

LAB PROGRAM-1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
class quad
{
    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in);
        System.out.println ("The quadratic equation  
is  $a x^2 + b x + c = 0$ ");
        System.out.println ("Enter the value a");
        int a = s.nextInt();
        System.out.println ("Enter the value b");
        int b = s.nextInt();
        System.out.println ("Enter the value c");
        int c = s.nextInt();
        double d = Math.pow(b, 2) - (4 * a * c);
        if (d > 0)
        {
            System.out.println ("The roots are real  
and unequal");
            r1 = (-b + Math.sqrt(d)) / (2 * a);
            r2 = (-b - Math.sqrt(d)) / (2 * a);
            System.out.println ("The roots are  
" + r1 + " and " + r2);
        }
        else
        {
            System.out.println ("The roots are  
real and equal");
            r1 = -b / (2 * a);
            System.out.println ("The roots are  
" + r1);
        }
    }
}
```

else if ($d == 0$)
{

System.out.println("The roots are
real and equal");

$r1 = (-b) / (deno);$

$r2 = (-b) / (deno);$

System.out.println("The roots are"
+ r1 + "and" +
r2);

y

else

{

System.out.println("There are no
real solutions for
the equation");

f

f

A screenshot of a Java code editor window titled "quadratic.java". The code implements a quadratic equation solver. It uses a Scanner to read input values for coefficients a, b, and c. It calculates the discriminant d = b^2 - 4ac. Based on the value of d, it prints different messages and calculates the roots r1 and r2 using the quadratic formula. The code is written in Java syntax with standard imports.

```
import java.util.Scanner;
class quad
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("THE QUADRATIC EXPRESSION IS ax^2+bx+c");
        System.out.println("ENTER THE VALUE OF a");
        int a=s.nextInt();
        System.out.println("ENTER THE VALUE OF b");
        int b=s.nextInt();
        System.out.println("ENTER THE VALUE OF c");
        int c=s.nextInt();
        double d=Math.pow(b,2)-(4*a*c);
        int deno=2*a;
        double r1,r2;
        if(d>0)
        {
            System.out.println("THE ROOTS ARE REAL AND UNEQUAL");
            r1=(-b+Math.sqrt(d))/(deno);
            r2=(-b-Math.sqrt(d))/(deno);
            System.out.println("the roots are "+r1+"and"+r2);
        }
        else if(d==0)
        {
            System.out.println("THE ROOTS ARE REAL AND EQUAL");
            r1=(-b)/(deno);
        }
        else
        {
            System.out.println("THERE ARE NO REAL SOLUTIONS FOR THE EQUATION");
        }
    }
}
```

A screenshot of a Java code editor window titled "quadratic.java". The code is identical to the one in the first screenshot, implementing a quadratic equation solver. It reads coefficients a, b, and c, calculates the discriminant d, and prints the roots r1 and r2 based on the value of d. The code uses standard Java syntax and imports.

```
System.out.println("ENTER THE VALUE OF c");
int c=s.nextInt();
double d=Math.pow(b,2)-(4*a*c);
int deno=2*a;
double r1,r2;
if(d>0)
{
    System.out.println("THE ROOTS ARE REAL AND UNEQUAL");
    r1=(-b+Math.sqrt(d))/(deno);
    r2=(-b-Math.sqrt(d))/(deno);
    System.out.println("the roots are "+r1+"and"+r2);
}
else if(d==0)
{
    System.out.println("THE ROOTS ARE REAL AND EQUAL");
    r1=(-b)/(deno);
    r2=(-b)/(deno);
    System.out.println("the roots are "+r1+"and"+r2);
}
else
{
    System.out.println("THERE ARE NO REAL SOLUTIONS FOR THE EQUATION");
}
```

```
c:\JAVA\bin>javac quad.java
javac: file not found: quad.java
Usage: javac <options> <source files>
use -help for a list of possible options

c:\JAVA\bin>javac quadratic.java

c:\JAVA\bin>java quad
THE QUADRATIC EXPRESSION IS ax^2+bx+c
ENTER THE VALUE OF a
1
ENTER THE VALUE OF b
1
ENTER THE VALUE OF c
1
THERE ARE NO REAL SOLUTIONS FOR THE EQUATION

c:\JAVA\bin>java quad
THE QUADRATIC EXPRESSION IS ax^2+bx+c
ENTER THE VALUE OF a
1
ENTER THE VALUE OF b
4
ENTER THE VALUE OF c
4
THE ROOTS ARE REAL AND EQUAL
the roots are -2.0and-2.0
```

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.

Lab program-2

```
import java.util.Scanner;
class student
{
    private String usn;
    private String name;
    private int[] credit;
    private int[] marks;
    void getdetails()
    {
        System.out.println("Enter details of student");
        Scanner s = new Scanner(System.in);
        System.out.print("Enter name");
        name = s.nextLine();
        System.out.print("Enter usn");
        usn = s.nextLine();
        credit = new int[4];
        marks = new int[4];
        for (int i = 0; i < 4; i++)
        {
            System.out.print("Enter credit of subject " + (i + 1));
            credit[i] = s.nextInt();
        }
        for (int i = 0; i < 4; i++)
        {
            System.out.print("Enter mark of subject " + (i + 1));
            marks[i] = s.nextInt();
        }
    }
    void printdetails()
    {
    }
```

```
SOP("Usn of student is = " + usn);
```

```
SOP("Name of student is " + name);
```

```
for (int i = 0; i < 4; i++)
```

```
{  
    SOP("Mark of sub " + i + " is " + marks[i]);  
}
```

```
for (int i = 0; i < 4; i++)
```

```
{  
    SOP("Credit of subject " + i + " is " + credit[i]);  
}
```

```
Void calculate()
```

```
int [] total = new int [4];
```

```
double tc = 0.0;
```

```
double t = 0.0;
```

```
double sgpa = 0.0;
```

```
for (int i = 0; i < 4; i++)
```

```
{  
    If (marks[i] >= 85 && marks[i] <= 100)
```

```
        total[i] = credit[i] + 10;
```

```
    else if (marks[i] >= 70 && marks[i] < 85)
```

```
        total[i] = credit[i] + 9;
```

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

```
        total[i] = credit[i] + 8;
```

else if (Mark[i] >= 50 && Mark[i] < 60)

total[i] = credit[i] + 7;

else

total[i] = credit[i] + 6;

}

{

tc = tc + credit[i];

}

for (int i = 0; i < 4; i++)

{

t = t + total[i];

}

sgpa = (t / (tc + 0));

System.out.println("The sgpa of student is " + sgpa);

}

}

class student main

{

student st = new student();

st.getdetails();

st.printdetails();

st.calculate();

}

}

```
VA > bin > sgpajava > ...
import java.util.Scanner;
class Student
{
    private String usn;
    private String name;
    private int[] credit;
    private int[] marks;
    void getdetails()
    {
        System.out.println("ENTER THE DETAILS OF THE STUDENTS");
        Scanner s=new Scanner(System.in);
        System.out.println("ENTER NAME OF STUDENT");
        name=s.nextLine();
        System.out.println("ENTER THE USN OF STUDENT");
        usn=s.nextLine();
        credit=new int[4];
        marks=new int[4];
        for(int i =0;i<4;i++)
        {
            System.out.println("ENTER THE CREDIT OF SUBJECT"+(i+1));
            credit[i]=s.nextInt();
        }
        for(int i =0;i<4;i++)
        {
            System.out.println("ENTER THE MARK OF SUBJECT"+(i+1));
            marks[i]=s.nextInt();
        }
    }
    void printdetails()
    {
        System.out.println("THE USN OF STUDENT IS="+usn);
```

The screenshot shows a Visual Studio Code interface with the title bar "sgpajava - Visual Studio Code". The left sidebar shows a file tree with files like login.html, login.css, adminlogin.html, login.js, admin.js, sgpajava.java (which is currently selected), and player.java. The main editor area contains Java code for calculating SGPA based on credits and marks.

```
C: > JAVA > bin > sgpajava > ...
38     {
39         System.out.println(" THE CREDITS OF SUBJECT"+(i+1)+"="+credit[i]);
40     }
41     void calculate()
42     {
43         int[] total=new int[4];
44         double tc=0.0;
45         double t=0.0;
46         double sgpa=0.0;
47         for(int i=0;i<4;i++)
48         {
49             if(marks[i]>=85 && marks[i]<=100)
50                 total[i]=credit[i]*10;
51             else if(marks[i]>=70 && marks[i]<85)
52                 total[i]=credit[i]*9;
53             else if(marks[i]>=60 && marks[i]<70)
54                 total[i]=credit[i]*8;
55             else if(marks[i]>=50 && marks[i]<60)
56                 total[i]=credit[i]*7;
57             else
58                 total[i] =credit[i]*6;
59         }
60         for(int i=0;i<4;i++)
61         {
62             tc=tc+credit[i];
63         }
64         for(int i=0;i<4;i++)
65         {
66             t=t+total[i];
67         }
68         sgpa=(t/(tc+0.0));
69         System.out.println("THE SGPA OF STUDENT IS="+sgpa);
70     }
71 }
72 class studentmain
```

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "aved" containing files like login.html, login.css, adminlogin.html, login.js, admin.js, sgpa.java, and player.java.
- Code Editor:** Displays Java code for calculating SGPA. The code defines a class `studentmain` with a `main` method. It creates a `Student` object `s1`, gets details, prints them, and calculates the SGPA by summing four subjects and dividing by 4.0.

