

Week 4 Lab Program 1:

Q. 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 discriminate $b^2 - 4ac$ is negative, display a message stating that there are no real solution

Observation copy:

Date : 29 September 2020 Name : Shrikant Singh, USN : 1 BM19 CS 063

OOJ Lab Exercise - 1 :-

- Q 1. Develop a Java Program that prints all real solutions to the quadratic equations $ax^2 + bx + c = 0$. 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.

Ans. Input :-

```
import java.util.*;  
  
class quadraticRoot {  
    public static void main (String args[]) {  
        double a, b, c, d;  
        double root1, root2;  
        Scanner scr = new Scanner (System.in);  
        System.out.println ("Enter the value of a, b, c :");  
        a = scr.nextDouble();  
        b = scr.nextDouble();  
        c = scr.nextDouble();  
        d = (b * b) - (4 * a * c);  
        if (d > 0)  
            root1 = (-b + Math.sqrt (d)) / (2 * a);  
            root2 = (-b - Math.sqrt (d)) / (2 * a);  
            System.out.println ("Roots are positive and unequal  
            1st Root = " + root1 + " and  
            2nd Root = " + root2);  
        }  
        else if (d == 0)
```

Date :- 29 September 2020 Name :- Shridik Singh , USN :- 1BM19CS063

$$\text{root 1} = (-b + \text{Math.sqrt}(d)) / (2 * a);$$

System.out.println (" Roots are positive and equal , 1st Root = $\frac{-b + \sqrt{d}}{2a}$
= 2nd Root ");

}

else

{

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

?

{

}

Output of the above code :-

Enter the values of a, b, c :

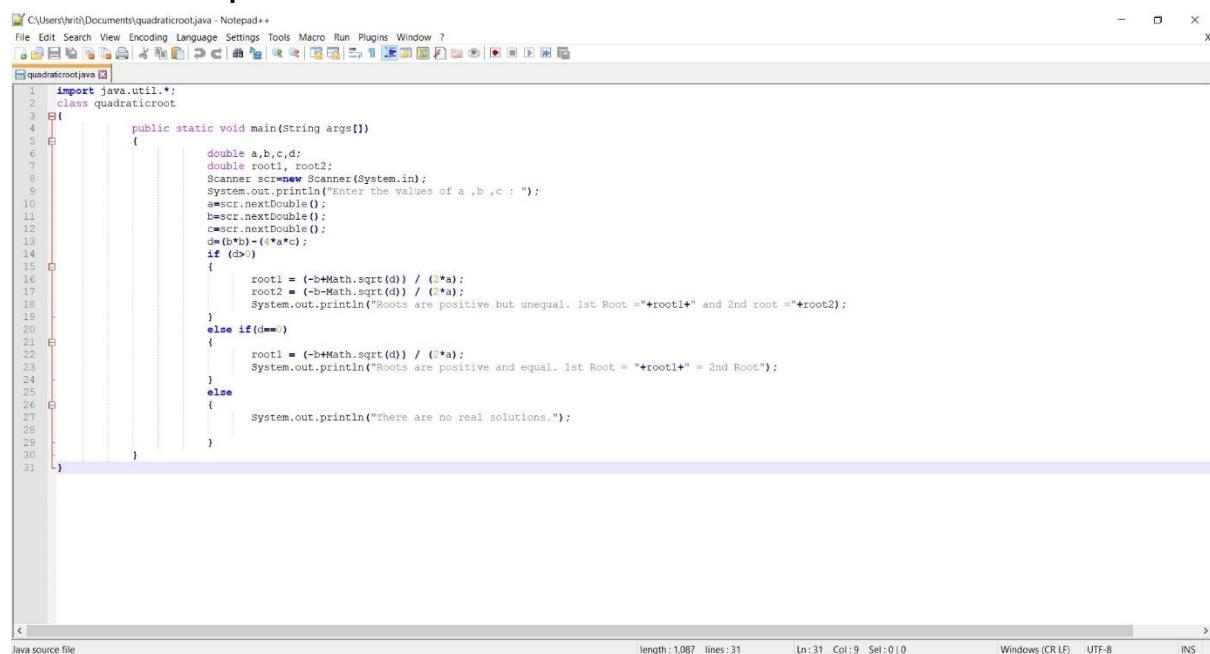
1

2

-3

Roots are positive but unequal , 1st Root = 1.0 and
2nd Root = -3.0 .

Screen Output:



The screenshot shows a Notepad++ window with the file 'quadraticroot.java' open. The code implements a Java program to calculate the roots of a quadratic equation. It uses the Math.sqrt() method to find square roots and conditional statements to handle different cases based on the discriminant (d). The code includes imports for java.util.* and java.lang.*.

```
1 import java.util.*;
2 class quadraticroot
3 {
4     public static void main(String args[])
5     {
6         double a,b,c,d;
7         double root1,root2;
8         Scanner scr=new Scanner(System.in);
9         System.out.println("Enter the values of a ,b ,c : ");
10        a=scr.nextDouble();
11        b=scr.nextDouble();
12        c=scr.nextDouble();
13        d=(b*b)-(4*a*c);
14        if (d>0)
15        {
16            root1 = (-b+Math.sqrt(d)) / (2*a);
17            root2 = (-b-Math.sqrt(d)) / (2*a);
18            System.out.println("Roots are positive but unequal. 1st Root =" +root1+ " and 2nd root =" +root2);
19        }
20        else if(d==0)
21        {
22            root1 = (-b+Math.sqrt(d)) / (2*a);
23            System.out.println("Roots are positive and equal. 1st Root = " +root1+ " = 2nd Root");
24        }
25        else
26        {
27            System.out.println("There are no real solutions.");
28        }
29    }
30 }
```

Week 4 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.

Observation copy:

Date: 6 October 2020, Name: Shrikant Singh, USN: 1BM19CS063.

OOJ Lab Program 2:-

Input :-

```

import java.util.*;
class Student
{
    String usn, name;
    static int credits[];
    static double marks[];
    void input (int n)
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter USN and Name");
        usn = sc.nextLine ();
        name = sc.nextLine ();
        System.out.println ("Enter marks along with credits");
        for (int i=0; i<n; i++)
        {
            marks [i] = sc.nextDouble();
            credits [i] = sc.nextInt();
        }
    }
    double calculate (int n)
    {
        int c, cred = 0;
        double tot, total = 0.0;
        for (int i=0; i<n; i++)
        {
            tot = marks [i];
            if (tot >= 90)
                c = 10;
            else if (tot >= 80)

```

Date : 6 October 2020, Name : Sritik Singh, USN : 1BM19CS063

```

c = 9;
else if (tot >= 70)
    c = 8;
else if (tot >= 60)
    c = 7;
else if (tot >= 50)
    c = 6;
else if (tot >= 40)
    c = 4;
else
    c = 0;

total = total + (c * credits[i]);
cred = cred + credits[i];

}

total = total / cred;
return (total);
}

void display (int n, double total)
{
    System.out.println ("Enter the name of student : " + name);
    System.out.println ("USN of Student : " + usn);

    System.out.println ("marks of student along with credits of course");
    for (int i = 0; i < n; i++)
    {
        System.out.print (marks[i] + " " + credits[i]);
    }

    System.out.println ("syra of student : " + total);
}

public static void main (String args[])
{
    Scanner sc = new Scanner (System.in);
    Student obj = new Student ();
    System.out.print ("Enter the no. of course : ");
}

```


Date : 6 October 2020, Name : Hritik Singh, USN : 1BM19CS063.

```
int n = sc.nextInt();
credit = new int[n];
marks = new double[n];
obj. input(n);
double total = obj.calculate(n);
obj. display(n, total);
}
```

Output:-

Enter USN and Name

1BM19CS063 Hritik

Enter marks along with credit:

89	5
79	7
78	6
89	8
78	6

Name of student : Hritik

USN of student : 1BM19CS063

Marks of student along with credits

89.0	5
79.0	7
78.0	6
89.0	8
78.0	6

sgpa of student = 8.40625

Screen Output:

The screenshot shows a Windows desktop environment. In the foreground, there is a Notepad++ window titled "Student.java" containing Java code. In the background, there is a Command Prompt window showing the execution of the program and its output.

```
1 import java.util.*;
2 class student
3 {
4     String usn,name;
5     static int credits[];
6     static double marks[];
7     void input(int n)
8     {
9         Scanner sc=new Scanner(System.in);
10        System.out.println("Enter usn and name");
11        usn=sc.nextLine();
12        name=sc.nextLine();
13        System.out.println("Enter marks along with credits");
14        for(int i=0;i<n;i++)
15        {
16            marks[i]=sc.nextDouble();
17            credits[i]=sc.nextInt();
18            System.out.println();
19        }
20    }
21    double calculate(int n)
22    {
23        int c,cred=0;
24        double tot,total=0.0;
25        for(int i=0;i<n;i++)
26        {
27            tot=marks[i];
28            if(tot>=90)
29            {
30                cred=10;
31            }
32            else if(tot>=80)
33            {
34                cred=9;
35            }
36            else if(tot>=70)
37            {
38                cred=8;
39            }
40            else
41            {
42                total+=tot+(c*credits[i]);
43                cred+=cred+credits[i];
44            }
45        }
46    }
47    public static void main(String args[])
48    {
49        student s=new student();
50        s.input(5);
51        double result=s.calculate(5);
52        System.out.println("Name of student : "+name);
53        System.out.println("USN of student : "+usn);
54        System.out.println("Marks of student along with credits of course");
55        System.out.println("99.0 5");
56        System.out.println("79.0 7");
57        System.out.println("78.0 6");
58        System.out.println("89.0 8");
59        System.out.println("78.0 6");
60        System.out.println("SGPA of student : "+result);
61    }
62}
```

Command Prompt Output:

```
Enter usn and name
18M19CSE03 Hritik
Enter marks along with credits
89
5
79
7
78
6
89
8
78
6
Name of student :
USN of student : 18M19CSE03 Hritik
Marks of student along with credits of course
99.0 5
79.0 7
78.0 6
89.0 8
78.0 6
SGPA of student : 8.40625
```

Week 5 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.

