

LAB REPORT

NAME : NIRANJAN NAGARAJ SAVANUR

USN: 1BM19CS104

SUBJECT : OOJ

ACADEMIC YEAR : 2020-21

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.

SOURCE CODE:

```
import java.lang.Math;  
import java.util.Scanner;  
  
public class Quadratic{  
    public static void main(String args[]){  
        Scanner sc = new Scanner(System.in);  
        System.out.println("-----Finding roots of a quadratic equation-----");  
        System.out.println("Enter coefficients a, b, c of a QE in order w.r.t  
equation  $ax^2+bx+c=0$ ");  
        double r, sqrt;  
        double a = sc.nextDouble();  
        double b = sc.nextDouble();  
        double c = sc.nextDouble();
```

```
double disc = (Math.pow(b, 2)) - 4 * a * c;  
if(disc < 0){  
    sqrt = (Math.sqrt(-disc))/ (2 * a);  
    r = -b /(2*a);  
  
    System.out.println("Discriminant is negative. So no real roots are  
possible");  
    System.out.println("Imaginary roots are: " + r + " +i " + sqrt + " and  
" + r + " -i " + sqrt );  
    System.out.printf("or\n Imaginary roots are: %.2f +i %.4f and %.2f  
-i %.4f", r, sqrt,r,sqrt);  
}  
else if(disc > 0){  
    sqrt = (Math.sqrt(disc)) / (2 * a);  
    r = -b / (2 * a);  
    System.out.println("Real roots are: "+ (r+sqrt) + " and " + (r-sqrt));  
    System.out.printf("or\n Real roots are: %.4f and %.4f ", (r + sqrt),  
(r - sqrt));  
}
```

```
else if(disc == 0){  
    r = -b / (2 * a);  
    System.out.println("Roots are equal to: "+ r);  
}  
  
}  
}
```

OBSERVATION:

Name of the Experiment : WEEK 3
Experiment No. :

Date :
Page No. :

```
import java.lang.Math;  
import java.util.Scanner;  
public class Quadratic {  
    public static void main(String args[]) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println(" --- Finding root of a Q.E --- ");  
        System.out.println("Enter co-efficients a,b,c of the Q.E ");  
        double r, sqrt;  
        double a = sc.nextDouble();  
        double b = sc.nextDouble();  
        double c = sc.nextDouble();  
        double disc = (Math.pow(b, 2)) - 4 * a * c;  
        if (disc < 0) {  
            sqrt = (Math.sqrt(-disc)) / (2 * a);  
            r = -b / (2 * a);  
            System.out.println("ve D. So no real roots possible");  
            System.out.println(" Img roots are :" + r + "+i" + sqrt +  
                " and " + r + "-i" + sqrt);  
            // System.out.printf(" Img roots are %.2f+i%.2f & %.2f-i%.2f",  
            // r,sqrt,r,sqrt);  
        } else if (disc > 0) {  
            sqrt = (Math.sqrt(disc)) / (2 * a);  
            r = -b / (2 * a);  
            System.out.println(" Real roots :" + (r + sqrt) + " and " +  
                (r - sqrt));  
            // System.out.printf("Real roots are %.4f & %.4f", (r+sqrt),  
            // (r-sqrt));  
        }  
    }  
}
```



Name of the Experiment :
Experiment No. :

Date :
Page No. :

else if ($d_{disc} == 0$) {

$r = -b / (2 + a);$

System.out.println ("Roots are equal to " + r);

}

}

}

ALGORITHM

1. Input value of a, b, c

2. Calculate $d_{disc} = b^2 - 4ac$

3. IF ($d_{disc} < 0$)

a) calculate $\text{sqrt} = \sqrt{-d_{disc}} / 2 + a$

b) calculate $r = -b / 2a$

c) Display ^{imaginary} roots

Else
4. IF ($d_{disc} > 0$)

a) calculate $\text{sqrt} = \sqrt{d_{disc}} / 2a$

b) calculate $r = -b / 2a$

real and distinct

c) Display roots

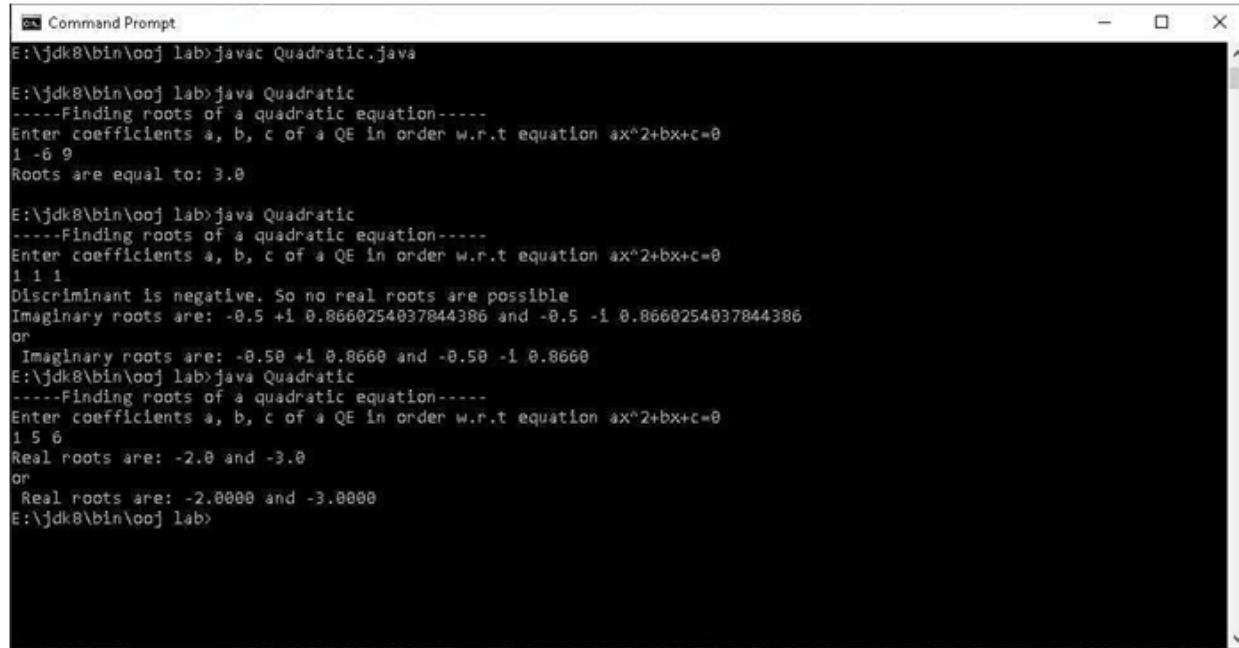
5. Else IF ($d_{disc} = 0$)