```
C:\>JAVA>bin> sgpa.java > ...
65     for(int i=0;i<4;i++)
66     {
67         t=t+total[i];
68     }
69     sgpa=(t/(tc*4.0));
70     System.out.println("THE SGPA OF STUDENT IS="+sgpa);
71 }
72 }
73 class studentmain
74 {
75     public static void main(String args[])
76     {
77         Student s1=new Student();
78         s1.getdetails();
79         s1.printdetails();
80         s1.calculate();
81     }
82 }
83
84
85
86
87
88
89
90
```

The screenshot shows a terminal window displaying the execution of the Java program and its output. The user inputs marks for four subjects, and the program calculates the SGPA.

```
\$0
ENTER THE MARK OF SUBJECT1
90
ENTER THE MARK OF SUBJECT2
50
ENTER THE MARK OF SUBJECT3
90
ENTER THE MARK OF SUBJECT4
74
THE USN OF STUDENT IS=45
THE NAME OF STUDENT IS f
THE MARK OF SUBJECT1=54
THE MARK OF SUBJECT2=50
THE MARK OF SUBJECT3=90
THE MARK OF SUBJECT4=74
THE CREDITS OF SUBJECT1=5
THE CREDITS OF SUBJECT2=5
THE CREDITS OF SUBJECT3=5
THE CREDITS OF SUBJECT4=5
THE SGPA OF STUDENT IS=8.25

c:\JAVA\bin>
```

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.

Info about `toString()` method:

```
class Test
{
    int a=10;

    public String toString() // when an object is printed this method is automatically called
    { return("a=" + a); } //return statement with String is required
}
```

```
class TestMain
{
    public static void main (String ss[])
    {
        Test t1=new Test ();
        System.out.println (t1);
        //t1 - First example where an object is printed directly
        //calls toString method with the object given as the invoking object
    }
}
```

LAB 3 program

```
import java.util.Scanner;
class Book
{
    private String name;
    private String auth;
    private double p;
    private int np;

    Book()
    {
        name=null;
        auth=null;
        p=0.0;
        np=0;
    }

    void getdetails()
    {
        Scanner s=new Scanner(System.in);

        System.out.println("Enter title of the book");
        name=s.nextLine();
        System.out.println("Enter author of the book");
        auth=s.nextLine();
        System.out.println("Enter price of book");
        p=s.nextDouble();
        System.out.println("Enter no of pages of book");
        np=s.nextInt();
    }
}
```

```
public String toString()
```

{

```
return ("The title of book: " + name + " if " + "Author of  
book: " + author + " if " + "Price of book: " + price  
+ " " + "No. of pages: " + np);
```

}

```
class Bookmain
```

{

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

{

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

```
SOP("Enter no. of books");
```

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

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

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

{

```
SOP("Enter details of book " + (i + 1));
```

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

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

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

```
{ SOP("details of book " + (i + 1));
```

```
    SOP(b[i]);
```

}

Go Run Terminal Help

book.java - bin - Visual Studio Code

```
player.java book1.java bookJava employee.java age.java distance.java sgp.java
```

```
book.java > Book > getdetails()
1 import java.util.Scanner;
2 class Book
3 {
4     private String name;
5     private String auth;
6     private double p;
7     private int np;
8     Book()
9     {
10         name=null;
11         auth=null;
12         p=0.0;
13         np=0;
14     }
15     void getdetails()
16     {
17         Scanner s=new Scanner(System.in);
18         System.out.println("ENTER THE TITLE OF BOOK");
19         name=s.next();
20         System.out.println("ENTER AUTHOR OF BOOK");
21         auth=s.next();
22         System.out.println("ENTER PRICE OF BOOK");
23         p=s.nextDouble();
24         System.out.println("ENTER NUMBER OF PAGES OF BOOK");
25         np=s.nextInt();
26     }
27     public String toString()
28     {
29         return("THE TITLE OF BOOK:"+name+ " "+ "AUTHOR OF BOOK:"+auth+ " "+ " PRICE OF BOOK: "+p+ " "+ " NUMBER OF PAGES OF BOOK:"+np);
30     }
31 }
32 class Bookmain
33 {
    Run | Debug
    public static void main(String args[])
    {
        Scanner ss=new Scanner(System.in);
        System.out.println("ENTER THE NUMBER OF BOOKS");
    }
}
```

Ln 26, Col 6 Spaces: 4 UTF-8 CRLF Java JavaSE-14

JNSAVED

```
book.java > Book > getdetails()
27     | public String toString()
28     {
29         return("THE TITLE OF BOOK:"+name+ " "+ "AUTHOR OF BOOK:"+auth+ " "+ " PRICE OF BOOK: "+p+ " "+ " NUMBER OF PAGES OF BOOK:"+np);
30     }
31 }
32 class Bookmain
33 {
    Run | Debug
    public static void main(String args[])
    {
        Scanner ss=new Scanner(System.in);
        System.out.println("ENTER THE NUMBER OF BOOKS");
        int n=ss.nextInt();
        Book b[]=new Book[n];
        for(int i=0;i<n;i++)
        {
            System.out.println("ENTER THE DETAILS OF BOOK"+(i+1));
            b[i]=new Book();
            b[i].getdetails();
        }
        for(int i=0;i<n;i++)
        {
            System.out.println("THE DETAILS OF BOOK"+(i+1));
            System.out.println(b[i]); /*SOP(b[i].toString()) will also work*/
        }
    }
}
```

```
c:\JAVA\bin>java Bookmain
ENTER THE NUMBER OF BOOKS
2
ENTER THE DETAILS OF BOOK1
ENTER THE TITLE OF BOOK
ad
ENTER AUTHOR OF BOOK
ed
ENTER PRICE OF BOOK
234
ENTER NUMBER OF PAGES OF BOOK
56
ENTER THE DETAILS OF BOOK2
ENTER THE TITLE OF BOOK
sdf
ENTER AUTHOR OF BOOK
ew
ENTER PRICE OF BOOK
234
ENTER NUMBER OF PAGES OF BOOK
45
THE DETAILS OF BOOK1
THE TITLE OF BOOK:ad AUTHOR OF BOOK:ed PRICE OF BOOK: 234.0 NUMBER OF PAGES OF BOOK:56
THE DETAILS OF BOOK2
THE TITLE OF BOOK:sdf AUTHOR OF BOOK:ew PRICE OF BOOK: 234.0 NUMBER OF PAGES OF BOOK:45
c:\JAVA\bin>
```

LAB PROGRAM-4

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 4 - program

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

```
abstract class Shape {
```

```
    double a;
```

```
    double b;
```

```
    Shape(double a, double b)
```

```
{
```

```
    this.a = a;
```

```
    this.b = b;
```

```
}
```

```
    abstract double area();
```

```
}
```

```
class Rectangle extends Shape
```

```
{
```

```
    Rectangle(double a, double b)
```

```
{
```

```
    super(a, b);
```

```
}
```

```
    double area()
```

```
{
```

```
    System.out.println("Area of Rectangle - ");
```

```
    return a * b;
```

```
}
```

```
}
```

EDGE

Class Triangle extends Shape

{
Triangle (double a, double b)

{
super (a, b);

double area()

{
System.out.println ("Area of triangle");

return (a+b)/2;

}

Class Circle extends Shape {

final double pi = 3.14;

Circle (double a, double b) {

super (a, b); }

double area()

{
System.out.println ("Area of Circle");

return pi * Math.pow (a, 2);

}

}

class shapein

{

public static void main (String args[])

{

Scanner s = new Scanner (System.in);

sop ("Enter length & breadth of
rectangle");

double l = s.nextDouble();

double b = s.nextDouble();

rectangle rec = new Rectangle (l, b);

sop ("Enter height and base of
triangle");

double h = s.nextDouble();

double ba = s.nextDouble();

Triangle tri = new Triangle (h, ba);

sop ("Enter radius of circle");

double r = s.nextDouble();

Circle c = new Circle (r, 2 * pi);

Shape sh;

sh = rec;

alternate

SOP (sh.area());
sh = tri;
SOP (sh.area());
sh = c;
SOP (sh.area());
}