Observation copy :

Page no.: ①

Date: 13 October 2020, Name: Shrikant Singh, USN: 1BM19CS063

OOJ Lab Program Week 5, Lab 3:

Lab - Program 3:-

Input:-

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

```
class Book {
```

```
    private String Bookname;  
    private String Authorname;  
    private double Bookprice;  
    private int number of pages;
```

Book()

```
{
```

System.out.println ("Name of book, Author of book, Price of book and Number of Pages of the book respectively inside Default constructor.");

Bookname = "Invisible Man";

Authorname = "H.G.Wells";

Bookprice = 30.67;

number of pages = 800;

Book(String Bookname, String Authorname, double Bookprice, int number of pages)

System.out.println ("Name of book, Author of book, Price of book and Number of Pages in the book respectively inside Parameterized constructor.");

Bookname = Bookname;

Authorname = Authorname;

Bookprice = Bookprice;

number of pages = number of pages;

```
}
```

Date: 13 October 2020, Name: Nitish Singh, USN: 1 BM19CS062

void getdetails ()

```
{  
Scanner sc = new Scanner (System.in);  
System.out.println ("Enter the name of the book:");  
Bookname = sc.nextLine();  
System.out.println ("Enter the name of the author.");  
Authorname = sc.nextLine();  
System.out.println ("Enter the Price of the book.");  
Bookprice = sc.nextDouble();  
System.out.println ("Enter the number of pages of the book.");  
number of pages = sc.nextInt();  
}
```

void printdetails ()

```
{  
System.out.println ("The name of the book is: " + Bookname);  
System.out.println ("The author of the book is: " + Authorname);  
System.out.println ("Price of the book is: " + Bookprice);  
System.out.println ("The number of pages in the book are:  
" + number of pages);  
}
```

}

public String toString ()

```
{  
return ("Name of the book is " + Bookname + " Name of the author"  
+ Authorname + " Price of the book " + Bookprice +  
" Number of pages " + number of pages);  
}
```

}

class Bookmain

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

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

Date: 13 October 2020, Name: - Shubham Singh, USN: - 2BM19CS063

System.out.println ("Book information");

Book

Book b1 = new Book();

b1.getDetails();

b1.printDetails();

Book b2 = new Book ("Relativity", "Hawking", 848.56, 789);

b2.printDetails();

Book s = new Book();

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

int n = sc.nextInt();

Book b[] = new Book[n];

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

{

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

b[i] = new Book();

b[i].getDetails();

b[i].printDetails();

}

}

Output of the above program:

Book information.

Enter the name of the book:

Sangam

Enter the Author of the book:

Easter

Enter the Price of the book.

789.99

Enter the number of pages of the book:

1500

Date :- 13 October 2020 , Name : Jyoti Singh . USN : 1BM19CS063

The name of the book is : Sangam

The author of the book is : Lauter.

Price of the book is : 789.99

The number of pages in the book are : 1500.

Name of book, Author of book, Price of book and Number of pages in the book respectively inside Parameterized constructor are :

The name of the book is Relativity

The author of the book is : Hawking

Price of the book is : 345.56

Number of pages in the book are : 789.

Enter the number of books:

2

Enter the details of the book: 1

1) Enter the name of the book:

Grindout

Enter the Name of the author:

ajay

Enter the Price of the book:

455.33

Enter the number of pages of the book:

5647

The name of the book is : grindout

The author of the book is : ajay.

Price of the book is 455.33

The number of pages in the book are : 5647.

Enter the details of the book : ?

Page no.: (5)

Date: 13 October 2020, Name: - Shrikant Singh, USN: IBM19C08063

Enter the name of the book:

Jash

Enter the name of the author:

Atreya

Enter the price of the book:

5439.33

Enter the number of pages of the book:

1900

The name of the book is: Jash.

The author of the book is: Atreya

Price of the book is 5439.33.

The number of pages in the book are: 1900

Screen Output :

The screenshot shows a Notepad++ window with Java code for a Book class. The code includes methods for printing book details and a main method for creating multiple books and reading their details from the user. To the right of the Notepad++ window is a Command Prompt window showing the execution of the program and its output.

```
35 System.out.print("Enter the Price of the book:");
36 Bookprice=sc.nextDouble();
37 System.out.print("Enter the number of pages of the book:");
38 numberofpages=sc.nextInt();
39 }
40 void printdetails()
41 {
42     System.out.println("The name of the book is: "+Bookname);
43     System.out.println("The author of the book is: "+Authorname);
44     System.out.println("Price of the book is: "+Bookprice);
45     System.out.println("The number of pages in the book are: "+numberofpages);
46 }
47 public String toString()
48 {
49     return ("Name of the book is "+Bookname+" Name of the author "+Authorname);
50 }
51 }
52 class Bookmain
53 {
54     public static void main(String ss[])
55     {
56         Scanner sc=new Scanner(System.in);
57         System.out.println("Book information");
58         Book b1=new Book();
59         b1.getdetails();
60         b1.printdetails();
61         Book b2=new Book("Relativity", "Hawking", 345.56, 789);
62         b2.printdetails();
63         Book s=new Book();
64         System.out.println("Enter the number of books");
65         int n=sc.nextInt();
66         Book b[] = new Book[n];
67         for(int i=0;i<n;i++)
68     {
69         System.out.println("Enter the details of the book:"+ (i+1));
70         b[i]=new Book();
71         b[i].getdetails();
72         b[i].printdetails();
73     }
74 }
75 }
76 }
77 }
78 }
```

```
5647
The name of the book is: insidetheout
The author of the book is: ajay
Price of the book is: 455.33
The number of pages in the book are: 5647
Enter the details of the book:2
Name of book, Author of book, Price of book and Number of pages in the book respectively inside Default constructor:
Enter the name of the book:
josh
Enter the Name of the author:
streyo
Enter the Price of the book:
5439.33
Enter the number of pages of the book:
1980
The name of the book is: josh
The author of the book is: streyo
Price of the book is: 5439.33
The number of pages in the book are: 1980
C:\Users\hriti\Documents>
```

Week 8 Lab Program 4 :

Q. Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle 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.

Observation copy :

Date: 3 - 11 - 2020, Name: Hritik Singh, USN: 1BM19CS063.

OOP Lab Program Week 8 :-

Lab Exercise 4

Input for Lab Exercise 4 :-

abstract class Shape

{

 int a, b;

 abstract void printArea(int a, int b);

}

class Rectangle extends Shape

{

 int areaOfRectangle;

 void printArea(int l, int m)

{

 a = l;

 b = m;

 areaOfRectangle = a * b;

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

}

class Triangle extends Shape

{

 int areaOfTriangle;

 void printArea(int x, int y)

{

 a = x;

 b = y;

 areaOfTriangle = (int)(0.5 * a * b);

 System.out.println("The area of triangle is: " + areaOfTriangle);

}

class Circle extends Shape.

Page no. : ②

Date: 3-11-2020, Name: Shubh Singh, USN: 1BM19CS063

{

```
double areaOfCircle;  
void printArea( int z, int d )  
{
```

a = z;

d = 0;

areaOfCircle = 3.14 * a * a;

```
} System.out.println ("The area of Circle is " + areaOfCircle);
```

}

```
class ShapeMain
```

{

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

Rectangl r1 = new Rectangl();

Triangl t1 = new Triangl();

Circle c1 = new Circle();

Shape RefOfShape:

RefOfShape = &r1;

RefOfShape.printArea(15, 20);

RefOfShape = &t1;

RefOfShape.printArea(30, 30);

RefOfShape = &c1;

RefOfShape.printArea(10, 0);

}

Output :

The area of Rectangl is: 300. unit²

The area of Triangl is: 450. unit²

The area of Circle is: 314. unit²

Screen Output:

The screenshot shows a Windows desktop environment. In the foreground, there is a Notepad++ window titled "C:\Users\hriti\Documents\oopweek8.java - Notepad++". The code in the editor is as follows:

```
14 System.out.println("The area of Rectangle is:"+areaofrectangle);
15 }
16 }
17 class Triangle extends Shape
18 {
19     int areaoftriangle;
20     void printArea(int x, int y)
21     {
22         a=x;
23         b=y;
24         areaoftriangle=(int) (0.5*a*b);
25         System.out.println("The area of Triangle is:"+areaoftriangle);
26     }
27 }
28
29 class Circle extends Shape
30 {
31     double areaofcircle;
32     void printArea(int x, int d)
33     {
34         a=x;
35         d=d;
36         areaofcircle=3.14*a*a;
37         System.out.println("The area of Circle is:"+areaofcircle);
38     }
39 }
40
41 class ShapeMain
42 {
43     public static void main(String[] args)
44     {
45         Rectangle r1=new Rectangle();
46         Triangle t1=new Triangle();
47         Circle c1=new Circle();
48         Shape Refofshape;
49         Refofshape=r1;
50         Refofshape.printArea(15,20);
51         Refofshape=t1;
52         Refofshape.printArea(30,30);
53         Refofshape=c1;
54         Refofshape.printArea(10,0);
55     }
56 }
```

Below the Notepad++ window is a Command Prompt window titled "Command Prompt". The command history and output are as follows:

```
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\hriti>cd..
C:\Users\hriti>cd..

C:\Users\hriti>cd C:\Users\hriti\Documents
C:\Users\hriti\Documents>javac oopweek8.java

C:\Users\hriti\Documents>java ShapeMain
The area of Rectangle is:300
The area of Triangle is:450
The area of Circle is:314.0
```

The status bar at the bottom of the screen shows the following information: length : 1,161 lines : 57 Ln : 46 Col : 32 Sel : 0 | 0 Windows (CR LF) UTF-8 96% 17:46 02-11-2020 ENG

Week 8 Lab Exercise 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 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

Observation Copy:

Date :- 3 November 2020, Name :- Harshit Singh, USN :- 1BM19CS063.

OOP Lab Program Week 8 :-

Lab Exercise 5 :-

Input for Lab Exercise 5 :-

```
import java.util.*;
class Account
{
    String name;
    int type;
    Account (String n, double a, int t, double b)
    {
        name = n;
        acc_no. = a;
        type = t;
        bal = b;
    }
    void display (String name, double acc_no, int type, double bal)
    {
        System.out.println ("Name :" + name);
        System.out.println ("Account No. :" + acc_no);
        System.out.println ("Type of account :" + type);
        System.out.println ("Balance :" + bal);
    }
}
class Sav extends Account
{
    double w, i;
    Sav (String n, double a, int t, double b, double wd, double in)
    {
        super (n, a, t, b);
        w = wd;
        i = in;
    }
}
```

Date: 3-November-2020, Name: Shrikant Singh, USN: 18M19C5063.

```

void wdbal(double bal, double w) {
    System.out.println("Withdrawal Amount : " + w);
    bal = bal - w;
    System.out.println("Remaining balance is : " + bal);
}

class Current extends Account {
    double p;
    Current(String n, double a, int t, double b, double pe, double i) {
        super(n, a, t, b);
        p = pe;
    }
    void pen(double bal) {
        System.out.println("Penalty is imposed of 500 Rupees as
                           balance is below 100 Rupees");
        bal = bal - 500;
        System.out.println("Remaining balance is : " + bal);
    }
}

class Series {
    public static void main (String [ ] args) {
        Scanner in = new Scanner (System.in);
        String name = " ";
        double acc_no = 0, bal = 0, w = 0, i = 0, p = 0;
        int type = 0;
        Account a = new Account (name, acc_no, type, bal);
        Sav as = new Sav (name, acc_no, type, bal, w, i);
        System.out.println();
        Current c = new Current (name, acc_no, type, bal, p);
    }
}

```

Date:- 3 November 2020, Name:- Shrikhil Singh, OSNL:- 1BM19CS063

```

System.out.print("Name : ");
name = in.next();

System.out.print("Account type (1: Savings Account | 2: Current Account) : ");
type = in.nextInt();

System.out.print("Enter initial amount in account : ");
bal = in.nextDouble();

a. display(name, acc_no, type, bal);

if (type == 1)
{
    System.out.print("Do you want to withdraw (1: Yes 2: No) : ");
    j = in.nextInt();

    if (j == 1)
    {
        System.out.print("Enter withdrawal amount : ");
        w = in.nextDouble();
        ac.withdraw(bal, w);
    }
    else if (type == 2)
    {
        if (bal < 1000)
            ac.pen(bal);
    }
}

System.out.print("Do you want to withdraw (1: Yes 2: No) ? : ");
K = in.nextInt();

if (K == 1)
{
    System.out.print("Enter withdrawal amount : ");
    w = in.nextDouble();
    ac.withdraw(bal, w);
}

```

Date: 3-November 2020, Name: Shrikant Singh, USN: 1BME3CS063.

```
3
else {
    System.out.println ("\" You haven't entered proper choice !!!\"");
}
}
```

Output of the above program:

Name :

Shrikant

Amount No. :

30

Account type (1: Savings Account 2: Current Account) :

1.

Enter initial amount in account :

4000

Name: Shrikant

Amount No. : 30.0

Type of account : 1.

Balance : 4000.0

Do you want withdrawal (1: Yes 2: No) ?

1.

Enter withdrawal amount :

8000

Withdrawal Amount : 3000

Remaining Balance is : 1000 ;

Name :

Rishi

Amount no. :

40

Account type (1: Savings Account 2: Current Account) :

2.

Page no.: 5.

Date: 3 November 2020, Name: Prithviraj Singh, USN: 1BM29CS063

Enter initial amount in account:

600

Name: Richi

Account No.: 400.

Type of Account: 2.

Balance: 600.0.

Penalty is imposed of 500 Rupees as balance is below 1000 Rupees.
Remaining balance is: 100.

Screen Output:

The screenshot shows a Notepad++ interface with several tabs open, including `www.java`, `whatsapp.java`, `activeshellword.java`, `maven.java`, `segment.java`, `studentsee.java`, `PlayGame.java`, `oopweek0_5.java`, and `oopweek0_6.java`. The main editor area contains Java code for a bank account system. A Command Prompt window is open, showing the execution of the program. The user enters account details and a withdrawal amount, and the program calculates the balance and applies a penalty for a low balance.

```
55 Scanner in = new Scanner(System.in);
56 String name;
57 double bal,w,i,p;
58 int type;
59 Account a=new Account(name,acc_no,type,bal);
60 Sav as=new Sav(name,acc_no,type,bal,w,i);
61 Current c=new Current(name,acc_no,type,bal,p,i);
62 System.out.println("Name : ");
63 name=in.nextLine();
64 System.out.println("Account No. : ");
65 acc_no=in.nextDouble();
66 System.out.println("Account type (1:Savings Account\2:Current Account) : ");
67 type=in.nextInt();
68 bal=in.nextDouble();
69 System.out.println("Enter initial amount in account : ");
70 bal=as.wdbal(bal,w);
71 a.display(name,acc_no,type,bal);
72 if(type==1){
73     System.out.println("Do you want to withdraw (1:Yes 2:No)?");
74     jin.nextInt();
75     if(jin==1){
76         System.out.println("Enter withdrawal amount : ");
77         w=in.nextDouble();
78         as.wdbal(bal,w);
79     }
80 }else if(type==2){
81     if(bal<1000){
82         c.pen(bal);
83     }
84     else{
85         System.out.println("Do you want to withdraw (1:Yes 2:No)?");
86         k=jin.nextInt();
87         if(k==1){
88             System.out.println("Enter withdrawal amount : ");
89             w=in.nextDouble();
90             as.wdbal(bal,w);
91         }
92     }
93 }
94 else{
95     System.out.println("You havent entered proper choice!!!");
96 }
97 }
98 }
```

```
Enter initial amount in account : 10000
Name : Hritik
Account No. : 3000
Type of account : 1
Balance : 4000.0
Do you want to withdraw (1:Yes 2:No)?
1
Enter withdrawal amount :
3000
Withdrawal Amount : 3000.0
Remaining balance is : 10000.0
C:\Users\hriti\Documents>java Series
Name :
Rishi
Account No. :
40
Account type (1:Savings Account 2:Current Account) :
2
Enter initial amount in account :
500
Name : Rishi
Account No. : 40.0
Type of account : 2
Balance : 000.0
Penalty is imposed of 500 Rupees as balance is below 1000 Rupees
Remaining balance is : 100.0
```

Week 9-Lab Program 6

Solve this program and write the procedure you have used to execute this in your observation

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.

Observation Copy:

Page no.: ①

Date: 17 November 2020, Name: Shubh Singh, USN: 1BM13C5063.

OOP Lab Exercise:

Wk 9 - Lab Program 6:

Input of the program:

File Name: CSE, Class name: Student.

Student.java:

package CSE;

public class Student
{

 public String USN;
 public String name;
 public int sem;
 public Student()
}

}

 public Student(String USN, String name, int sem)
 {

 this.USN = USN;

 this.name = name;

 this.sem = sem;

}
}

File Name: CSE, Class Name: Internals.

Internals.java:

Date : 17 November 2020, Name : Shubh Singh, USN : 1BM19CS063.

```

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.print("Enter the CIE mark in Subject " + (i + 1));
            cie[i] = s.nextInt();
        }
    }
}

```

File name: SEE, Class Name: Internals

Internals.java

```

package SEE;
import java.util.Scanner;
public class Internals extends CIE.Student
{
    public Internals (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()
    {
    }
}

```

Date: 17 November 2020, Name: Hritik Singh, DSN: 2Bmgcs0

8.