a) calculate $r = -b / 2a$

b) Display roots

real and equal

OUTPUT:



```
E:\jdk8\bin\oopj lab>javac Quadratic.java
E:\jdk8\bin\oopj lab>java Quadratic
----Finding roots of a quadratic equation----
Enter coefficients a, b, c of a QE in order w.r.t equation ax^2+bx+c=0
1 -6 9
Roots are equal to: 3.0

E:\jdk8\bin\oopj lab>java Quadratic
----Finding roots of a quadratic equation----
Enter coefficients a, b, c of a QE in order w.r.t equation ax^2+bx+c=0
1 1 1
Discriminant is negative. So no real roots are possible
Imaginary roots are: -0.5 +i 0.8660254037844386 and -0.5 -i 0.8660254037844386
or
Imaginary roots are: -0.50 +i 0.8660 and -0.50 -i 0.8660
E:\jdk8\bin\oopj lab>java Quadratic
----Finding roots of a quadratic equation----
Enter coefficients a, b, c of a QE in order w.r.t equation ax^2+bx+c=0
1 5 6
Real roots are: -2.0 and -3.0
or
Real roots are: -2.0000 and -3.0000
E:\jdk8\bin\oopj lab>
```

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.

SOURCE CODE:

```
import java.util.Scanner;

class Student{

    private String usn;

    private String name;

    private int[] credits = new int[20];

    private int[] marks = new int[20];

    private int n;

    void getDetails(){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter no of subjects:");

        n = sc.nextInt();

        System.out.println("Enter Student usn");

        usn = sc.next();

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

```
name = sc.next();

for(int i = 0; i < n; i++){
    System.out.println("Enter credits followed by marks for subject " +
(i+1) + ":" );
    credits[i] = sc.nextInt();
    marks[i] = sc.nextInt();
}

}
```

```
void printDetails(){
    System.out.println("Student details are as follows:");
    System.out.println("Name: " + name + "\tusn: " + usn);
    for(int i = 0; i < n; i++){
        System.out.println("Sub" + (i+1) + " Marks is: " + marks[i] +
"\tCredit is: " + credits[i]);
    }
}
```

```
void sgpaClac(){

double sgpa;

int[] gpcr = new int[n];

int credSum = 0, gp, sgpcr = 0;

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

credSum += credits[i];

if (marks[i] >= 90){

gp = 10;

}

else if(marks[i] >= 80){

gp = 9;

}

else if(marks[i] >= 70){

gp = 8;

}

else if(marks[i] >= 60){

gp = 7;

}

}
```

```
else if(marks[i] >= 50){  
    gp = 5;  
}  
  
else if(marks[i] >= 40){  
    gp = 4;  
}  
  
else{  
    gp = 0;  
}  
  
gpcr[i] = gp * credits[i];  
  
sgpcr += gpcr[i];  
  
}  
  
sgpa = sgpcr / (credSum + 0.0);  
  
System.out.println("Student's sgpa is: " + sgpa);  
}  
}  
  
public class Main {
```

```
public static void main(String[] args){  
    Student s1 = new Student();  
  
    s1.getDetails();  
    s1.printDetails();  
    s1.sgpacCalc();  
}  
}
```

OBSERVATION:

Name of the Experiment :
Experiment No. : WEEK 4

Date :
Page No. :

```
import java.util.Scanner;  
class student {  
    private String usn;  
    private String name;  
    private int[] credits = new int[20];  
    private int[] marks = new int[20];  
    private int n;
```

```
void getDetails(){  
    Scanner sc = new Scanner(System.in);  
    System.out.println("Enter no of subjects:");  
    n = sc.nextInt();  
    System.out.println("Enter student USN:");  
    usn = sc.next();  
    System.out.println("Enter student name");  
    name = sc.next();
```

```
for (int i=0; i<n; i++) {  
    System.out.println("Enter credits followed by marks  
        for subject " +(i+1) + ":");  
    credits[i] = sc.nextInt();  
    marks[i] = sc.nextInt();
```

y

3

Name of the Experiment :
Experiment No. :

Date :
Page No. : .

void printDetails() {

System.out.println("Student details are as follows:");

System.out.println("Name: " + name + " USN: " + usn);

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

System.out.println("Subject " + (i + 1) + " Mark is: " + marks[i])

+ " Credits: " + credits[i]);

}

}

void sgpaCalc() {

double sgpa;

int[] gpcr = new int[n];

int credSum = 0, gp, sgpcr = 0;

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

credSum += credits[i];

if (marks[i] >= 90) {

gp = 10;

}

else if (marks[i] >= 80) {

gp = 9;

}

else if (marks[i] >= 70) {

gp = 8;

}

Name of the Experiment :
Experiment No. :

Date :
Page No. :

else if (marks[3] >= 60) {

gp = 7;

}

else if (marks[3] >= 50) {

gp = 5;

}

else if (marks[3] >= 40) {

gp = 4;

}

else { gp = 0;

}

gpcr[3] = gp * credits[3];

sgpcr += gpcr[3];

y

sgpa = sgpcr / (reditsum + 0.0);

System.out.println ("Student sgpa is: " + sgpa);

y

public class Main {

public static void main (String [] args) {

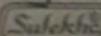
student sl = new student();

sl.getDetails();

sl.printDetails();

sl.sgpacalc();

y



OUTPUT:

CL Command Prompt