```
shapejava > ⚡ shapemain > ⏱ main(String[])
1 import java.util.Scanner;
2 abstract class Shape {
3     double a;
4     double b;
5     Shape(double a, double b)
6     {
7         this.a = a;
8         this.b = b;
9     }
10    abstract double area();
11 }
12
13 class Rectangle extends Shape
14 {
15     Rectangle(double a, double b)
16     {
17         super(a, b);
18     }
19     double area()
20     {
21         System.out.println("Area of Rectangle.");
22         return a*b;
23     }
24 }
```

```
# submitdetails.css      JS submitdetails.js      student1.html      Bank.java      shape.java      player.java
shape.java > shapemain > main(String[])
System.out.println("Area of Rectangle.");
return a*b;
}
}
class Triangle extends Shape {
Triangle(double a, double b) {
super(a, b);
}
double area() {
System.out.println(" Area of Triangle.");
return (a*b)/ 2;
}
}

class Circle extends Shape {
final double pi=3.14;
Circle(double a,double b) {
super(a, b);
}
double area() {
System.out.println(" Area of Circle.");
return pi*Math.pow(a,2);
}
}
```

Action View Go Run Terminal Help

shape.java - bin - Visual Studio Code

1 UNSAVED

```
... # submitdetails.css JS submitdetails.js student1.html Bank.java shape.java player.java
shape.java > shapemain > main(String[])
44 }
45 class shapemain
46 {
47     Run | Debug
48     public static void main [String args[]]
49     {
50         Scanner s=new Scanner(System.in);
51         System.out.println("ENTER LENGTH AND BREADTH OF RECTANGLE");
52         double l=s.nextDouble();
53         double b=s.nextDouble();
54         Rectangle rec = new Rectangle(l,b);
55         System.out.println("ENTER HEIGHT AND BASE LENGTH OF TRIANGLE");
56         double h=s.nextDouble();
57         double ba=s.nextDouble();
58         Triangle tri = new Triangle(h,ba);
59         System.out.println("ENTER RADIUS OF CIRCLE");
60         double r=s.nextDouble();
61         Circle c=new Circle(r,2*r);
62         Shape sh;
63         sh=rec;
64         System.out.println( sh.area());
65         sh=tri;
66         System.out.println( sh.area());
67         sh=c;
68         System.out.println( sh.area());
69     }
70 }
```

PROBLEMS 73 OUTPUT DEBUG CONSOLE TERMINAL

Filter (e.g. text,)

- shape.java 1
 - Resource leak: 's' is never closed Java[536871799] [49, 15]
- actor.java 2
 - Resource leak: 's' is never closed Java[536871799] [10, 17]
 - Resource leak: 'ss' is never closed Java[536871799] [35, 17]

```
# submittodefault.css          # submittodefault.js          # studentdetails.html      # login.html  
① shape.java > ⚡ shapemain > ⌂ main(String[])  
60   | Circle c=new Circle(r,2*r);  
61   | Shape sh;  
62   | sh=rec;  
63   | System.out.println( sh.area());  
64   | sh=tri;  
65   | System.out.println( sh.area());  
66   | sh=c;  
67   | System.out.println( sh.area());  
68 }  
69 }
```

```
① 4-exit  
ENTER THE CHOICE  
④  
④  
④ c:\JAVA\bin>java shapemain  
ENTER LENGTH AND BREADTH OF RECTANGLE  
④ 12  
④ 12  
④ ENTER HEIGHT AND BASE LENGTH OF TRIANGLE  
④ 2  
④ 4  
④ ENTER RADIUS OF CIRCLE  
④ 5  
④ Inside of Rectangle.  
④ 144.0  
④ Area for Triangle.  
④ 4.0  
④ Area for Circle.  
④ 78.5  
④  
④ c:\JAVA\bin>javac Bank.java  
④  
④ c:\JAVA\bin>java bank  
④ 1-CURRENT ACCOUNT  
④ 2-SAVINGS ACCOUNT
```

LAB PROGRAM-5

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

- 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 Program - 5

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

```
abstract class Account
```

```
{
```

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

```
    String name;
```

```
    long num;
```

```
    String type;
```

```
    double bal;
```

```
    Account() { }
```

```
    Account(String name, long num, String type,  
            double bal)
```

```
{
```

```
    this.name = name;
```

```
    this.num = num;
```

```
    this.type = type;
```

```
    this.bal = bal;
```

double Min = 2000.00;

abstract void deposit();

abstract void withdrawal();

abstract void display();

{

class Coract extends Account

{

Coract (String name, long num, String type,
double bal)

{

super (name, num, type, bal);

{

Void withdrawal()

{

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

int amt = System.in.nextInt();

if (bal == 0 || amt > bal)

System.out.println ("Withdrawal not
possible");

else

{

bal = bal - amt;

SOP ("Amount of " + amt + " is withdrawn");

SOP ("Rem balance = " + bal);

{

}

void deposit()

{

SOP ("Enter amount");

int amt = cin.get();

bal = bal + amt;

SOP ("Remaining balance of

account = " + bal);

{

void display()

{

if (bal < min)

{

SOP ("Amt of 145/- is
deducted");

bal = bal - 145;

SOP ("Balance = " + bal);

{

else

SOP ("Balance = " + bal);

{

class Savact extends Account

{

Savact (String name, long num, String type,
double bal)

{

super (name, num, type, bal);

}

void withdrawal()

{

SOP ("Enter amount");

int amt = sc.nextInt();

if (bal == 0 || amt > bal)

{

SOP ("Withdrawal not
possible");

}

else

{

bal = bal - amt;

SOP ("Amount of " + amt + " is with-
drawn from ac")

SOP ("Rem balance is " + bal);

}

```
void deposit()
```

{

```
SOP ("Enter amt to be deposited")
```

```
int amt = s.nextInt();
```

```
SOP ("The rate of interest is 5%");
```

```
double ci = amt * (100.05);
```

```
bal = bal + ci;
```

```
SOP ("The balance = " + bal);
```

}

```
void display()
```

{

```
SOP ("Balance = " + bal);
```

}

}

```
class bank
```

{

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

{

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

```
SOP ("1 - Current account
```

```
2 - Savings account");
```

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

```
String nam;
```

```
long n;
```

```
double openbal;
```

```
if (c == 1)  
{
```

```
SOP ("Enter name, acc no and  
opening balance");
```

```
nam = s.next();
```

```
n = s.nextLong();
```

```
openbal = s.nextDouble();
```

```
Curact cur = new Curact(nam, n, openbal);
```

```
int i = 0;
```

```
while (i <= 4)
```

```
SOP (i: Deposit In 2: Display
```

```
In 3: Withdrawal 4: Exit);
```

```
SOP ("Enter choice");
```

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

```
switch(ch){
```

Case 1:

```
cu.deposit();
```

```
break;
```

Case 2:

```
cu.display();
```

```
break;
```

Case 3:

```
cu.withdrawal();
```

```
break;
```

Case 4:

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

```
break;
```

default:

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

g

t

p

```
else if (c == 2)
```

```
System.out.println("Enter name, ac no and opening  
balance");
```

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

```
ac = s.nextLong();
```

```
openingBal = s.nextDouble();
```

Savacct sa

= new Savacct (nam, n, bal, serivce,

openbal);

int j = 0;

while (j != 4)

{

SOP ("1 - Deposit In 2 - Display\n3 - Withdrawal 4 - Exit");

SOP ("Enter choice");

int ch = s.nextInt();

switch(ch){

case 1:

sa.deposit();

break;

case 2:

sa.display();

break;

case 3:

sa.withdrawal();

break;

case 4:

System.exit(0);

default: break;

SOP ("Invalid choice");

URBAN
EDGE

```
    }  
    }  
    }  
else  
{  
    System.out.println("Invalid choice");  
}  
}  
}
```

```
Bank.java > 📁 bank > ⚒ main(String[])  
1 import java.util.Scanner;  
2 abstract class Account  
3 {  
4     Scanner s=new Scanner(System.in);  
5     String name;  
6     long num;  
7     String type;  
8     double bal;  
9     Account(){  
10    Account(String name,long num,String type,double bal)  
11    {  
12        this.name=name;  
13        this.num=num;  
14        this.type=type;  
15        this.bal=bal;  
16    }  
17    String acc()  
18    {  
19        return type;  
20    }  
21    double min=2000.00;  
22    abstract void deposit();  
23    abstract void withdrawal();  
24    abstract void display();
```