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

{

System.out.println ("Enter the
SSE mark in
Subject " + (i+1))

See [i] = s.nextInt();

{

3.

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

9.

File Name: Final Marks

Class: Final Marks

FinalMarks.java: CSE.Externals = [i];

```
import CSE.*;
```

```
import SEE.*;
```

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

```
class Final Marks
```

{

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

Scanner s = new Scanner (System.in);
System.out.print ("Enter the number of students")

int n = s.nextInt();

CSE.Externals ob1[] = new SEE.Externals[n];

CSE.Internals ob2[] = new CSE.Internals[n];

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

System.out.print ("Enter the USN, Name")

[i] and Gender of

student " + (i+1))

Date: 17 November 2020, Name: Hritik Singh, USN: 1BM12CS043.

```

String u = s.next();
String na = s.next();
int se = s.nextInt();
ob1[i] = new SEE.Externals(u, na, se);
ob1[i].get();
ob[i] = new CIE.Externals();
ob[i].get();

for (int i = 0; i < n; i++) {
    System.out.print("The Total Marks of Student"
                    + (i + 1) + " are ");
    for (int j = 0; j < 5; j++) {
        System.out.print("Subject" + (j + 1) + " = "
                        + ob[i].cse[j] + ob1[i].see[j]);
    }
}

```

Output of the above program is:-

Enter the number of Students:-

1.

Enter the USN, Name and semester of Student 1.

Bmcs Hritik S.

Enter the SEE mark in Subject 1.

20

Enter the SEE mark in Subject 2.

23

Enter the SEE mark in Subject 3.

Date : 17 November 2020, Name : Srikish Singh, USN : 18M43CS063

25.

Enter the SEE mark in Subject 4:-

23

Enter the SEE mark in Subject 5:-

25.

Enter the CIE mark in Subject 1:-

67

Enter the CIE mark in Subject 2:-

69

Enter the CIE mark in Subject 3:-

80

Enter the CIE mark in Subject 4:-

70

Enter the CIE mark in Subject 5:-

87.

The "Total Marks" of Student 1 are:-

Subject 1 = 87

Subject 2 = 92

Subject 3 = 105

Subject 4 = 93

Subject 5 = 112

Screen Output:

The screenshot shows a Notepad++ window with the following Java code:

```
1 package CIE;
2 import java.util.Scanner;
3 public class Internals extends Student {
4     Scanner s=new Scanner(System.in);
5     public int[] cie=new int[5];
6     public void get()
7     {
8         for(int i=0;i<5;i++)
9             System.out.print("ENTER THE CIE MARK IN SUBJECT"+(i+1));
10            cie[i]=s.nextInt();
11    }
12 }
13 }
```

The code is part of a class named 'Internals' which extends 'Student'. It contains a constructor that initializes an array 'cie' of size 5. The 'get' method prompts the user to enter five integers, one for each subject, and stores them in the 'cie' array.

Scanned with CamSc

The screenshot shows a Notepad++ window with the following Java code:

```
1 package SEE;
2 import java.util.Scanner;
3
4 public class Externals extends CIE.Student {
5     public Externals(String usn, String name, int sem)
6     {
7         super(usn, name, sem);
8     }
9     Scanner s = new Scanner(System.in);
10    public int[] see=new int[5];
11    public void get()
12    {
13        for(int i=0;i<5;i++)
14        {
15            System.out.println("ENTER THE SEE MARK IN SUBJECT"+(i+1));
16            see[i]=s.nextInt();
17        }
18    }
}
```

The code defines a class `Externals` that extends `CIE.Student`. It has a constructor that takes `usn`, `name`, and `sem` as parameters and calls the superclass's constructor. It also has a `see` array of integers of size 5. The `get` method prompts the user to enter marks for five subjects and stores them in the `see` array.

Scanned with CamScanner

```
1 import CIE.*;
2 import SEE.*;
3
4 import java.util.Scanner;
5
6 class FinalMarks {
7     public static void main(String args[]) {
8         Scanner s=new Scanner(System.in);
9         System.out.println("ENTER THE NUMBER OF STUDENTS");
10        int n=s.nextInt();
11        SEE.Externals ob[] =new SEE.Externals[n];
12        CIE.Internals ob[] =new CIE.Internals[n];
13        for(int i=0;i<n;i++)
14        {
15            System.out.println("ENTER THE USN,NAME AND SEMESTER OF STUDENT"+(i+1));
16            String usn=s.next();
17            String name=s.next();
18            int sem=s.nextInt();
19            ob[i]=new SEE.Externals(usn,name,sem);
20            ob[i].get();
21            ob[i]=new CIE.Internals();
22            ob[i].get();
23        }
24        for(int i=0;i<n;i++)
25        {
26            System.out.println("THE TOTAL MARKS OF STUDENT"+(i+1)+" ARE");
27            for(int j=0;j<5;j++)
28            {
29                System.out.println("SUBJECT"+(j+1)+"="+ob[i].cie[j]+ob[i].see(j));
30            }
31        }
32    }
33}
```

Scanned with CamScanner

The screenshot shows a Windows desktop environment. In the center, there is a Notepad++ window titled "C:\Users\hriti\Documents\FinalMarks\FinalMarks.java - Notepad++". The code in the editor is as follows:

```

1 import CIE.*;
2 import SEE.*;
3
4 import java.util.Scanner;
5
6 class FinalMarks {
7     public static void main(String args[]) {
8         Scanner s=new Scanner(System.in);
9         System.out.println("ENTER THE NUMBER OF STUDENTS");
10        int n=s.nextInt();
11        SEE.Externals ob1[]=new SEE.Externals[n];
12        CIE.Internals ob2[]=new CIE.Internals[n];
13        for(int i=0;i<n;i++) {
14            System.out.println("ENTER THE USN,NAME AND SEMESTER OF STUDENT"+(i+1));
15            String usn=s.next();
16            String na=s.next();
17            int se=s.nextInt();
18            int see=s.nextInt();
19            ob1[i]=new SEE.Externals(usn,na,se);
20            ob1[i].get();
21            ob2[i]=new CIE.Internals();
22            ob2[i].get();
23        }
24        for(int i=0;i<n;i++) {
25            {
26                System.out.println("THE TOTAL MARKS OF STUDENT"+(i+1)+" ARE");
27                for(int j=0;j<5;j++) {
28                    {
29                        System.out.println("SUBJECT"+(j+1)+"="+ob1[i].cie[j]+ob1[i].see[j]);
30                    }
31                }
32            }
33        }
34    }
}

```

To the right of the Notepad++ window is a Command Prompt window titled "Command Prompt - java FinalMarks". The command entered is "C:\Users\hriti\Documents\FinalMarks>javac FinalMarks.java". The output of the program is displayed, asking for student details and printing total marks.

Scanned with CamScanner

The screenshot shows a Windows desktop environment. In the foreground, there is a Notepad++ window titled "C:\Users\hrith\Documents\FinalMarks\FinalMarks.java - Notepad++". The code in the editor is as follows:

```

1 import CIE.*;
2 import SEE.*;
3
4 import java.util.Scanner;
5
6 class FinalMarks {
7     public static void main(String args[]) {
8         Scanner s=new Scanner(System.in);
9         System.out.println("ENTER THE NUMBER OF STUDENTS");
10        int n=s.nextInt();
11        SEE.Externals ob1[]=new SEE.Externals[n];
12        CIE.Internals ob2[]=new CIE.Internals[n];
13        for(int i=0;i<n;i++)
14        {
15            System.out.println("ENTER THE USN,NAME AND SEMESTER OF STUDENT"+(i+1));
16            String usn=s.next();
17            String na=s.next();
18            int sem=s.nextInt();
19            ob1[i]=new SEE.Externals(usn,na,sem);
20            ob1[i].get();
21            ob2[i]=new CIE.Internals();
22            ob2[i].get();
23        }
24        for(int i=0;i<n;i++)
25        {
26            System.out.println("THE TOTAL MARKS OF STUDENT"+(i+1)+" ARE");
27            for(int j=0;j<5;j++)
28            {
29                System.out.println("SUBJECT"+(j+1)+"="+ob1[i].cie[j]+ob2[i].see[j]);
30            }
31        }
32    }
33 }

```

To the right of the Notepad++ window is a Command Prompt window titled "Command Prompt". The output of the program is displayed in the Command Prompt window:

```

ENTER THE USN,NAME AND SEMESTER OF STUDENT
Bmcs Hritik 3
ENTER THE SEE MARK IN SUBJECT1
20
ENTER THE SEE MARK IN SUBJECT2
23
ENTER THE SEE MARK IN SUBJECT3
25
ENTER THE SEE MARK IN SUBJECT4
23
ENTER THE SEE MARK IN SUBJECT5
25
ENTER THE CIE MARK IN SUBJECT1
67
ENTER THE CIE MARK IN SUBJECT2
69
ENTER THE CIE MARK IN SUBJECT3
80
ENTER THE CIE MARK IN SUBJECT4
70
ENTER THE CIE MARK IN SUBJECT5
87
THE TOTAL MARKS OF STUDENT1 ARE
SUBJECT1=87
SUBJECT2=92
SUBJECT3=105
SUBJECT4=93
SUBJECT5=112

```

The Command Prompt window also shows the path "C:\Users\hrith\Documents\FinalMarks>" at the bottom.