```
E:\jdk8\bin\ojj lab>javac Main.java
E:\jdk8\bin\ojj lab>java Main
Enter no of subjects:
5
Enter Student usn
1BM19CS104
Enter Student name
Niranjan
Enter credits followed by marks for subject 1:
3 98
Enter credits followed by marks for subject 2:
4 96
Enter credits followed by marks for subject 3:
5 83
Enter credits followed by marks for subject 4:
4 83
Enter credits followed by marks for subject 5:
4 94
Student details are as follows:
Name: Niranjan usn: 1BM19CS104
Sub1 Marks is: 98      Credit is: 3
Sub2 Marks is: 96      Credit is: 4
Sub3 Marks is: 83      Credit is: 5
Sub4 Marks is: 83      Credit is: 4
Sub5 Marks is: 94      Credit is: 4
Student's sgpa is: 9.55
E:\jdk8\bin\ojj lab>
```

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.

SOURCE CODE:

```
import java.util.Scanner;

class Book{

    private String author;
    private String name;
    private int num_pages;
    private double price;

    Scanner sc = new Scanner(System.in);

    Book(){}
    Book(String author, String name, int num_pages, double price){
        this.author = author;
```

```
this.name = name;  
this.num_pages = num_pages;  
this.price = price;  
}  
  
  
  
void getDetails(){  
    System.out.println("Enter author name");  
    author = sc.next();  
    System.out.println("Enter bookname");  
    name = sc.next();  
    System.out.println("Enter no of pages");  
    num_pages = sc.nextInt();  
    System.out.println("Enter price");  
    price = sc.nextDouble();  
}  
  
  
  
public String toString(){  
    return ("AUTHOR :" +author +"\nBOOK NAME :" +name +"\nPAGES  
:" +num_pages +"\nPRICE :" +price);  
}
```

```
}
```

```
}
```

```
class BookMain{  
    public static void main(String[] args){  
        Scanner sc = new Scanner(System.in);  
        int n;  
        System.out.println("ENTER THE NUMBER OF BOOKS");  
        n = sc.nextInt();  
        System.out.println("-----");  
        Book b[] = new Book[n];  
        b[0] = new Book("Tony", "Computers", 699, 2000.99 );  
        for(int i=1;i<n;i++){  
            System.out.println("ENTER DETAILS OF BOOK :" +(i+1));  
            b[i] = new Book();  
            b[i].getDetails();  
            System.out.println("-----");  
        }  
    }  
}
```

```
System.out.println("\n\nALL BOOK DETAILS THAT YOU  
ENTERED");  
  
System.out.println("-----");  
  
for(int i=0;i<n;i++){  
  
    System.out.println("\n*****");  
  
    System.out.println("BOOK :" +(i+1));  
  
    System.out.println(b[i]);  
  
}  
  
}  
  
}
```

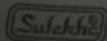
OBSERVATION:

Name of the Experiment :
Experiment No.

: WEEK-5

Date :
Page No.:

```
import java.util.Scanner;  
class Book{  
    private String author;  
    private String name;  
    private int num_pages;  
    private double price;  
    Scanner sc = new Scanner(System.in);  
    Book(){  
        Book(String author, String name, int num_pages,  
             double price){  
            this.author = author;  
            this.name = name;  
            this.num_pages = num_pages;  
            this.price = price;  
        }  
        void getDetails(){  
            System.out.println("Enter author name");  
            author = sc.next();  
            System.out.println("Enter bookname");  
            name = sc.next();  
            System.out.println("Enter no.of pages");  
            num_pages = sc.nextInt();  
            System.out.println("Enter price");  
            price = sc.nextDouble();  
        }  
}
```



King Size Practical Book

Signature.....

Name of the Experiment : Date :
Experiment No. : Page No. :

```
public String tostring() {
    return ("AUTHOR: " + author + "\nBOOK NAME: " + name
        + "\n PAGES: " + num_pages + "\n PRICE: " + price);
```

3

3

class BookMain

```
public static void main(String[] args) {
    Scanner sc = new Scanner (System.in);
    int n;
    System.out.println("ENTER NO OF BOOKS");
    n = sc.nextInt();
    System.out.println("-----");
    Book b[] = new Book[n];
    b[0] = new Book ("Tony", "Computers", 699, 2000.99);
    for (int i=1; i<n; i++) {
        System.out.println("Enter Book details of book " + (i+1));
        b[i] = new Book();
        b[i].getDetails();
        System.out.println("-----");
    }
    System.out.println("All books you entered");
    System.out.println("-----");
    for (int i=0; i<n; i++)
        System.out.println("\n*****");
```

4

System.out.println("All books you entered");

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

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

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

Signature.....

Name of the Experiment :
Experiment No. :

Date :
Page No. :

System.out.println ("Book :" + (g+1));

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

}

}

}

OUTPUT:

Command Prompt
E:\jdk8\bin\java BookMain
ENTER THE NUMBER OF BOOKS
3

ENTER DETAILS OF BOOK :2
Enter author name
cdvdd
Enter bookname
cx
Enter no of pages
345
Enter price
677.9

ENTER DETAILS OF BOOK :3
Enter author name
ccv
Enter bookname
cdvdsv
Enter no of pages
456
Enter price
78.9

ALL BOOK DETAILS THAT YOU ENTERED

BOOK :1
AUTHOR :Tony
BOOK NAME :Computers
PAGES :699
PRICE :2000.99

BOOK :2
AUTHOR :cdvdd
BOOK NAME :cx
PAGES :345
PRICE :677.9

First book details set
using
constructor

Command Prompt
cdvdd
Enter bookname
cx
Enter no of pages
345
Enter price
677.9

ENTER DETAILS OF BOOK :3
Enter author name
ccv
Enter bookname
cdvdsv
Enter no of pages
456
Enter price
78.9

ALL BOOK DETAILS THAT YOU ENTERED

BOOK :1
AUTHOR :Tony
BOOK NAME :Computers
PAGES :699
PRICE :2000.99

BOOK :2
AUTHOR :cdvdd
BOOK NAME :cx
PAGES :345
PRICE :677.9

BOOK :3
AUTHOR :ccv
BOOK NAME :cdvdsv
PAGES :456
PRICE :78.9

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.

SOURCE CODE:

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