```

24    abstract void display();
25 }
26 class Curact extends Account
27 {
28     Curact(String name,long num,String type,double bal)
29     {
30         super( name ,num, type, bal);
31     }
32     void withdrawal()
33     {
34         System.out.println("ENTER THE AMOUNT TO BE WITHDRAWED");
35         int amt=s.nextInt();
36         if(bal==0 || amt>bal )
37         {
38             System.out.println("WITHDRAWAL NOT POSSIBLE");
39         }
40         else
41         {
42             bal=bal-amt;
43             System.out.println("AMOUNT OF"+amt+"IS WITHDRAWN FROM THE ACCOUNT");
44             System.out.println("REMAINING BALANCE IS="+bal);
45         }
46     }
47     void deposit()
48     {

```

submitdetails.css submitdetails.js student1.html Bank.java shape.java player.java playerinheritance.java

```

46 }
47 void deposit()
48 {
49     System.out.println("ENTER THE AMOUNT TO BE DEPOSITED");
50     int amt1=s.nextInt();
51     bal=bal+amt1;
52     System.out.println("THE REMAINING BALANCE OF THE ACOOUNT= "+bal);
53 }
54 void display()
55 {
56     if (bal<min)
57     {
58         System.out.println("AMOUNT OF 145/- IS DEDUCTED FROM UR ACCOUNT DUE TO LESS BALANCE");
59         bal=bal-145;
60         System.out.println("BALANCE="+bal);
61     }
62     else
63         System.out.println("BALANCE="+bal);
64 }
65 }
66 class Savact extends Account
67 {
68     Savact (String name,long num,String type,double bal)
69     {

```

Filter (e.g. text, **/*.js, **/node_modules/)

```
Bank.java > 🐣 bank > ⚒ main(String[])
+
5     }
6
7     class Savact extends Account
8     {
9         Savact (String name,long num,String type,double bal)
10        {
11            super( name ,num, type, bal);
12        }
13        void withdrawal()
14        {
15            System.out.println("ENTER THE AMOUNT TO BE WITHDRAWED");
16            int amt=s.nextInt();
17            if(bal==0 || amt>bal)
18            {
19                System.out.println("WITHDRAWAL NOT POSSIBLE");
20            }
21            else
22            {
23                bal=bal-amt;
24                System.out.println("AMOUNT OF "+amt+" IS WITHDRAWN FROM THE ACCOUNT");
25                System.out.println("REMAINING BALANCE IS="+bal);
26            }
27        }
28        void deposit()
29        {
30
31
32
33
34
35
36
37
38 }
```

```
Bank.java > 🐣 bank > ⚒ main(String[])
+
void deposit()
{
    System.out.println("ENTER THE AMOUNT TO BE DEPOSITED");
    int amti=s.nextInt();
    System.out.println("THE RATE OF INTEREST IS 5%");
    double ci=amti*(1+0.05);
    bal=bal+ci;
    System.out.println("THE BALANCE OF THE ACCOUNT= "+bal);
}
void display()
{
    System.out.println("BALANCE="+bal);
}

class bank
{
    Run | Debug
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        System.out.println("1-CURRENT ACCOUNT \n 2-SAVINGS ACCOUNT");
        int c=s.nextInt();
        String nam;
        long n;
        double openbal;
        if(c==1)
        {
            System.out.println("ENTER NAME,ACCOUNT NUMBER AND OPENING BALANCE");
            nam=s.next();
            n=s.nextLong();
            openbal=s.nextDouble();
            System.out.println("ACCOUNT DETAILS \n *****");
            System.out.println("NAME-"+ nam +" ACC NUMBER-"+n +" OPENING BALANCE-"+openbal+" TYPE-Current");
            Curact cu=new Curact(nam,n,"current",openbal);
            int i=0;
            while( i != 4 )
        }
    }
}
```

```
Bank.java > ⚡ bank > ⏪ main(String[])
  21     int i=0;
  22     while( i != 4 )
  23     {
  24         System.out.println("1:DEPOSIT \n2:DISPLAY BALANCE\n 3:WITHDRAWAL \n 4-exit");
  25         System.out.println("ENTER THE CHOICE");
  26         int ch = s.nextInt();
  27         switch (ch) {
  28             case 1:
  29                 cu.deposit();
  30                 break;
  31
  32             case 2:
  33                 cu.display();
  34                 break;
  35
  36             case 3:
  37                 cu.withdrawal();
  38                 break;
  39             case 4:
  40                 System.exit(0);
  41                 break;
  42             default:
  43                 System.out.println("INVALID CHOICE");
  44         }
  45     }
  46
  47     else if(c==2){
  48         System.out.println("ENTER NAME,ACCOUNT NUMBER AND OPENING BALANCE");
  49         nam=s.next();
  50         n=s.nextLong();
  51         openbal=s.nextDouble();
  52         System.out.println("ACCOUNT DETAILS \n *****");
  53         System.out.println("NAME- " + nam + " ACC NUMBER-" +n + "OPENING BALANCE-"+openbal+"TYPE-Savings");
  54         Savact sa=new Savact(nam,n,"savings",openbal);
  55         int j=0;
  56         while( j != 4 )
  57         {
  58             System.out.println("1:DEPOSIT \n2:DISPLAY BALANCE\n 3:WITHDRAWAL \n 4-exit");
  59             System.out.println("ENTER THE CHOICE");
  60             int ch1 = s.nextInt();
  61             switch (ch1) {
  62                 case 1:
  63                     sa.deposit();
  64                     break;
  65
  66                 case 2:
  67                     sa.display();
  68                     break;
  69
  70                 case 3:
  71                     sa.withdrawal();
  72                     break;
  73                 case 4:
  74                     System.exit(0);
  75                     break;
  76                 default:
  77                     System.out.println("INVALID CHOICE");
  78             }
  79         }
  80     }
  81
  82     else
  83     {
  84         System.out.println("INVALID CHOICE");
  85     }
  86 }
```

```
54     Savact sa=new Savact(nam,n,"savings",openbal);
 55     int j=0;
 56     while( j != 4 )
 57     {
 58         System.out.println("1:DEPOSIT \n2:DISPLAY BALANCE\n 3:WITHDRAWAL \n 4-exit");
 59         System.out.println("ENTER THE CHOICE");
 60         int ch1 = s.nextInt();
 61         switch (ch1) {
 62             case 1:
 63                 sa.deposit();
 64                 break;
 65
 66             case 2:
 67                 sa.display();
 68                 break;
 69
 70             case 3:
 71                 sa.withdrawal();
 72                 break;
 73             case 4:
 74                 System.exit(0);
 75                 break;
 76             default:
 77                 System.out.println("INVALID CHOICE");
 78         }
 79     }
 80 }
 81
 82 else
 83 {
 84     System.out.println("INVALID CHOICE");
 85 }
```

```
c:\JAVA\bin>javac Bank.java  
c:\JAVA\bin>java bank  
1-CURRENT ACCOUNT  
2-SAVINGS ACCOUNT  
1  
ENTER NAME,ACCOUNT NUMBER AND OPENING BALANCE  
nithin  
12344  
3000  
1:DEPOSIT  
2:DISPLAY BALANCE  
3:WITHDRAWAL  
4-exit  
ENTER THE CHOICE  
1  
ENTER THE AMOUNT TO BE DEPOSITED  
2000  
THE REMAINING BALANCE OF THE ACOOUNT= 5000.0  
1:DEPOSIT  
2:DISPLAY BALANCE  
3:WITHDRAWAL  
4-exit  
ENTER THE CHOICE  
3  
ENTER THE AMOUNT TO BE WITHDRAWED  
2000  
AMOUNT OF 2000 IS WITHDRAWN FROM THE ACCOUNT  
REMAINING BALANCE IS=3000.0  
1:DEPOSIT
```

```
> 3000  
1:DEPOSIT  
2:DISPLAY BALANCE  
3:WITHDRAWAL  
4-exit  
ENTER THE CHOICE  
3  
ENTER THE AMOUNT TO BE WITHDRAWED  
2000  
AMOUNT OF 2000 IS WITHDRAWN FROM THE ACCOUNT  
REMAINING BALANCE IS=1000.0  
1:DEPOSIT  
2:DISPLAY BALANCE  
3:WITHDRAWAL  
4-exit  
ENTER THE CHOICE  
2  
AMOUNT OF 145/- IS DEDUCTED FROM UR ACCOUNT DUE TO LESS BALANCE  
BALANCE=855.0  
1:DEPOSIT  
2:DISPLAY BALANCE  
3:WITHDRAWAL  
4-exit  
ENTER THE CHOICE  
4
```

LAB PROGRAM-6

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- 6 program

Here package CIE has been created which has two class files student and Internals. Internals is a derived class of student. Now another package has been created with name SEE and a class file named External has been created. External is a sub class of Student from CIE package. Now in file which has driver class CIE & SEE package has been imported and an object of Internals and External has been created. The total marks is then calculated from members of these two classes.

Student.java