Scanned with CamScanner

The screenshot shows a window titled "C:\Users\hrin\Documents\FinalMarks\cie\Student.java - Notepad+" with the following Java code:

```
1 package CIE;
2
3 public class Student
4 {
5     public String usn;
6     public String name;
7     public int sem;
8     public Student()
9     {
10    }
11 }
12 public Student(String usn, String name, int sem)
13 {
14     this.usn=usn;
15     this.name=name;
16     this.sem=sem;
17 }
```

The code defines a class named "Student" with three fields: "usn", "name", and "sem". It has a default constructor and a parameterized constructor that initializes the fields. The code is highlighted in purple and black.

Scanned with CamScanner

Week 10: Lab Programs-7 :

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

Observation Copy:

Date: 24 November 2020, Name: Shrikant Singh, USN: 18M19CS063

OOP Lab Exercise:

Wk 10 - Lab Program 7:-

Input of the program:-

```
import java.util.Scanner;
class MultipleObjpara < N, A, M >
{
    N obj 1;
    A obj 2;
    M obj 3;
}

Multiple obj para (N x1, A x2, M x3)
{
    obj 1 = x1;
    obj 2 = x2;
    obj 3 = x3;
}

void ObjDynes()
{
    System.out.print("Type of name is " + obj 1.get Class().getName());
    System.out.print(" Type of age is " + obj 2.get Class().getName());
    System.out.print(" Type of marks is " + obj 3.get Class().getName());
}

N get Obj 1()
{
    return obj 1;
}

A get Obj 2()
{
    return obj 2;
}

M get Obj 3()
```


Date: 24 November 2020, Name: Srikkrishna Singh, USN: IBM19.ICS063

```

3
    return obj3;
3
class Multipleobjpara {
    public static void main (String args[]) {
        String name;
        int age;
        double marks;
        System.out.println ("Enter the name of the student");
        Scanner sc = new Scanner (System.in);
        name = sc.next();
        System.out.println ("Enter the age of " + name);
        age = sc.nextInt();
        System.out.println ("Enter the marks of " + name);
        marks = sc.nextDouble();
    }
}

Multiple obj para < String, Integer, Double > tyObj = new Multipleobjpara
< String, Integer, Double > (name, age, marks);
tyObj. Obj types ();
String n = tyObj.getObj1 ();
System.out.println ("Name : " + n);
int a = tyObj.getObj2 ();
System.out.println ("Age : " + a);
double m = tyObj.getObj3 ();
System.out.println ("Marks : " + m);
3

```

Output of the above program is given as..

Page no. :- 3

Date: 24 November 2020, Name: Hritik Singh, USM: 18M13CS063.

Enter the name of the student:
Hritik

Enter the age of Hritik
30

Enter the marks of Hritik
99.99

Type of name is java.lang.String.
Type of age is java.lang.Integer
Type of marks is java.lang.Double

Name : Hritik

Age : 30

Marks : 99.99

Screen Output:

The screenshot shows a Windows desktop environment. In the foreground, there is a Notepad++ window titled "C:\Users\hriti\Documents\Multipleobjpara.java - Notepad++". The code in the editor is as follows:

```
1 import java.util.Scanner;
2 class Multipleobjpara<N,A,M>
3 {
4     N obj1;
5     A obj2;
6     M obj3;
7     Multipleobjpara(N x1, A x2, M x3)
8     {
9         obj1=x1;
10        obj2=x2;
11        obj3=x3;
12    }
13    void Objtypes()
14    {
15        System.out.println("Type of name is "+obj1.getClass().getName());
16        System.out.println("Type of age is "+obj2.getClass().getName());
17        System.out.println("Type of marks is "+obj3.getClass().getName());
18    }
19    N getobj1()
20    {
21        return obj1;
22    }
23    A getobj2()
24    {
25        return obj2;
26    }
27    M getobj3()
28    {
29        return obj3;
30    }
31 }
32 class Multipleobjparal
33 {
34     public static void main(String args[])
35     {
36         String name;
37         int age;
38         double marks;
39         System.out.println("Enter the name of the student");
40         Scanner sc=new Scanner(System.in);
41         name=sc.next();
42         System.out.println("Enter the age of "+name);
43         age=sc.nextInt();
44         age=sc.nextDouble();
45     }
46 }
```

Below the Notepad++ window is a Command Prompt window titled "Command Prompt". The output from the command prompt is:

```
Microsoft Windows [Version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\hriti>cd..
C:\Users>cd..
C:\Users\hriti>cd C:\Users\hriti\Documents
C:\Users\hriti>javac Multipleobjpara.java
C:\Users\hriti>java Multipleobjpara1
Enter the name of the student
Hritik
Enter the age of Hritik
30
Enter the marks of Hritik
99.99
Type of name is java.lang.String
Type of age is java.lang.Integer
Type of marks is java.lang.Double
Name: Hritik
Age: 30
Marks: 99.99
```

The taskbar at the bottom of the screen shows several pinned icons, including File Explorer, Edge, Mail, and File History. The system tray indicates battery level at 99%, network connection, and system status.

Scanned with CamScanner

The screenshot shows a Notepad++ window with the following Java code:

```
1 import java.util.Scanner;
2 class Multipleobjpara<N,A,M>
3 {
4     N obj1;
5     A obj2;
6     M obj3;
7
8     Multipleobjpara(N x1, A x2, M x3)
9     {
10         obj1=x1;
11         obj2=x2;
12         obj3=x3;
13     }
14     void Objtypes()
15     {
16         System.out.println("Type of name is "+obj1.getClass().getName());
17         System.out.println("Type of age is "+obj2.getClass().getName());
18         System.out.println("Type of marks is "+obj3.getClass().getName());
19     }
20     N getobj1()
21     {
22         return obj1;
23     }
24     A getobj2()
25     {
26         return obj2;
27     }
28     M getobj3()
29     {
30         return obj3;
31     }
32 }
33 class Multipleobjparal
34 {
35     public static void main(String args[])
36     {
37         String name;
38         int age;
39         double marks;
40         System.out.println("Enter the name of the student");
41         Scanner sc=new Scanner(System.in);
42         name=sc.next();
43         System.out.println("Enter the age of "+name);
44         age=sc.nextInt();
45     }
46 }
```

The code defines a class `Multipleobjpara` with three generic type parameters: `N`, `A`, and `M`. It contains methods to set and get objects of type `N`, `A`, and `M`. The `main` method prompts the user for the student's name and age.

Scanned with CamScanner

```
13     }
14     void Objtypes()
15     {
16         System.out.println("Type of name is "+obj1.getClass().getName());
17         System.out.println("Type of age is "+obj2.getClass().getName());
18         System.out.println("Type of marks is "+obj3.getClass().getName());
19     }
20     M getobj1()
21     {
22         return obj1;
23     }
24     A getobj2()
25     {
26         return obj2;
27     }
28     M getobj3()
29     {
30         return obj3;
31     }
32 }
33 class Multipleobjpara
34 {
35     public static void main(String args[])
36     {
37         String name;
38         int age;
39         double marks;
40         System.out.println("Enter the name of the student");
41         Scanner sc=new Scanner(System.in);
42         name=sc.nextLine();
43         System.out.println("Enter the age of "+name);
44         age=sc.nextInt();
45         System.out.println("Enter the marks of "+name);
46         marks=sc.nextDouble();
47         Multipleobjpara<String, Integer, Double> tObj = new Multipleobjpara<String, Integer, Double>(name, age, marks);
48         tObj.Objtypes();
49         String s=tObj.getobj1();
50         int i=tObj.getobj2();
51         double m=tObj.getobj3();
52         System.out.println("Name: " +s);
53         System.out.println("Age: " +i);
54         System.out.println("Marks: " +m);
55     }
56 }
```

Scanned with CamScanner

Week 10: Lab Programs 8:

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=father’s age.

Observation Copy:

Date: 24 November 2020, Name: Harik Singh, USN: 1 BM 19 CS 063

OOJ Lab Exercise:

Week - 10 - Lab Program 8:-

Input of the program:

```

import java.util.Scanner;
class MyException extends Exception {
    private int detail;
    MyException (int a) {
        detail = a;
    }
    public String toString() {
        return "ageException[" + detail + "]";
    }
}
class Father {
    int fatherage;
    String fathername;
}
class Son extends Father {
    int Sage;
    String Sname;
    Son (String name, int age, int fatherage, String fathername)
        throws MyException {
        System.out.println ("Son Name : " + name);
        System.out.println ("Son age : " + age);
        System.out.println ("Father's age " + fatherage);
        System.out.println ("Father's name : " + fathername);
    }
}

```

Date: 24 November 2020; Name: Bhilal Singh, USN: 2BM19CS063

```
if (age >= fatherage)
    throw new MyException(fatherage)
System.out.println("Constructor exit");
```

{

class ExceptionDemo

{

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

```
String a;
```

```
int b;
```

```
int c;
```

```
String d;
```

```
Scanner sc = new Scanner(SystemIn);
```

```
System.out.print("Enter name of the son");
```

```
a = sc.next();
```

```
System.out.print("Enter the age of the son");
```

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

```
System.out.print("Enter the Age of the father");
```

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

```
System.out.print("Enter the name of the father");
```

{

try

```
{
```

```
Son s = new Son(a, b, c, d);
```

```
}
```

```
catch (MyException e)
```

{

```
System.out.print("Caught "+e);
```

}

{

}

Date: 24 November 2020, Name: Hritik Singh, Date: 24 November 2020

Output of the above program is:

Enter the name of the son

Hritik

Enter the age of the son

30

Enter the age of the father

30

Enter the name of the father

T

Son Name : Hritik

Son age : 30

Father's age : 30

Father's name : T

(right age exception [30]).