```
import java.lang.Math;
```

```
abstract class Shape {
```

```
    int c, d;
```

```
    Shape(int a, int b) {
```

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

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

```
}
```

```
    Shape(int a) {
```

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

```
}
```

```
abstract void printArea();
```

```
}
```

```
class Rectangle extends Shape {
```

```
    Rectangle(int a, int b) {
```

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

```
}
```

```
    void printArea() {
```

```
        System.out.println("Area of rectangle is: " + c * d);
```

```
}
```

```
}
```

```
class Triangle extends Shape {
```

```
    Triangle(int a, int b) {
```

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

```
}
```

```
void printArea() {  
    System.out.println("Area of triangle is: " + c * d / 2);  
}  
}
```

```
class Circle extends Shape {  
    Circle(int a) {  
        super(a);  
    }
```

```
void printArea() {  
    System.out.println("Area of circle is: " + Math.PI * Math.pow(c, 2));  
}  
}
```

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

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

Shape s;

Rectangle r = new Rectangle(4, 10);

Triangle t = new Triangle(6, 3);

Circle c = new Circle(3);

s = r;

s.printArea();

s = t;

s.printArea();

s = c;

s.printArea();

}
```

OBSERVATION:

Name of the Experiment : WEEK-8-1
Experiment No. :

Date :
Page No.:

```
import java.util.Scanner;  
import java.lang.Math;  
abstract class Shape {  
    int c, d;  
    Shape (int a, int b) {  
        this.c = a;  
        this.d = b;  
    } Shape (int a) {  
        this.c = a;  
    }  
    abstract void printArea();  
}  
class Rectangle extends Shape {  
    Rectangle (int a, int b) {  
        super(a, b);  
    }  
    void printArea() {  
        System.out.println ("Area of rectangle is :" + (c*d));  
    }  
}  
class Triangle extends Shape {  
    Triangle (int a, int b) {  
        super(a, b);  
    }  
}
```

Signature.....

Name of the Experiment :
Experiment No. :

Date :
Page No. :

void printArea() {

System.out.println ("Area of triangle is :" + c * d / 2);

}

}

class Circle extends Shape {

Circle (int a) {

super (a);

}

void printArea() {

System.out.println ("Area of circle is :" + Math.PI *
Math.pow (c, 2)); } }

public class shapeMain {

public static void main (String [] args) {

Scanner sc = new Scanner (System.in);

Shape s;

System.out.println ("Enter rectangle dimensions :");

int len = sc.nextInt();

int br = sc.nextInt();

Rectangle r = new Rectangle (len, br);

System.out.println ("Enter triangle dimensions :");

int b = sc.nextInt();

int h = sc.nextInt();

Triangle t = new Triangle (b, h);

Name of the Experiment :
Experiment No. :

Date :
Page No. :

```
System.out.println("Enter circle radius");
int rad = sc.nextInt();
Circle c = new Circle(rad);
s = r;
s.printArea();
s = t;
s.printArea();
s = c;
s.printArea();
3
3
```

OUTPUT:

```
E:\jdk8\bin\oobj lab>java shapeMain
Enter dimensions of rectangle:
4 6
Enter dimensions of triangle:
6 3
Enter radius of circle:
3
Area of rectangle is: 24
Area of triangle is: 9
Area of circle is: 28.274333882308138
```

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 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.

SOURCE CODE:

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

```
abstract class Account {
```

```
String cName, accType;  
  
long accNo;  
  
double bal;  
  
final double minBal = 1000.0;
```

```
Account(String cName, long accNo, double bal, String accType) {  
    this.accNo = accNo;  
    this.cName = cName;  
    this.bal = bal;  
    this.accType = accType;  
}
```

```
abstract void addBal(double amt);
```

abstract void dispBal();

abstract void withBal(double amt);

}

```
class Curr_acct extends Account {  
    Curr_acct(String cName, long accNo, double bal) {  
        super(cName, accNo, bal, "Current");  
        System.out.println("name: " + cName + "\taccno: " + accNo +  
            "\tbl: " + bal + "\ttype: " + accType);  
    }  
}
```

```
void addBal(double amt) {  
    this.bal += amt;  
}
```

```
void dispBal() {  
    System.out.println("Your balance is: " + this.bal);  
}
```

```
void withBal(double amt) {  
    if (this.bal == 0 || amt > this.bal) {  
        System.out.println("withdrawal not possible");  
    }else{
```

```
this.bal -= amt;
```

checkBal();

}

}

```
void checkBal() {
```

```
if (this.bal < minBal) {
```

```
this.bal -= this.bal * 0.02;
```

}

}

}

```
class Sav_acct extends Account {
```

```
Sav_acct(String cName, long accNo, double bal) {
```

```
super(cName, accNo, bal, "Savings");
```

```
System.out.println("name: " + cName + "\taccno: " + accNo +  
"\tbl: " + bal + "\ttype: " + accType);
```

}

```
void addBal(double amt) {
```

```
this.bal += amt;  
addIntr();  
}
```

```
void addIntr() {  
this.bal += this.bal * 0.07;  
}
```

```
void dispBal() {  
System.out.println("Your balance is: " + this.bal);  
}
```

```
void withBal(double amt) {  
if (this.bal == 0 || amt > this.bal) {  
System.out.println("withdrawal not possible");  
}else{  
this.bal -= amt;  
}
```

