitle ("Divide Two Numbers");
```

```
    appwin.setVisible (true);
```

```
}
```

```
class ResultDialog extends Dialog implements ActionListener {  
    divide nums pt;
```

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

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

```
        pt = (divide nums) parent;
```

```
        setLayout (new Flow Layout());
```

```
        setSize (250,100);
```

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

```
        add (new Label (pt.nsg));
```

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

```
        add (b);
```

```
        addWindowListener (new WindowAdapter ()) {
```

```
            public void windowClosing (WindowEvent we) {
```

```
                dispose();
```

```
            }
```

```
        };
```

```
}
```

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

```
{
```

```
            dispose();
```

```
}
```

```
}
```



## Divide Two Numbers



Numerator:

12

Denominator:

3

Divide



## Result



The result is: 4.0

OK



## Divide Two Numbers

-



Numerator:

Denominator:

Divide



Result



There is an Arithmetic Exception.

OK



## Divide Two Numbers



Numerator:

Denominator:

Divide



### Result



There is an Formal Exception.

OK