Another Output for the same program with different input:

Enter the name of the son

Hritik

Enter the age of the son

30

Enter the age of the father

60

Enter the name of the father

T

Son Name : Hritik

Son age : 30

Father age : 60

Father's name : T

(Constructor exit.)

Screen Output:

The screenshot shows a Windows desktop environment. In the center, there is a Notepad++ window titled "C:\Users\hriti\Documents\ExceptionDemo.java - Notepad++". The code in the editor is as follows:

```
24 String Sname;
25
26 Son(String name,int age,int fatherage, String fathernam) throws MyException
27 {
28     System.out.println("Son Name: "+name);
29     System.out.println("Son age: "+age);
30     System.out.println("Father age "+ fatherage);
31     System.out.println("Father's name :" +fathernam);
32     if(age>fatherage)
33         throw new MyException(fatherage);
34     System.out.println("Constructor exit");
35 }
36 }
37 class ExceptionDemo
38 {
39     public static void main(String args[])
40     {
41         String a;
42         int b;
43         int c;
44         String d;
45
46         Scanner sc=new Scanner(System.in);
47         System.out.println("Enter the name of the son");
48         a=sc.nextLine();
49         System.out.println("Enter the age of the son");
50         b=sc.nextInt();
51         System.out.println("Enter the Age of the father");
52         c=sc.nextInt();
53         System.out.println("Enter the name of the Father");
54         d=sc.nextLine();
55
56         try
57         {
58             Son s=new Son(a,b,c,d);
59         }
60         catch (MyException e)
61         {
62             System.out.println("Caught " + e);
63         }
64     }
65 }
66 }
67 }
```

To the right of the Notepad++ window, a Command Prompt window is open, showing the execution of the Java program. The command entered is "java ExceptionDemo". The output of the program is displayed in the Command Prompt window:

```
C:\Users\hriti\Documents>java ExceptionDemo
Enter the name of the son
Hritik
Enter the age of the son
30
Enter the Age of the father
60
Enter the name of the Father
T
Son Name: Hritik
Son age: 30
Father age 60
Father's name :T
Caught ageException[30]
```

The Command Prompt window also shows the current working directory as "C:\Users\hriti\Documents". At the bottom of the screen, the taskbar is visible with various icons and the system tray showing battery level (99%), network status, and date/time (24-11-2020).

Scanned with CamScanner

The screenshot shows a Windows desktop environment. In the foreground, there is a Notepad++ window titled "C:\Users\hriti\Documents\ExceptionDemo.java - Notepad++". The code in the editor is as follows:

```

1 import java.util.Scanner;
2 class MyException extends Exception
3 {
4     private int detail;
5     MyException(int a)
6     {
7         detail = a;
8     }
9     public String toString()
10    {
11        return "ageException[" + detail + "]";
12    }
13 }
14
15 class Father
16 {
17     int fatherage;
18     String fathernam;
19 }
20
21
22 class Son extends Father
23 {
24     int Sage;
25     String Sname;
26     Son(String name,int age,int fatherage, String fathernam) throws MyException
27     {
28         System.out.println("Son Name: " +name);
29         System.out.println("Son age: " +age);
30         System.out.println("Father age " + fatherage);
31         System.out.println("Father's name :" +fathernam);
32         if(age>fatherage)
33             throw new MyException(fatherage);
34         System.out.println("Constructor exit");
35     }
36 }
37
38 class ExceptionDemo
39 {
40     public static void main(String args[])
41     {
42         String a;
43         int b;
44         int c;
45         String d;
46     }
}

```

Below the Notepad++ window is a Command Prompt window titled "Command Prompt". The session shows the following interaction:

```

Microsoft Windows [Version 10.0.18363.1108]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\hriti>cd..
C:\Users\hriti>cd C:\Users\hriti\Documents
C:\Users\hriti>javac ExceptionDemo.java
C:\Users\hriti>java ExceptionDemo
Enter the name of the son
Hritik
Enter the age of the son
30
Enter the Age of the father
30
Enter the name of the Father
Hritik
Son Name: Hritik
Son age: 30
Father age 30
Father's name :T
Caught ageException[30]

```

Scanned with CamScanner

WEEK 11

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.

Observation Copy:

Date :- 8 December 2020, Name:- Kritik Singh, USN:- 1BM19CS063.

OOJ Lab Exercises:-

Week 11 - Lab Program 9:-

Input of the program:-

Class NewThread implements Runnable

```
String name;  
String identity;  
Thread t;  
long time;  
New Thread (String threadname, long time)
```

time = time 1;

name = thread name;

t = new Thread (this, name);

System.out.println ("First Thread" + t);

t.start();

}

public void run ()

{

try

```
for (int i = 0; i < 20; i++)
```

{

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

Thread.sleep (time);

}

catch (InterruptedException e)

{

```
System.out.println (Name + " Interrupted");
```

Date:- 8 December 2020, Name:- Tribikram Singh, USN:- 1BM19CS068

? System.out.println (name + " exiting");

class Multithread {

 public static void main (String args [])

 new NewThread ("BMS College of Engineering", 1000);

 new NewThread ("CSE", 2000);

Output of the above program is:-

First Thread Thread [BMS College of Engineering, 5, main]

First thread Thread [CSE , 5 , Main]

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS college of Engineering

CSE

CSE

CSE

CSE

BMS college of Engineering

CSE

CSE

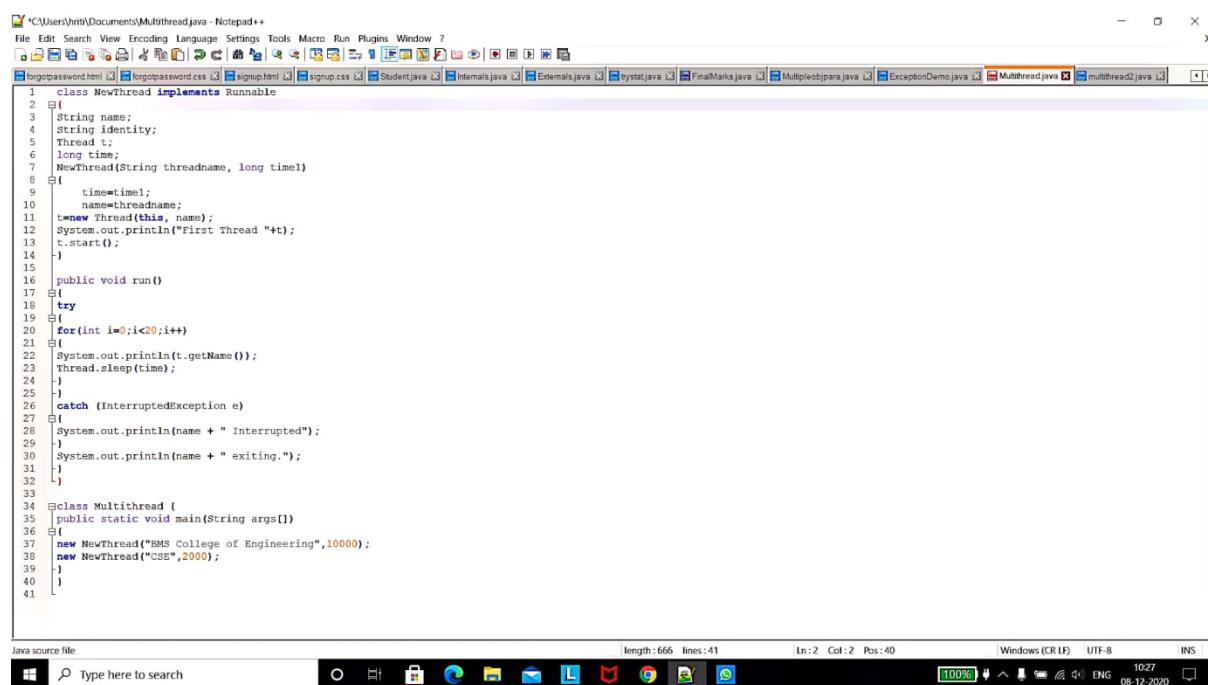
CSE

CSE

CSE

BMS college of Engineering.