}

}

```
class AccountMain {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        Double amt;  
        int flag = 0;  
        while (flag == 0) {  
            System.out.println("1:Current acc.\n2:Savings acc.\ndefault:exit");  
            int ch = sc.nextInt();  
            String nam;  
            long acno;  
            double balan;  
            switch (ch) {  
                case 1:  
                    System.out.println("Enter name, acc no, initial balance in order:");  
                    nam = sc.next();
```

```
acno = sc.nextLong();
balan = sc.nextDouble();
Curr_acct c = new Curr_acct(nam, acno, balan);
System.out.println("\nCurrent_acct\n");
int flag1 = 0;

while (flag1 == 0) {

    System.out.println("1:Addamount\n2:displayBalance\n3:withdraw\
        \ndefault:exit");

    int ch1 = sc.nextInt();

    switch (ch1) {

        case 1:

            System.out.println("enter amt to be added:");
            amt = sc.nextDouble();
            c.addBal(amt);

            break;

        case 2:

            c.dispBal();

            break;
    }
}
```

case 3:

```
System.out.println("enter amt to be withdrawn:");
amt = sc.nextDouble();
c.withBal(amt);
break;
```

default:

```
flag1 = 1;
```

```
}
```

```
}
```

```
break;
```

case 2:

```
System.out.println("\nSavings_acct\n");
System.out.println("Enter name, acc no, initial balance in order:");
nam = sc.next();
acno = sc.nextLong();
balan = sc.nextDouble();
Sav_acct s = new Sav_acct(nam, acno, balan);
```

```
int flag2 = 0;

while (flag2 == 0) {

    System.out.println("1:AddBal\n2:displayBal\n3:withdraw\ndefault:
exit");

    int ch2 = sc.nextInt();

    switch (ch2) {

        case 1:

            System.out.println("enter amt to be added:");
            amt = sc.nextDouble();
            s.addBal(amt);

            break;

        case 2:

            s.dispBal();

            break;

        case 3:

            System.out.println("enter amt to be withdrawn:");
            amt = sc.nextDouble();
            s.withBal(amt);

            break;
    }
}
```

```
break;
```

```
default:
```

```
flag2 = 1;
```

```
}
```

```
}
```

```
break;
```

```
default:
```

```
flag = 1;
```

```
}
```

```
}
```

```
}
```

```
}
```

OBSERVATION:

Name of the Experiment : WEEK-8-2
Experiment No. :

Date :
Page No. :

```
import java.util.Scanner;  
abstract class Account {  
    String cName, accType;  
    long accNo;  
    double bal;  
    final double minBal = 1000.0;  
    Account (String cName, long accNo, double bal,  
             String accType) {  
        this.accNo = accNo;  
        this.cName = cName;  
        this.bal = bal;  
        this.accType = accType;  
    }  
    abstract void addBal (double amt);  
    abstract void drpBal();  
    abstract void withBal (double amt);  
}  
  
class curr_acct extends Account {  
    curr_acct (String cName, long accNo, double bal) {  
        super (cName, accNo, bal, "Current");  
        System.out.println ("name:" + cName + "Acctno:" +  
                            accNo + "Bal:" + bal + "Type" + accType);  
    }  
}
```

Name of the Experiment :
Experiment No. :

Date :
Page No. :

void addBal (double amt) {

this. bal += amt;

}

void dropBal () {

System.out.println ("Your balance is: " + this. bal); }

void withdraw (double amt) {

if (this. bal == 0 || amt > this. bal) {

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

} else {

this. bal -= amt;

checkBal ();

}

}

void checkBal () {

if (this. bal < minBal) {

this. bal -= this. bal * 0.02; } }

class SavAcct extends Account {

SavAcct (String cName, long accNo, double bal) {

super (cName, accNo, bal, "savings");

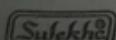
System.out.println ("name: " + cName + " | accno: " +
accNo + " | bal: " + bal + " | type: " + accType); }

void addBal (double amt) {

this. bal += amt;

addInts ();

}



Name of the Experiment :
Experiment No. :

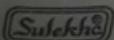
Date :
Page No. :

```
void addInt() {
    this.bal += this.bal * 0.07; }

void dispBal() { System.out.println("Balance:" + this.bal); }

void withdrawl(double amt) {
    if (this.bal == 0 || amt > this.bal) {
        System.out.println("withdrawal not possible");
    } else {
        this.bal -= amt;
    }
}

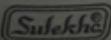
class AccountMain {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Double amt;
        int flag = 0;
        while (flag == 0) {
            System.out.println("1:Current 2:Saver 3:exit");
            int ch = sc.nextInt();
            String nam;
            long acno;
            double balan;
            switch (ch) {
                case 1: System.out.println("Enter name,acno,bal");
                    nam = sc.next();
                    acno = sc.nextLong();
                    balan = sc.nextDouble();
                    break;
                case 2: System.out.println("Enter name,acno,bal");
                    nam = sc.next();
                    acno = sc.nextLong();
                    balan = sc.nextDouble();
                    break;
                case 3: flag = 1;
            }
        }
    }
}
```



Name of the Experiment :
Experiment No. :

Date :
Page No. :

```
Curr-Accnt c = new Curr-Accnt (nam, acno, balan);  
int flag1 = 0;  
while (flag1 == 0) {  
    System.out.println ("1: add 2: display 3: withdraw  
    default: exit");  
    int ch1 = sc.nextInt();  
    switch (ch1) {  
        case 1: System.out.println ("enter amt to add");  
        amt = sc.nextDouble();  
        c.addBal (amt);  
        break;  
        case 2: c.displayBal (); break;  
        case 3: System.out.println ("ent amt to remove");  
        amt = sc.nextDouble();  
        c.withBal (amt);  
        break;  
        default: flag1 = 1; } } break;  
case 2:  
    System.out.println ("Enter name, acno and bal");  
    nam = sc.next();  
    acno = sc.nextLong();  
    balan = sc.nextDouble();  
    Sav-Accnt s = new Sav-Accnt (nam, acno, balan);  
    int flag2 = 0;
```



Name of the Experiment :
Experiment No. :