```
package CIE;
public class student {
    public String vrn;
    public String name;
    public int sem;
    public student(){}
    public student(String vrn, String name, int sem)
    {
        this.vrn = vrn;
        this.name = name;
        this.sem = sem;
    }
}
```

Internals.java

URBAN
EDGE

```
package CIE;  
import java.util.Scanner;  
public class Internals extends Student {  
    Scanner s = new Scanner (System.in);  
    public int [] cie = new int [5];  
    public void get()  
    {  
        for (int i=0; i<5; i++)  
            System.out.println("Enter CIE  
marks in subject  
+ (PHI);");  
        cie[i] = s.nextInt();  
    }  
}
```

Externals.java

```
package SEE;  
import java.util.Scanner;  
public class Externals extends CIE.Student {  
    public Externals (String usn, String name,  
                    int sem)  
    {  
        super (usn, name, sem);  
        Scanner s = new Scanner (System.in);  
        public int [] See = new int [5];  
        public void get()  
        {
```

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

URBAN
EDGE

SOP("Enter score in Subject " + i);

see [i] = s.nextInt();

f
f
f

final marks.java

```
import CIE.*;
```

```
import SEE.*;
```

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

```
class final Marks {
```

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

{

Scanner s = new Scanner (System.in);

SOP("Enter no of students");

int n = s.nextInt();

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

[n];

CIE.Internal ob[] = new CIE.Internal[n];

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

{

SOP("Enter vrn, name, sem of student " + i);

String u = s.nextLine();

String na = s.nextLine();

int se = s.nextInt();

ob[i] = new CIE.Internal();

ob[i].get();

ob[i] = new SEE.External(u, na, se);

ob[i].get();

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

{

SOP("Name") + ob[i].name + " &" +
ob[i].mark + " &" +
ob[i].sem);

SOP("The details mark of student(" +
"are")");

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

{

SOP("Subject" + (j+1) + "=" + ob[j])

(ob[j].get(j) + ob[i].sem[j]/2))

```
Usage: javac <options> <source files>
use -help for a list of possible options

c:\JAVA\bin>javac CIE/Internals.java

c:\JAVA\bin>javac CIE/Student.java

c:\JAVA\bin>javac SEE/Externals.java

c:\JAVA\bin>javac finalmarks.java

c:\JAVA\bin>java finalmarks
ENTER THE NUMBER OF STUDENTS
2
ENTER THE USN,NAME AND SEMESTER OF STUDENT1
123
nithin
2
ENTER THE CIE MARK IN SUBJECT(OUT OF 50)1
34
ENTER THE CIE MARK IN SUBJECT(OUT OF 50)2
23
ENTER THE CIE MARK IN SUBJECT(OUT OF 50)3
34
ENTER THE CIE MARK IN SUBJECT(OUT OF 50)4
23
ENTER THE CIE MARK IN SUBJECT(OUT OF 50)5
34
ENTER THE SEE MARK IN SUBJECT(OUT OF 100)1
98
```

```
98
ENTER THE SEE MARK IN SUBJECT(OUT OF 100)4
78
ENTER THE SEE MARK IN SUBJECT(OUT OF 100)5
78
ENTER THE USN,NAME AND SEMESTER OF STUDENT2
1234
gd
4
ENTER THE CIE MARK IN SUBJECT(OUT OF 50)1
45
ENTER THE CIE MARK IN SUBJECT(OUT OF 50)2
45
ENTER THE CIE MARK IN SUBJECT(OUT OF 50)3
45
ENTER THE CIE MARK IN SUBJECT(OUT OF 50)4
45
ENTER THE CIE MARK IN SUBJECT(OUT OF 50)5
45
ENTER THE SEE MARK IN SUBJECT(OUT OF 100)1
89
ENTER THE SEE MARK IN SUBJECT(OUT OF 100)2
78
ENTER THE SEE MARK IN SUBJECT(OUT OF 100)3
98
ENTER THE SEE MARK IN SUBJECT(OUT OF 100)4
78
ENTER THE SEE MARK IN SUBJECT(OUT OF 100)5
78
NAME:nithinUSN:123SEMESTER:2
```

```
98  
ENTER THE SEE MARK IN SUBJECT(OUT OF 100)4  
78  
ENTER THE SEE MARK IN SUBJECT(OUT OF 100)5  
78  
NAME:nithinUSN:123SEMESTER:2  
THE TOTAL MARKS OF STUDENT1 ARE  
SUBJECT1=83  
SUBJECT2=62  
SUBJECT3=83  
SUBJECT4=62  
SUBJECT5=73  
NAME:gdUSN:1234SEMESTER:4  
THE TOTAL MARKS OF STUDENT2 ARE  
SUBJECT1=89  
SUBJECT2=84  
SUBJECT3=94  
SUBJECT4=84  
SUBJECT5=84  
c:\JAVA\bin>
```

LAB PROGRAM-7

Write a program to demonstrate generics with multiple object parameters.

Lab program - 7

```
import java.util.Scanner;  
  
class generic<DT1, DT2>  
{  
    DT1 obj1;  
    DT2 obj2;  
    DT3 obj3;  
    generic(DT1 a, DT2 b, DT3 c)  
    {  
        obj1 = a;  
        obj2 = b;  
        obj3 = c;  
    }  
    DT1 get1()  
    {  
        return obj1;  
    }  
    DT2 get2()  
    {  
        return obj2;  
    }  
    DT3 get3()  
    {  
        return obj3;  
    }  
    void showdatatype()  
    {  
        System.out.println("The type of Datatype used is = "+  
                           obj1.getClass().getName());  
        System.out.println("The type of Datatype used is = "+obj2.getClass().  
                           getName());  
        System.out.println("The type of Datatype used is = "+obj3.getClass().  
                           getName());  
    }  
}
```

```
class genericMain
{
    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter the values");
        int x = s.nextInt();
        String str = s.next();
        double xx = s.nextDouble();
        generic < Integer, String, Double> a
            = new generic < Integer, String, Double> (x, str, xx);
        a.showdatatype();
        System.out.println ("The Integer entered is = " + a.get1());
        System.out.println ("The String entered is = " + a.get2());
        System.out.println ("The Double value entered is = " + a.get3());
    }
}
```

```
Student.java Internals.java Externals.java finalmarks.java gen.java exp.java
gen.java > genericmain > main(String[])
1 import java.util.Scanner;
2 class generic<DT1,DT2,DT3>
3 {
4     DT1 obj;
5     DT2 obj1;
6     DT3 obj2;
7     generic(DT1 a,DT2 b,DT3 c)
8     {
9         obj=a;
10        obj1=b;
11        obj2=c;
12    }
13    DT1 get1()
14    {
15        return obj;
16    }
17    DT2 get2()
18    {
19        return obj1;
20    }
21    DT3 get3()
22    {
23        return obj2;
24    }
25    void showdatatype()
26    {
27        System.out.println("THE TYPES OF DATATYPE USED IS="+obj.getClass().getName());
28        System.out.println("THE TYPES OF DATATYPE USED IS="+obj1.getClass().getName());
29        System.out.println("THE TYPES OF DATATYPE USED IS="+obj2.getClass().getName());
30    }
31 }
32 class genericmain
33 {
34     Run | Debug
35     public static void main(String args[])
36     {
37
38
39
40
41
42
43
44
45
46
47
48 }
```

```
Student.java Internals.java Externals.java finalmarks.java gen.java exp.java
gen.java > genericmain > main(String[])
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48 }
```

```
THE AGES ARE ENTERED CORRECTLY
FATHER'S AGE=56 SON'S AGE=34

c:\JAVA\bin>javac gen.java

c:\JAVA\bin>java genericmain
ENTER THE VALUES
23
nithin
56.244
THE TYPES OF DATATYPE USED IS=java.lang.Integer
THE TYPES OF DATATYPE USED IS=java.lang.String
THE TYPES OF DATATYPE USED IS=java.lang.Double
THE INTEGER ENTERED IS=23
THE STRING ENTERED IS=nithin
THE DOUBLE VALUE ENTERED IS=56.244

c:\JAVA\bin>
```

LAB PROGRAM-8

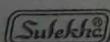
Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception Wrong Age() when the input age<0. In Son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

Name of the Experiment :
Experiment No.

: WEEK-#0 LAB-8 NEW

Date :
Page No.:

```
import java.util.Scanner;  
class WrongAge extends Exception {  
    int age;  
    WrongAge(int x) {  
        age = x;  
    } public String toString() {  
        return "AGE OF FATHER " + age + " INCORRECT";  
    }  
}  
class WrongAgeSon extends Exception {  
    int age;  
    WrongAgeSon(int x) {  
        age = x;  
    } public String toString() {  
        return "AGE OF SON " + age + " INCORRECT";  
    }  
}  
class Father {  
    int a;  
    Father(int x) {  
        a = x;  
    } void check() throws WrongAge {  
        if (a < 0) {  
            throw new WrongAge(a);  
        }  
    }  
}
```



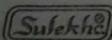
King Size Practical Book

Signature.....

Name of the Experiment :
Experiment No. :

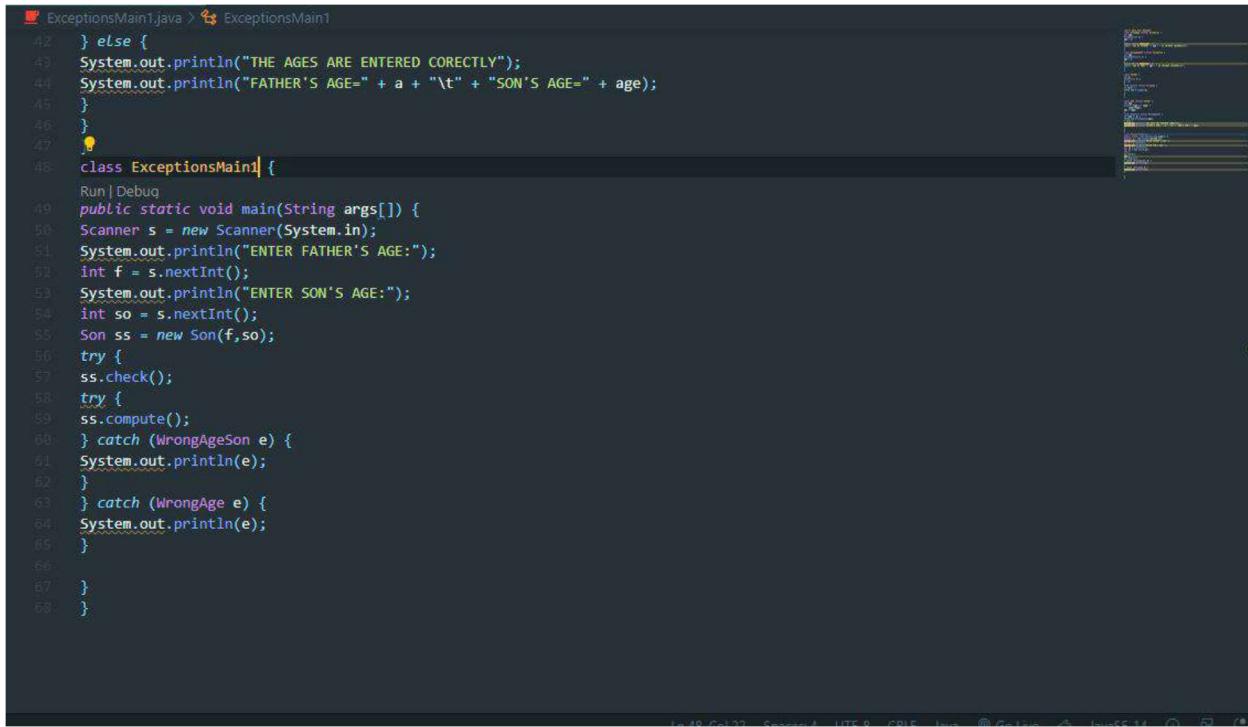
Date :
Page No. :

```
class Son extends Father {  
    int age;  
    Son(int sage, int sage) {  
        super(sage);  
        age = sage;  
    } void compute() throws WrongAgeSon {  
        if (age <= 0) {  
            throw new WrongAgeSon(age);  
        } else {  
            System.out.println("CORRECT" + sage + age " " + age);  
        }  
    }  
    class ExceptionMain {  
        public static void main(String[] args) {  
            Scanner s = new Scanner(System.in);  
            System.out.println("Enter father's and son's ages");  
            int f = s.nextInt();  
            int so = s.nextInt();  
            Son ss = new Son(f, so);  
            try { ss.check(); }  
            try { ss.compute(); }  
            catch (WrongAgeSon e) {  
                System.out.println(e);  
            } catch (WrongAge e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```



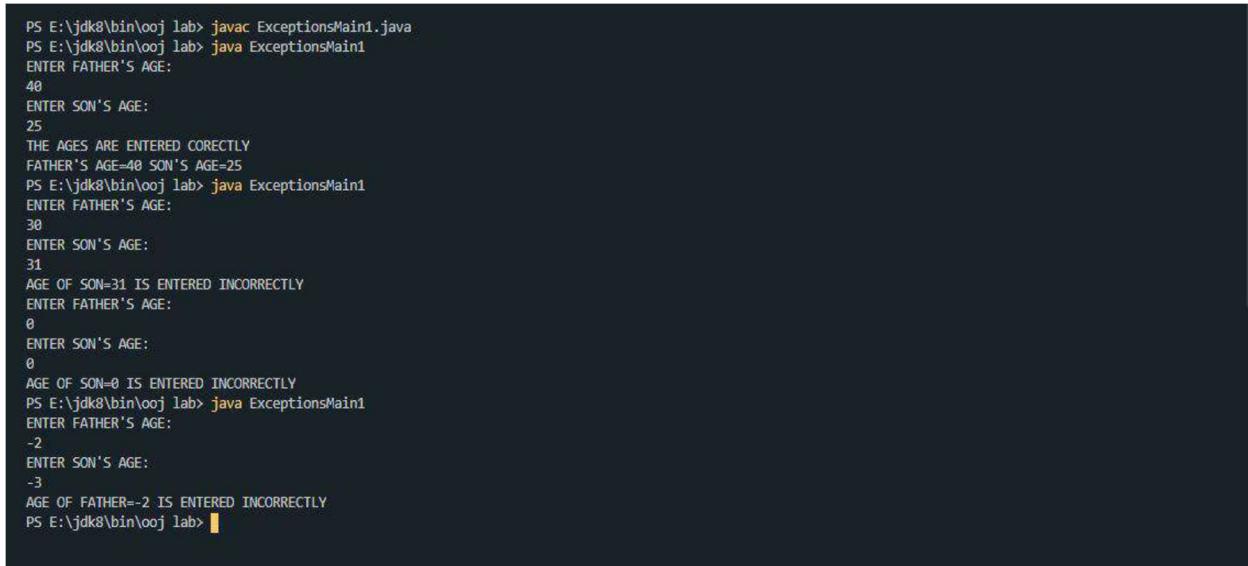
```
ExceptionsMain1.java > ExceptionsMain1
1 import java.util.Scanner;
2 class WrongAge extends Exception {
3     int age;
4     WrongAge(int x) {
5         age = x;
6     }
7     public String toString() {
8         return "AGE OF FATHER=" + age + " IS ENTERED INCORRECTLY";
9     }
10 }
11 class WrongAgeSon extends Exception {
12     int age;
13     WrongAgeSon(int x) {
14         age = x;
15     }
16     public String toString() {
17         return "AGE OF SON=" + age + " IS ENTERED INCORRECTLY";
18     }
19 }
20
21 class Father {
22     int a;
23     Father(int x) {
24         a = x;
25     }
26     void check() throws WrongAge {
27         if (a<0) {
28             throw new WrongAge(a);
29         }
30     }
31 }
32
33 class Son extends Father {
34     int age;
35     Son(int fage,int sage) {
36         super(fage);
37         age = sage;
38     }
39     void compute() throws WrongAgeSon {
40         if (age >= a) {
41             throw new WrongAgeSon(age);
42         } else {
43             System.out.println("THE AGES ARE ENTERED CORECTLY");
44             System.out.println("FATHER'S AGE=" + a + "\t" + "SON'S AGE=" + age);
45         }
46     }
47
48 class ExceptionsMain1 {
49     Run | Debug
50     public static void main(String args[]) {
51         Scanner s = new Scanner(System.in);
52         System.out.println("ENTER FATHER'S AGE:");
53         int f = s.nextInt();
54         System.out.println("ENTER SON'S AGE:");
55         int so = s.nextInt();
56         Son ss = new Son(f,so);
57         try {
58             ss.check();
59         } catch (WrongAge e) {
60             System.out.println(e.toString());
61         }
62         try {
63             ss.compute();
64         } catch (WrongAgeSon e) {
65             System.out.println(e.toString());
66         }
67     }
68 }
```

```
ExceptionsMain1.java > ExceptionsMain1
28         throw new WrongAge(a);
29     }
30 }
31 }
32
33 class Son extends Father {
34     int age;
35     Son(int fage,int sage) {
36         super(fage);
37         age = sage;
38     }
39     void compute() throws WrongAgeSon {
40         if (age >= a) {
41             throw new WrongAgeSon(age);
42         } else {
43             System.out.println("THE AGES ARE ENTERED CORECTLY");
44             System.out.println("FATHER'S AGE=" + a + "\t" + "SON'S AGE=" + age);
45         }
46     }
47
48 class ExceptionsMain1 {
49     Run | Debug
50     public static void main(String args[]) {
51         Scanner s = new Scanner(System.in);
52         System.out.println("ENTER FATHER'S AGE:");
53         int f = s.nextInt();
54         System.out.println("ENTER SON'S AGE:");
55         int so = s.nextInt();
56         Son ss = new Son(f,so);
57         try {
58             ss.check();
59         } catch (WrongAge e) {
60             System.out.println(e.toString());
61         }
62         try {
63             ss.compute();
64         } catch (WrongAgeSon e) {
65             System.out.println(e.toString());
66         }
67     }
68 }
```