Screen Output:



The screenshot shows a Notepad++ window with the following Java code:

```
1  class NewThread implements Runnable
2  {
3      String name;
4      String identity;
5      Thread t;
6      long time;
7      NewThread(String threadname, long time)
8      {
9          timetime;
10         namethreadname;
11         tnew Thread(this, name);
12         System.out.println("First Thread "+t);
13         t.start();
14     }
15
16     public void run()
17     {
18         try
19         {
20             for(int i=0;i<20;i++)
21             {
22                 System.out.println(t.getName());
23                 Thread.sleep(time);
24             }
25         }
26         catch (InterruptedException e)
27         {
28             System.out.println(name + " Interrupted");
29         }
30         System.out.println(name + " exiting.");
31     }
32 }
33
34 class Multithread {
35     public static void main(String args[])
36     {
37         new NewThread("BMS College of Engineering",10000);
38         new NewThread("CSS",2000);
39     }
40 }
```

The code defines a `NewThread` class that implements the `Runnable` interface. It has fields for `name`, `identity`, and `Thread`. The constructor takes a `String` for the name and a `long` for the sleep time. The `run` method contains a loop that prints the thread's name every second and sleeps for the specified time. It handles `InterruptedException`. The `Multithread` class has a static `main` method that creates two `NewThread` objects with different names and sleep times.

Scanned with CamSc

The screenshot shows a Windows desktop environment. In the foreground, there is a Notepad++ window titled "C:\Users\hrithi\Documents\Multithread.java - Notepad++". The code in the editor is as follows:

```
1 class NewThread implements Runnable
2 {
3     String name;
4     String identity;
5     Thread t;
6     long time;
7     NewThread(String threadname, long time)
8     {
9         this.name = threadname;
10        this.identity = identity;
11        t = new Thread(this, name);
12        System.out.println("First Thread "+t);
13        t.start();
14    }
15    public void run()
16    {
17        try
18        {
19            for(int i=0;i<20;i++)
20            {
21                System.out.println(t.getName());
22                Thread.sleep(time);
23            }
24        }
25        catch (InterruptedException e)
26        {
27            System.out.println(name + " Interrupted");
28        }
29        System.out.println(name + " exiting.");
30    }
31 }
32
33
34 class Multithread {
35     public static void main(String args[])
36     {
37         new NewThread("BMS College of Engineering",10000);
38         new NewThread("CSE",2000);
39     }
40 }
```

Below the Notepad++ window, a Command Prompt window is open with the title "Command Prompt - java Multithread". The output of the command is displayed:

```
C:\Users\hrithi\Documents>java Multithread
First Thread Thread[BMS College of Engineering,5,main]
First Thread Thread[CSE,5,main]
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE exiting.
```

The taskbar at the bottom of the screen shows several pinned icons, including File Explorer, Edge, Mail, and File History. The system tray indicates the date as 08-12-2020 and the time as 10:30 AM. The status bar at the bottom right of the Notepad++ window shows "length: 666 lines: 41" and "Windows (CR LF) | UTF-8".

Scanned with CamScanner

Week 12 Programs:

Lab Program:

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 Arithmetic Exception Display the exception in a message dialog box.

Observation Copy:

Date:- 15 December 2020 , Name: Kritika Singh , USN:- 1BM19CS063

OOJ Lab Exercise:-

Week -12 - Lab Program 10 :-

Input of the program:

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Division extends Frame implements ActionListener
{
    JTextField n1, n2, res;
    Label l1, l2, lres;
    Button b;

    public Division()
    {
        setLayout(new FlowLayout());
        l1 = new Label("NUM1", Label.RIGHT);
        l2 = new Label("NUM2", Label.RIGHT);
        lres = new Label("RESULT", Label.RIGHT);
        n1 = new JTextField(10);
        n2 = new JTextField(8);
        res = new JTextField(10);
        b = new Button("DIVIDE");
        add(l1);
        add(n1);
        add(l2);
        add(n2);
        add(lres);
        add(res);
        b.addActionListener(this);
        addWindowListener(new MyWindowAdapter());
    }

    public void actionPerformed(ActionEvent ae)
    {
    }
}

```

Date: 15 December 2020, Name: Krittish Singh, USN: 1BM19CS063

{

try {

```
int num 1 = Integer.parseInt(n1.getText());  
int num 2 = Integer.parseInt(n2.getText());  
int num 3 = num 1 / num 2;
```

```
res. setText("String.. valueOf(num 3));  
} catch (NumberFormatException e) {
```

JOptionPane.showMessageDialog(this, res, "ERROR",
JOptionPane.ERROR_MESSAGE);

}

}

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

{

```
Division i = new Division();  
i.setSize(new Dimension(400, 400));  
i.setTitle("Integer Division of Two Numbers");  
i.setVisible(true);
```

}

```
class MyWindowAdapter extends WindowAdapter {
```

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

{

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

}

}

Screen Output:

C:\Users\hriti\Documents\Division.java - Notepad+

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

examform.html examform.css login.html adminlogin.html login.css abc.html Division.java admin3.html

Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\hriti>cd..
C:\Users\hriti>cd C:\Users\hriti\Documents
C:\Users\hriti\Documents>javac Division.java
C:\Users\hriti\Documents>java Division

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class Division extends Frame implements ActionListener{
6     TextField n1,n2,res;
7     Label l1,l2,lres;
8     Button b;
9     public Division(){
10         setLayout(new FlowLayout());
11         l1=new Label("NUM1",Label.RIGHT);
12         l2=new Label("NUM2",Label.RIGHT);
13         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(l1);
19         add(n1);
20         add(l2);
21         add(n2);
22         add(b);
23         add(lres);
24         add(res);
25         b.addActionListener(this);
26         addWindowListener(new MyWindowAdapter());
27     }
28     public void actionPerformed(ActionEvent ae){
29         if(ae.getSource()==b)
30         {
31             try{
32                 int num1=Integer.parseInt(n1.getText());
33                 int num2=Integer.parseInt(n2.getText());
34                 int num3=num1/num2;
35                 res.setText(String.valueOf(num3));
36             }catch(NumberFormatException ne ){
37                 JOptionPane.showMessageDialog(this,ne,"ERROR",JOptionPane.ERROR_MESSAGE);
38             }
39         }
40         catch(ArithmaticException a){
41             JOptionPane.showMessageDialog(this,a,"ERROR",JOptionPane.ERROR_MESSAGE);
42         }
43     }
44 }
```

length : 1,627 lines : 59 ln:59 Col:2 Pos:1,628 Windows (CR LF) UTF-8 INS

Scanned with CamScanner

The screenshot shows a Notepad++ window with the following Java code:

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class Division extends Frame implements ActionListener{
6     TextField n1,n2,res;
7     Label l1,l2,lres;
8     Button b;
9
10    public Division(){
11        setLayout(new FlowLayout());
12        l1=new Label("NUM1",Label.RIGHT);
13        l2=new Label("NUM2",Label.RIGHT);
14        lres=new Label("RESULT",Label.RIGHT);
15        n1=new TextField(13);
16        n2=new TextField(10);
17        res=new TextField();
18        b=new Button("DIVIDE");
19        add(l1);
20        add(n1);
21        add(n2);
22        add(lres);
23        add(res);
24        b.addActionListener(this);
25        addWindowListener(new MyWindowAdapter());
26    }
27    public void actionPerformed(ActionEvent ae){
28        if(ae.getSource()==b){
29            try{
30                int num1=Integer.parseInt(n1.getText());
31                int num2=Integer.parseInt(n2.getText());
32                int num3=num1/num2;
33                res.setText(String.valueOf(num3));
34            }catch(NumberFormatException ne ){
35                JOptionPane.showMessageDialog(this,ne,"ERROR", JOptionPane.ERROR_MESSAGE);
36            }
37            catch(ArithmeticException a){
38                JOptionPane.showMessageDialog(this,a,"ERROR", JOptionPane.ERROR_MESSAGE);
39            }
40        }
41    }
42}
43}
```

The code defines a Java application named 'Division' that extends the 'Frame' class. It uses a 'FlowLayout' for its layout. The application has three text fields: 'n1' and 'n2' for input numbers, and 'res' for the result. It also has three labels: 'l1', 'l2', and 'lres'. A button labeled 'DIVIDE' is used to trigger an action. The application implements the 'ActionListener' interface to handle button presses. It includes error handling for non-integer inputs and division by zero.

Scanned with CamScanner

The screenshot shows a Java application window titled "Division.java" in Notepad+. The code implements a simple integer division calculator. The application has three text fields: NUM1, NUM2, and RESULT, and one button labeled "DIVIDE". The "DIVIDE" button is an ActionListener that performs the division and updates the RESULT field. An error dialog is shown if the input is invalid or if division by zero is attempted.

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class Division extends Frame implements ActionListener{
6     TextField n1,n2,res;
7     Label l1,l2,lres;
8     Button b;
9     public Division(){
10         setLayout(new FlowLayout());
11         l1=new Label("NUM1",Label.RIGHT);
12         l2=new Label("NUM2",Label.RIGHT);
13         lres=new Label("RESULT",Label.RIGHT);
14         n1=new TextField(12);
15         n2=new TextField(12);
16         res=new TextField(10);
17         b=new Button("DIVIDE");
18         add(l1);
19         add(n1);
20         add(l2);
21         add(n2);
22         add(b);
23         add(lres);
24         add(res);
25         b.addActionListener(this);
26         addWindowListener(new MyWindowAdapter());
27     }
28     public void actionPerformed(ActionEvent ae){
29         if(ae.getSource()==b)
30         {
31             try{
32                 int num1=Integer.parseInt(n1.getText());
33                 int num2=Integer.parseInt(n2.getText());
34                 int num3=num1/num2;
35                 res.setText(String.valueOf(num3));
36             }catch(NumberFormatException ne ){
37                 JOptionPane.showMessageDialog(this,ne,"ERROR",JOptionPane.ERROR_MESSAGE);
38             }
39         }
40         catch(ArithmaticException a){
41             JOptionPane.showMessageDialog(this,a,"ERROR",JOptionPane.ERROR_MESSAGE);
42         }
43     }
44 }
```