Date :
Page No. :

```
while(Flag2 == 0) {
    System.out.println("1:add 2:display 3:remove else:exit");
    int ch2 = sc.nextInt();
    switch(ch2) {
        case 1: System.out.println("enter amt to add");
                  amt = sc.nextDouble();
                  s.addBal(amt);
                  break;
        case 2: s.displayBal(); break;
        case 3: System.out.println("enter amount to remove");
                  amt = sc.nextDouble();
                  s.withBal(amt);
                  break;
        default: Flag2 = 1;
                  3
                  3
                  3
    }
}
```

Signature

OUTPUT:

```
E:\jdk8\bin\ojf lab>java AccountMain
1:Current acc.
2:Savings acc.
default:exit
1
Enter name, acc no, initial balance in order:
nakdkd 123456778 5000
name: nakdkd    accno: 123456778     bal: 5000.0      type: Current

Current_acct

1:Addamount
2:displayBalance
3:withdraw
default:exit
1
enter amt to be added:
1000
1:Addamount
2:displayBalance
3:withdraw
default:exit
2
Your balance is: 6000.0
1:Addamount
2:displayBalance
3:withdraw
default:exit
3
enter amt to be withdrawn:
1000
1:Addamount
2:displayBalance
3:withdraw
default:exit
2
Your balance is: 882.0
1:Addamount
2:displayBalance
3:withdraw
default:exit
1

default:exit
2
Your balance is: 882.0
1:Addamount
2:displayBalance
3:withdraw
default:exit
3
enter amt to be withdrawn:
100
1:Addamount
2:displayBalance
3:withdraw
default:exit
4
1:Current acc.
2:Savings acc.
default:exit
2

Savings_acct

Enter name, acc no, initial balance in order:
nasdr 12457467 5000
name: nasdr    accno: 12457467 bal: 5000.0      type: Savings
1:AddBal
2:displayBal
3:withdraw
default:exit
1
enter amt to be added:
1000
1:AddBal
2:displayBal
3:withdraw
default:exit
2
Your balance is: 6420.0
1:AddBal
2:displayBal
3:withdraw
default:exit
3
enter amt to be withdrawn:
```

```
Enter name, acc no, initial balance in order:  
nasdr 12457467 5000  
name: nasdr      accno: 12457467 bal: 5000.0      type: Savings  
1:AddBal  
2:displayBal  
3:withdraw  
default:exit  
1  
enter amt to be added:  
1000  
1:AddBal  
2:displayBal  
3:withdraw  
default:exit  
2  
Your balance is: 6420.0  
1:AddBal  
2:displayBal  
3:withdraw  
default:exit  
3  
enter amt to be withdrawn:  
4000  
1:AddBal  
2:displayBal  
3:withdraw  
default:exit  
2  
Your balance is: 2420.0  
1:AddBal  
2:displayBal  
3:withdraw  
default:exit  
4  
1:Current acc.  
2:Savings acc.  
default:exit  
4  
E:\jdk8\bin\oobj lab>
```

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.

SOURCE CODE:

```
import CIE.*;
import SEE.*;
import java.util.Scanner;

class finalmarks {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        System.out.println("ENTER THE NUMBER OF STUDENTS");
        int n = s.nextInt();
        SEE.Externals ob1[] = new SEE.Externals[n];
        CIE.Internals ob[] = new CIE.Internals[n];
```

```
for (int i = 0; i < n; i++) {  
  
    System.out.println("\nENTER THE USN,NAME AND SEMESTER  
    OF STUDENT" + (i + 1));  
  
    String u = s.next();  
  
    String nm = s.next();  
  
    int se = s.nextInt();  
  
    ob1[i] = new SEE.Externals(u, nm, se);  
  
  
  
    ob[i] = new CIE.Internals();  
  
    ob[i].get();  
  
    ob1[i].get();  
  
}  
  
for (int i = 0; i < n; i++) {  
  
    System.out.println("\nSTUDENT_" + (i + 1));  
  
    System.out.println("name = " + ob1[i].name + "\tusn = " +  
        ob1[i].usn + "\tsem = " + ob1[i].sem);  
  
    for (int j = 0; j < 5; j++) {  
  
        System.out.println("SUBJECT_" + (j + 1) + "_MARKS = " +  
            (ob[i].cie[j] + (ob1[i].see[j])/2));  
  
    }  
}
```

```
}
```

```
}
```

PACKAGE SEE:-

```
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 < 5; i++) {
```

```
            System.out.println("ENTER THE SEE MARK IN SUBJECT " + (i + 1));
```

```
            see[i] = s.nextInt();
```

```
}
```

```
}
```

```
}
```

PACKAGE CIE:-

```
package CIE;
```

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

```
public class Internals extends CIE.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 THE CIE MARK IN SUBJECT" + (i + 1));
```

```
cie[i] = s.nextInt();
```

```
}
```

```
}
```

```
}
```

```
package CIE;
```

```
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;
```

```
}
```

```
}
```

OBSERVATION:

}

Procedure used :-

- 1) Created CIE and SEE folders
- 2) In CIE folder created two files Student.java and Internals.java which are public class with public methods.
- 3) In SEE folder created ~~two~~^{one} file Externals.java a public class with public method.
- 4) Compiled each files separately within package
- 5) From main folder, compiled the driver class
- 6) Executed main driver file from the pack.

Name of the Experiment : WEEK-9
Experiment No. :

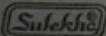
Date :
Page No.:

CIE / student.java

```
package CIE;
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;
    }
}
```

CIE / Internals.java

```
package CIE;
import java.util.Scanner;
public class Internals extends CIE.Student {
    Scanner s = new Scanner (System.in);
    public int [] mre = new int [5];
    public void get () {
        for (int i=0; i<5; i++) {
            System.out.println ("Enter CIE mark in sub " + (i+1));
            mre[i] = s.nextInt ();
        }
    }
}
```



King Size Practical Book

Signature.....