A screenshot of an IDE showing Java code. The code defines a class `ExceptionsMain1` with a `main` method. It uses `Scanner` to read ages from the user and `System.out` to print messages. It includes exception handling for `WrongAgeSon` and `WrongAge` exceptions.

```
112 } else {
113     System.out.println("THE AGES ARE ENTERED CORECTLY");
114     System.out.println("FATHER'S AGE=" + a + "\t" + "SON'S AGE=" + age);
115 }
116 }
117
118 class ExceptionsMain1 {
119     Run | Debug
120     public static void main(String args[]) {
121         Scanner s = new Scanner(System.in);
122         System.out.println("ENTER FATHER'S AGE:");
123         int f = s.nextInt();
124         System.out.println("ENTER SON'S AGE:");
125         int so = s.nextInt();
126         Son ss = new Son(f,so);
127         try {
128             ss.check();
129             try {
130                 ss.compute();
131             } catch (WrongAgeSon e) {
132                 System.out.println(e);
133             }
134             } catch (WrongAge e) {
135                 System.out.println(e);
136             }
137         }
138     }
139 }
```



A screenshot of a terminal window showing the execution of the Java program. It runs `javac ExceptionsMain1.java` and then `java ExceptionsMain1`. The user enters ages for father and son, and the program handles various error cases like negative ages and zero ages.

```
PS E:\jdk8\bin\oobj lab> javac ExceptionsMain1.java
PS E:\jdk8\bin\oobj lab> java ExceptionsMain1
ENTER FATHER'S AGE:
40
ENTER SON'S AGE:
25
THE AGES ARE ENTERED CORECTLY
FATHER'S AGE=40 SON'S AGE=25
PS E:\jdk8\bin\oobj lab> java ExceptionsMain1
ENTER FATHER'S AGE:
30
ENTER SON'S AGE:
31
AGE OF SON=31 IS ENTERED INCORRECTLY
ENTER FATHER'S AGE:
0
ENTER SON'S AGE:
0
AGE OF SON=0 IS ENTERED INCORRECTLY
PS E:\jdk8\bin\oobj lab> java ExceptionsMain1
ENTER FATHER'S AGE:
-2
ENTER SON'S AGE:
-3
AGE OF FATHER=-2 IS ENTERED INCORRECTLY
PS E:\jdk8\bin\oobj lab>
```

LAB PROGRAM-9

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.

Lab program - 9

```
class Thread1 implements Runnable {  
    String name;  
    Thread t;  
    int time;  
    Thread1 (String threadname, int time) {  
        name = threadname;  
        this.name = name;  
        t = new Thread (this, name);  
        System.out.println ("Thread:" + t);  
        t.start();  
    }  
    public void run() {  
        try {  
            for (int i = 5; i > 0; i--) {  
                System.out.println (name);  
                Thread.sleep (time);  
            }  
        } catch (InterruptedException e) {  
            System.out.println (name + " interrupted");  
        }  
        System.out.println (name + " exiting");  
    }  
}
```

class ThreadMain

{

public static void main (String
args [])

{

Thread t1 = new Thread ("BMS College
of engineering", 10000);

Thread t2 = new Thread ("Computer Science
and engineering", 2000);

{}

terminal Help

multithreading.java - PROJECT-1 - Visual Studio Co

```
# app.css      ◊ appointment.html      # appointment.css      JS app.js      JS
C: > JAVA > bin > multithreading.java > Thread1 > run()
1 class Thread1 implements Runnable {
2     String name;
3     Thread t;
4     int time;
5     Thread1(String threadname,int time) {
6         name = threadname;
7         this.time=time;
8         t = new Thread(this, name);
9         System.out.println("thread:"+ t);
10        t.start();
11    }
12    public void run() {
13        try {
14            for(int i = 5; i > 0; i--) {
15                System.out.println(name);
16                Thread.sleep(time);
17            }
18        } catch (InterruptedException e) {
19            System.out.println(name + " Interrupted");
20        }
21        System.out.println(name + " exiting.");
22    }
23}
24 class threadmain {
Run | Debug
25     public static void main(String args[]) {
26         Thread1 t1=new Thread1("BMS COLLEGE OF ENGINEERING",10000);
27         Thread1 t2=new Thread1("COMPUTER SCIENCE AND ENGINEERING",2000);
28     }
29 }
```

```
innable  Command Prompt
SUM OF EVEN NUMBERS=2550
ODD THREAD quitting
SUM OF ODD NUMBERS=2500
e,int
c:\JAVA\bin>javac multithreading.java

e);
ad:"+
c:\JAVA\bin>java threadmain
thread:Thread[BMS COLLEGE OF ENGINEERING,5,main]
thread:Thread[COMPUTER SCIENCE AND ENGINEERING,5,main]
BMS COLLEGE OF ENGINEERING
COMPUTER SCIENCE AND ENGINEERING
BMS COLLEGE OF ENGINEERING
COMPUTER SCIENCE AND ENGINEERING exiting.
BMS COLLEGE OF ENGINEERING
BMS COLLEGE OF ENGINEERING
BMS COLLEGE OF ENGINEERING
BMS COLLEGE OF ENGINEERING exiting.

+ "Int
+ " ex
c:\JAVA\bin>

(string
"BMS CO
```

LAB PROGRAM-10

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.

```
import java.awt.*;  
import java.awt.event.*;  
import javax.swing.*;  
  
public class Integerdivision extends Frame  
implements ActionListener {  
    JTextField n1, n2, res;  
    Label ln1, ln2, lres;  
    Button b;  
    public Integerdivision () {  
        setLayout (new FlowLayout ());  
        Label ln1 = new Label ("Number 1",  
                             Label.RIGHT);  
        Label ln2 = new Label ("Number  
2", Label.RIGHT);  
        Label lres = new Label ("Result", Label.  
                             RIGHT);  
        n1 = new JTextField (12);  
        n2 = new JTextField (8);  
        res = new JTextField (10);  
        b = new Button ("Divide");
```

```
odd(ln1);  
odd(n1);  
odd(ln2);  
odd(n2);  
odd(b);  
odd(lres);  
odd(rres);
```

b. addActionListener (new Window
-Adapter());

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

```
{ if (ae.getSource() == b)  
try {  
int num1 = Integer.parseInt (n1.get  
Text());  
int num2 = Integer.parseInt (n2.get  
Text());  
int num3 = num1 / num2;  
res.setText (String.valueOf (num3));  
}
```

```
catch (NumberFormatException ne)
```

```
{
```

```
JOptionPane.showMessageDialog(this, ne,  
"ERROR", JOptionPane.ERROR_MESSAGE);
```

```
}
```

```
catch (ArithmaticException a) {
```

```
JOptionPane.showMessageDialog(this, a,  
"ERROR", JOptionPane.ERROR_MESSAGE);
```

```
}
```

```
}
```

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

```
{
```

```
IntegerDivision i = new IntegerDivision();
```

```
i.setSize (new Dimension (400, 400));
```

```
i.setTitle ("INTEGER DIVISION OF  
TWO NUMBERS");
```

```
i.setVisible (true);
```

```
}
```

```
class WindowAdapter extends Window
```

```
Adapter {
```

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

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

```
}
```

File Terminal Help

integerdivision.java - bin - Visual Studio Code

integerdivision.java > integerdivision > integerdivision()