Scanned with CamScanner

The screenshot shows a Java application window titled "Division.java" running in Notepad++. The code implements a simple division calculator. A command prompt window is visible, showing the execution of the application and an error message dialog box.

```

1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class Division extends Frame implements ActionListener{
6     TextField n1,n2,res;
7     Label l1,l2,lres;
8     Button b;
9     public Division(){
10         setLayout(new FlowLayout());
11         l1=new Label("NUM1",Label.RIGHT);
12         l2=new Label("NUM2",Label.RIGHT);
13         lres=new Label("RESULT",Label.RIGHT);
14         n1=new TextField(12);
15         n2=new TextField(12);
16         res=new TextField(10);
17         b=new Button("DIVIDE");
18         add(l1);
19         add(n1);
20         add(l2);
21         add(n2);
22         add(b);
23         add(lres);
24         add(res);
25         b.addActionListener(this);
26         addWindowListener(new MyWindowAdapter());
27     }
28     public void actionPerformed(ActionEvent ae){
29         if(ae.getSource()==b){
30             try{
31                 int num1=Integer.parseInt(n1.getText());
32                 int num2=Integer.parseInt(n2.getText());
33                 int num3=num1/num2;
34                 res.setText(String.valueOf(num3));
35             }catch(NumberFormatException ne ){
36                 JOptionPane.showMessageDialog(this,ne,"ERROR",JOptionPane.ERROR_MESSAGE);
37             }
38         }
39     }
40     catch(ArithmeticException a){
41         JOptionPane.showMessageDialog(this,a,"ERROR",JOptionPane.ERROR_MESSAGE);
42     }
43 }

```

Command Prompt output:

```

Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\hriti>cd..
C:\Users\hriti>cd Documents
C:\Users\hriti>javac Division.java
C:\Users\hriti>java Division

```

Error dialog message:

ERROR
java.lang.ArithmaticException: / by zero
OK

Scanned with CamScanner

The screenshot shows a Java application window titled "Division.java" in Notepad+. The code implements a simple integer division calculator. A command prompt window titled "Command Prompt - java Division" is also visible, showing the execution of the program. A small application window titled "INTEGER DIVISION OF TWO NUMBERS" is displayed, showing the result of dividing 12 by 4.

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class Division extends Frame implements ActionListener{
6     TextField n1,n2,res;
7     Label l1,l2,lres;
8     Button b;
9     public Division(){
10         setLayout(new FlowLayout());
11         l1=new Label("NUM1");
12         l2=new Label("NUM2");
13         lres=new Label("RESULT");
14         n1=new TextField(12);
15         n2=new TextField(8);
16         res=new TextField(10);
17         b=new Button("DIVIDE");
18         add(l1);
19         add(n1);
20         add(l2);
21         add(n2);
22         add(lres);
23         add(res);
24         b.addActionListener(this);
25         b.addActionListener(this);
26         addWindowListener(new MyWindowAdapter());
27     }
28     public void actionPerformed(ActionEvent ae){
29         if(ae.getSource()==b)
30         {
31             try{
32                 int num1=Integer.parseInt(n1.getText());
33                 int num2=Integer.parseInt(n2.getText());
34                 int num3=num1/num2;
35                 res.setText(String.valueOf(num3));
36             }catch(NumberFormatException ne ){
37                 JOptionPane.showMessageDialog(this,ne,"ERROR",JOptionPane.ERROR_MESSAGE);
38             }
39         }
40         catch(ArithmaticException a){
41             JOptionPane.showMessageDialog(this,a,"ERROR",JOptionPane.ERROR_MESSAGE);
42         }
43     }
44 }
```

Scanned with CamScanner

The screenshot shows a Windows desktop environment. In the center is a Java application window titled "INTEGER DIVISION OF TWO NUMBERS". It contains three text input fields: "NUM1" with value "12", "NUM2" with value "4", and a button labeled "DIVIDE". To the right of the button is a text field labeled "RESULT" with value "3". Above this window is a command prompt window titled "Command Prompt - java Division". It shows the output of running the Java compiler ("javac Division.java") and then executing the Java program ("java Division"). The Java code for the "Division" class is visible in the Notepad++ window on the left, which includes imports for AWT, Event, and Swing, and implements the ActionListener interface.

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class Division extends Frame implements ActionListener{
6     TextField n1,n2,res;
7     Label l1,l2,lres;
8     Button b;
9     public Division(){
10         setLayout(new FlowLayout());
11         l1=new Label("NUM1",Label.RIGHT);
12         l2=new Label("NUM2",Label.RIGHT);
13         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(l1);
19         add(n1);
20         add(l2);
21         add(n2);
22         add(b);
23         add(lres);
24         add(res);
25         b.addActionListener(this);
26         addWindowListener(new MyWindowAdapter());
27     }
28     public void actionPerformed(ActionEvent ae){
29         if(ae.getSource()==b)
30         {
31             try{
32                 int num1=Integer.parseInt(n1.getText());
33                 int num2=Integer.parseInt(n2.getText());
34                 int num3=num1/num2;
35                 res.setText(String.valueOf(num3));
36             }catch(NumberFormatException ne ){
37                 JOptionPane.showMessageDialog(this,ne,"ERROR",JOptionPane.ERROR_MESSAGE);
38             }
39         }
40         catch(ArithmaticException a){
41             JOptionPane.showMessageDialog(this,a,"ERROR",JOptionPane.ERROR_MESSAGE);
42         }
43     }
44 }
```

Scanned with CamScanner

The screenshot shows a Notepad++ window displaying a Java source file named Division.java. The code implements a simple integer division application with two text input fields and a result label. It includes exception handling for division by zero and invalid input.

```
18     add(l1);
19     add(n1);
20     add(l2);
21     add(n2);
22     add(b);
23     add(lres);
24     add(res);
25     b.addActionListener(this);
26     addWindowListener(new MyWindowAdapter());
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         {
39             JOptionPane.showMessageDialog(this, ne, "ERROR", JOptionPane.ERROR_MESSAGE);
40         }catch(ArithmaticException a)
41         {
42             JOptionPane.showMessageDialog(this, a, "ERROR", JOptionPane.ERROR_MESSAGE);
43         }
44     }
45     public static void main(String args[])
46     {
47         Division i = new Division();
48         i.setSize(new Dimension(50, 400));
49         i.setTitle("INTEGER DIVISION OF TWO NUMBERS");
50         i.setVisible(true);
51     }
52     class MyWindowAdapter extends WindowAdapter{
53         public void windowClosing(WindowEvent we)
54         {
55             System.exit(0);
56         }
57     }
58 }
59 }
```

Scanned with CamScanner

The screenshot shows a Windows desktop environment. In the foreground, the Notepad++ application is open, displaying a Java source file named `Division.java`. The code implements a `Frame` with a `FlowLayout` containing two `TextField`s for input numbers, a `Button` labeled "DIVIDE", and a `Label` for the result. It includes exception handling for division by zero.

```
1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4
5 public class Division extends Frame implements ActionListener{
6     TextField n1,n2,res;
7     Label l1,l2,lres;
8     Button b;
9     public Division(){
10         setLayout(new FlowLayout());
11         l1=new Label("NUM1");
12         l2=new Label("NUM2");
13         lres=new Label("RESULT");
14         n1=new TextField(13);
15         n2=new TextField(13);
16         res=new TextField(10);
17         b=new Button("DIVIDE");
18         add(l1);
19         add(n1);
20         add(l2);
21         add(n2);
22         add(b);
23         add(lres);
24         add(res);
25         b.addActionListener(this);
26         addWindowListener(new MyWindowAdapter());
27     }
28     public void actionPerformed(ActionEvent ae){
29         if(ae.getSource()==b)
30         {
31             try{
32                 int num1=Integer.parseInt(n1.getText());
33                 int num2=Integer.parseInt(n2.getText());
34                 int num3=num1/num2;
35                 res.setText(String.valueOf(num3));
36             }catch(NumberFormatException ne ){
37                 JOptionPane.showMessageDialog(this,ne,"ERROR",JOptionPane.ERROR_MESSAGE);
38             }
39         }catch(ArithmeticException a){
40             JOptionPane.showMessageDialog(this,a,"ERROR",JOptionPane.ERROR_MESSAGE);
41         }
42     }
43 }
```

In the background, a Command Prompt window titled "Command Prompt - java Division" is running on Microsoft Windows 10. The command `javac Division.java` was entered, followed by `java Division`, which executed the application. The application's window title is "INTEGER DIVISION OF TWO NUMBERS".

Scanned with CamScanner