Name of the Experiment :
Experiment No. :

Date :
Page No. :

SEE / Externals.java

```
package SEE;  
import java.util.Scanner;  
public class Externals extends CIE.Student {  
    public Externals(String urn, String name, int sem){  
        super(urn, name, sem);  
    }  
    Scanner s = new Scanner(System.in);  
    public int [] see = new int [5];  
    public void get () {  
        for (int i=0; i<5; i++) {  
            System.out.println("Enter see in sub "+(i+1));  
            see[i] = s.nextInt();  
        }  
    }  
}
```

Finalmarks.java

```
import CIE.*;
import SEE.*;

class Finalmarks {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter no of students:");
        int n = s.nextInt();
        SEE.Externals ob[] = new SEE.Externals[n];
        CIE.Internals ob[] = new CIE.Internals[n];
        for (int i=0; i<n; i++) {
            System.out.println("Enter UN, name & sem for
student " + (i+1));
            String u = s.next();
            String nm = s.next();
            int se = s.nextInt();
            ob[i] = new SEE.Externals(u, nm, se);
            ob[i] = new CIE.Internals();
            ob[i].get();
            ob[i].get();
        }
        for (int i=0; i<n; i++) {
            System.out.println("Student " + (i+1));
        }
    }
}
```

Name of the Experiment :	Date :
Experiment No. :	Page No.:
<pre> System.out.println("name = " + ob[0].name + " usn = " + ob[0].usn + " sem = " + ob[0].sem); for (int g = 0; g < 5; g++) { System.out.println("SUB " + (g + 1) + " Marks" + (ob[1].cie[g] + (ob[1].see[g]) / 2)); } } </pre>	

OUTPUT:

```

E:\jdk8\bin>java finalmarks
ENTER THE NUMBER OF STUDENTS
2
ENTER THE USN,NAME AND SEMESTER OF STUDENT1
10m19cs104 niranjan 3
ENTER THE CIE MARK IN SUBJECT1
45
ENTER THE CIE MARK IN SUBJECT2
43
ENTER THE CIE MARK IN SUBJECT3
34
ENTER THE CIE MARK IN SUBJECT4
21
ENTER THE CIE MARK IN SUBJECTS
40
ENTER THE SEE MARK IN SUBJECT1
78
ENTER THE SEE MARK IN SUBJECT2
76
ENTER THE SEE MARK IN SUBJECT3
87
ENTER THE SEE MARK IN SUBJECT4
67
ENTER THE SEE MARK IN SUBJECTS
67

ENTER THE USN,NAME AND SEMESTER OF STUDENT2
10m19cs106 nitin 3
ENTER THE CIE MARK IN SUBJECT1
43
ENTER THE CIE MARK IN SUBJECT2
45
ENTER THE CIE MARK IN SUBJECT3
50
ENTER THE CIE MARK IN SUBJECT4
34
ENTER THE CIE MARK IN SUBJECTS
45
ENTER THE SEE MARK IN SUBJECT1
78
ENTER THE SEE MARK IN SUBJECT2

```

```
67
ENTER THE USN,NAME AND SEMESTER OF STUDENT2
ibm19cs106 nitin 3
ENTER THE CIE MARK IN SUBJECT1
43
ENTER THE CIE MARK IN SUBJECT2
45
ENTER THE CIE MARK IN SUBJECT3
50
ENTER THE CIE MARK IN SUBJECT4
34
ENTER THE SEE MARK IN SUBJECT1
45
ENTER THE SEE MARK IN SUBJECT2
78
ENTER THE SEE MARK IN SUBJECT3
98
ENTER THE SEE MARK IN SUBJECT4
87
ENTER THE SEE MARK IN SUBJECT5
67
ENTER THE SEE MARK IN SUBJECTS
56

STUDENT_1
name = niranjan usn = ibm19cs104      sem = 3
SUBJECT_1_MARKS = 84
SUBJECT_2_MARKS = 92
SUBJECT_3_MARKS = 77
SUBJECT_4_MARKS = 54
SUBJECT_5_MARKS = 73

STUDENT_2
name = nitin   usn = ibm19cs106      sem = 3
SUBJECT_1_MARKS = 82
SUBJECT_2_MARKS = 94
SUBJECT_3_MARKS = 93
SUBJECT_4_MARKS = 67
SUBJECT_5_MARKS = 73

E:\jdk8\bin>
```

LAB PROGRAM 7

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

SOURCE CODE:

```
class Gen<T, U, V> {
```

```
    T ob1;
```

```
    U ob2;
```

```
    V ob3;
```

```
    Gen(T o1, U o2, V o3) {
```

```
        ob1 = o1;
```

```
        ob2 = o2;
```

```
        ob3 = o3;
```

```
}
```

```
    void showTypes() {
```

```
        System.out.println("Type of T object is " +  
            ob1.getClass().getName());
```

```
System.out.println("Type of U object is " +  
ob2.getClass().getName());  
  
System.out.println("Type of V object is " +  
ob3.getClass().getName());  
}
```

```
T getob1() {  
return ob1;  
}
```

```
U getob2() {  
return ob2;  
}
```

```
V getob3() {  
return ob3;  
}  
}
```

```
class Generics {  
    public static void main(String args[]) {  
        Gen<Integer, Double, String> genOb = new Gen<Integer, Double,  
String>(15, 99.457, "Niranjan");  
  
        genOb.showTypes();  
  
        int t = genOb.getob1();  
        System.out.println("value in T: " + t);  
        Double u = genOb.getob2();  
        System.out.println("value in U: " + u);  
        String v = genOb.getob3();  
        System.out.println("value in V: " + v);  
    }  
}
```

OBSERVATION:

Name of the Experiment : Date :
Experiment No. : WEEK - 10 - LAB - 7 Page No. :

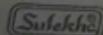
```
class Gen<T,U,V> {
    T ob1;
    U ob2;
    V ob3;
    Gen<T,U,V> o1, U o2, V o3);
    ob1 = o1;
    ob2 = o2;
    ob3 = o3;
}

void showType() {
    System.out.println("Type of T is " + ob1.getClass().getName());
    System.out.println("Type of U is " + ob2.getClass().getName());
    System.out.println("Type of V is " + ob3.getClass().getName());
}

T getob1() {
    return ob1;
}

U getob2() {
    return ob2;
}

V getob3() {
    return ob3;
}
```