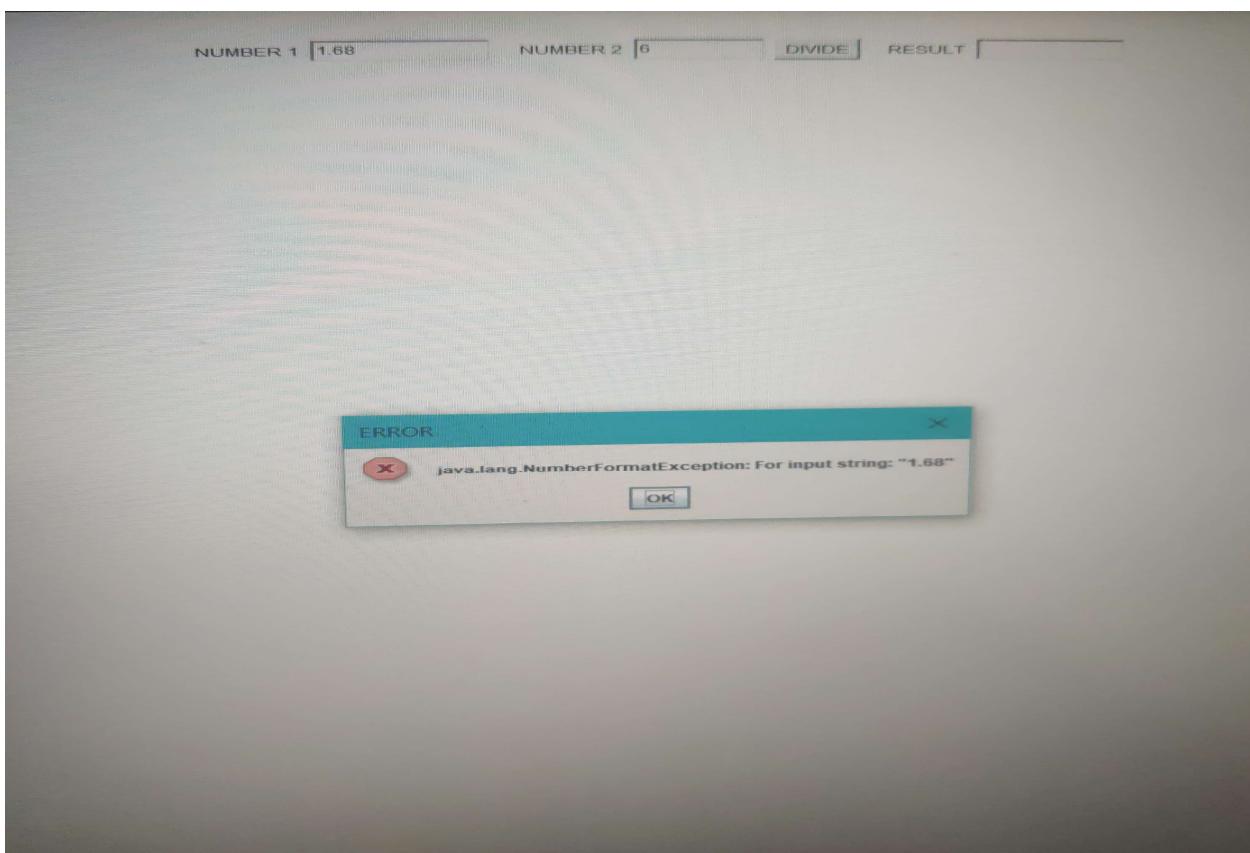
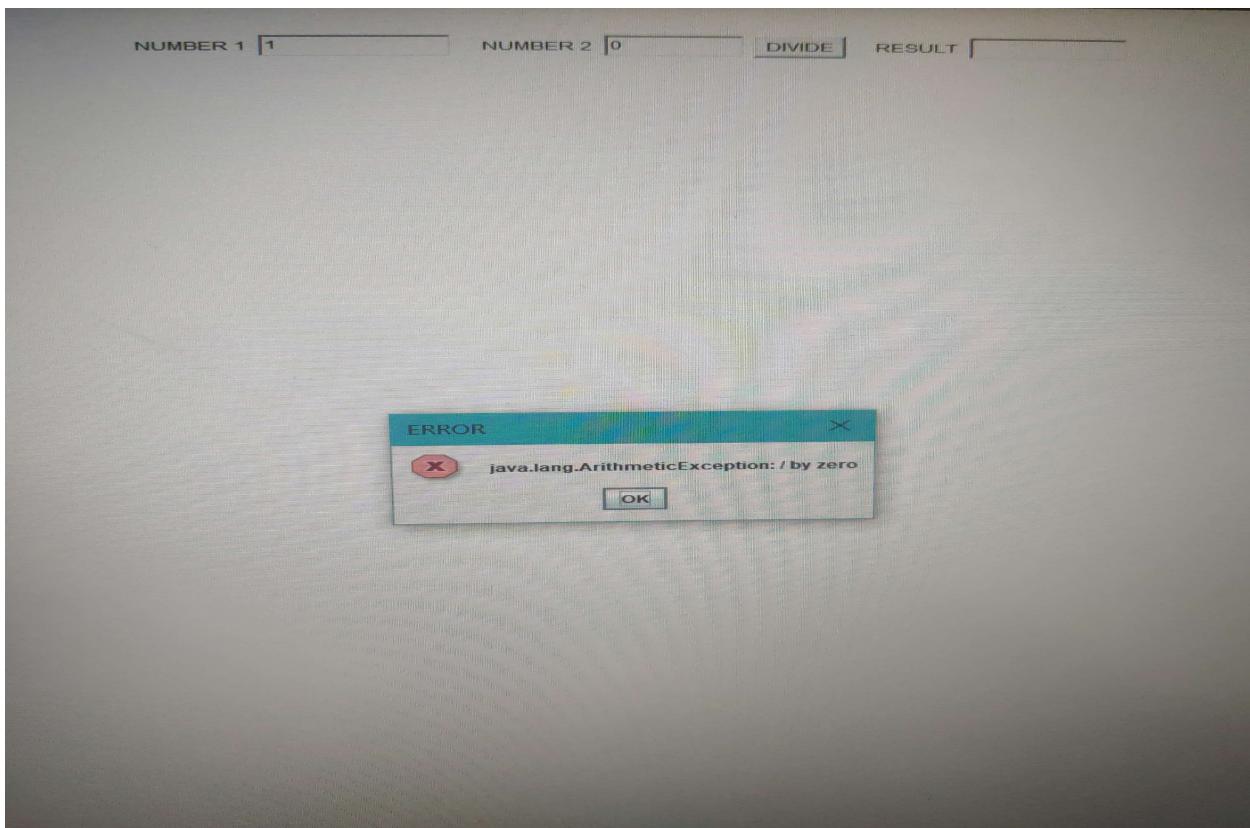
```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4 public class integerdivision extends Frame implements ActionListener{
5     String msg="";
6     TextField n1,n2,res;
7     Label ln1,ln2,lres;
8     Button b;
9     public integerdivision(){}
10    setLayout(new FlowLayout());
11    Label ln1=new Label("NUMBER 1",Label.RIGHT);
12    Label ln2=new Label("NUMBER 2",Label.RIGHT);
13    Label lres=new Label("RESULT",Label.RIGHT);
14    n1=new TextField(12);
15    n2=new TextField(8);
16    res=new TextField(10);
17    b=new Button("DIVIDE");
18    add(ln1);
19    add(n1);
20    add(ln2);
21    add(n2);
22    add(b);
23    add(lres);
24    add(res);
25    b.addActionListener(this);
26    addWindowListener(new WindowAdapter1());
27 }
28 public void actionPerformed(ActionEvent ae)
29 {
30     if(ae.getSource()==b)
31     {
32         try{
33             int num1=Integer.parseInt(n1.getText());
34             int num2=Integer.parseInt(n2.getText());
35             int num3=num1/num2;
36             res.setText(String.valueOf(num3));
37         }catch(NumberFormatException ne ){
38             JOptionPane.showMessageDialog(this, "ERROR", JOptionPane.ERROR_MESSAGE);
39         }
40     }
41 }
```

ninal Help integerdivision.java - bin - Visual Studio Code

java exp2.java abstract1.java queue.java gradandundergrad.java button.java

integerdivision.java > integerdivision > integerdivision()

```
34     int num2=Integer.parseInt(n2.getText());
35     int num3=num1/num2;
36     res.setText(String.valueOf(num3));
37 }catch(NumberFormatException ne ){
38     JOptionPane.showMessageDialog(this,ne,"ERROR", JOptionPane.ERROR_MESSAGE);
39     /* msg="NUMBERFORMAT EXCEPTION";
40     repaint();*/
41 }
42 catch(ArithmetricException a){
43     JOptionPane.showMessageDialog(this,a,"ERROR", JOptionPane.ERROR_MESSAGE);
44     /*msg="ARITHMETIC EXCEPTION";
45     repaint();*/
46 }
47 }
48 }
49 /*public void paint(Graphics g)
50 {
51     g.drawString(msg,200,200);
52 }*/
Run | Debug
53 public static void main(String args[])
54 {
55     integerdivision i=new integerdivision();
56     i.setSize(new Dimension(300,300));
57     i.setTitle("INTEGER DIVISION OF TWO NUMBERS");
58     i.setVisible(true);
59 }
60
61 class WindowAdapter1 extends WindowAdapter{
62     public void windowClosing(WindowEvent we)
63     {
64         System.exit(0);
65     }
66 }
67 }
```



NUMBER 1

NUMBER 2

DIVIDE

RESULT