Name of the Experiment :
Experiment No. :

Date :
Page No.:

class Phenotypes {

```
    public static void main (String args[]) {  
        Gen< Integer, Double, String> genOb = new Gen< Integer,  
            Double, String> (15, 99.457, "Ngrangan");  
        genOb.showTypes();
```

```
        int t = genOb.getOb1();
```

```
        System.out.println ("value in T " + t);
```

```
        double u = genOb.getOb2();
```

```
        System.out.println ("value in U " + u);
```

```
        String v = genOb.getOb3();
```

```
        System.out.println ("value in V " + v);
```

```
    }
```

```
}
```

OUTPUT:

```
E:\jdk8\bin\oobj lab>java Generics
Type of T object is java.lang.Integer
Type of U object is java.lang.Double
Type of V object is java.lang.String
value in T: 15
value in U: 99.457
value in V: Niranjan
```

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>=father’s age.

SOURCE CODE:

```
import java.util.Scanner;

class WrongAge extends Exception {

    int age;

    WrongAge(int x) {

        age = x;

    }

    public String toString() {

        return "AGE OF FATHER=" + age + " IS ENTERED  
INCORRECTLY";

    }

}

class WrongAgeSon extends Exception {
```

```
int age;  
  
WrongAgeSon(int x) {  
  
    age = x;  
  
}  
  
public String toString() {  
  
    return "AGE OF SON=" + age + " IS ENTERED INCORRECTLY";  
  
}  
  
}
```

```
class Father {  
  
    int a;  
  
    Father(int x) {  
  
        a = x;  
  
    }  
  
    void check() throws WrongAge {  
  
        if (a<0) {  
  
            throw new WrongAge(a);  
  
        }  
  
    }  
}
```

```
}
```

```
class Son extends Father {  
    int age;  
    Son(int fage,int sage) {  
        super(fage);  
        age = sage;  
    }  
    void compute() throws WrongAgeSon {  
        if (age >= a) {  
            throw new WrongAgeSon(age);  
        } else {  
            System.out.println("THE AGES ARE ENTERED CORECTLY");  
            System.out.println("FATHER'S AGE=" + a + "\t" + "SON'S AGE=" + age);  
        }  
    }  
}  
class ExceptionsMain1 {  
    public static void main(String args[]) {
```

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

System.out.println("ENTER FATHER'S AGE:");

int f = s.nextInt();

System.out.println("ENTER SON'S AGE:");

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);

}

}

}
```

OBSERVATION:

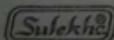
Name of the Experiment : Date :
Experiment No. : WEEK-#0 LAB-8 NEW Page No.:

```
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);  
        }  
    }  
}
```

Name of the Experiment :
Experiment No. :

Date :
Page No. :

```
class Son extends Father {  
    int age;  
    Son(int page, int sage) {  
        super(page);  
        age = sage;  
    } void compute() throws WrongAgeSon {  
        if (age <= 0) {  
            throw new WrongAgeSon(age);  
        } else {  
            System.out.println("CORRECT " + sage + " " + page);  
        }  
    }  
}  
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); }  
    }  
}
```



OUTPUT:

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

SOURCE CODE:

```
class Thread1 implements Runnable {
```

```
    String name;
```

```
    Thread t;
```

```
    int time;
```

```
    Thread1(String threadname, int time) {
```

```
        name = threadname;
```

```
        this.time = time;
```

```
        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 ThreadMain1 {  
    public static void main(String args[]) {  
        Thread1 t1 = new Thread1("BMS COLLEGE OF ENGINEERING",  
        10000);  
        Thread1 t2 = new Thread1("CSE", 2000);  
    }  
}
```

OBSERVATION:

Name of the Experiment : Week-II
Experiment No. :

Date :
Page No. :

class Thread1 implements Runnable {

String name;

Thread t;

int time;

Thread1(String threadname, int time) {

name = threadname;

this.time = time;

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");

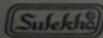
}

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

3

3

3



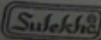
King Size Practical Book

Signature.....

Name of the Experiment :
Experiment No. :

Date :
Page No. :

```
class ThreadMain1 {  
    public static void main (String args[]) {  
        Thread1 t1 = new Thread1 ("BMS COLLEGE OF  
        ENGINEERING", 10000);  
        Thread1 t2 = new Thread1 ("CSE", 2000);  
    }  
}
```



Sulekha King Size Practical Book

Signature.....

OUTPUT:

```
PS E:\jdk8\bin\ojof lab> javac ThreadMain1.java
PS E:\jdk8\bin\ojof lab> java ThreadMain1
thread:Thread[BMS COLLEGE OF ENGINEERING,5,main]
thread:Thread[CSE,5,main]
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
CSE
CSE
BMS COLLEGE OF ENGINEERING
CSE exiting.
BMS COLLEGE OF ENGINEERING
BMS COLLEGE OF ENGINEERING
BMS COLLEGE OF ENGINEERING
BMS COLLEGE OF ENGINEERING
BMS COLLEGE OF ENGINEERING exiting.
PS E:\jdk8\bin\ojof lab> [ ]
```

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 Arithmetic Exception Display the exception in a message dialog box.

SOURCE CODE:

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

class dia extends Dialog implements ActionListener {
    integerdivision1 id;

    dia(Frame parent, String title) {
        super(parent, title, false);
        id = (integerdivision1) parent;
        setLayout(new FlowLayout());
        setSize(300, 200);
```

```
add(new Label(id.msg));  
Button b;  
add(b = new Button("OK"));  
b.addActionListener(this);  
}  
  
}
```

```
public void actionPerformed(ActionEvent ae) {  
    dispose();  
}  
}
```

```
public class integerdivision1 extends Frame implements  
ActionListener {  
    String msg = "";  
    TextField n1, n2, res;  
    Label ln1, ln2, lres;  
    Button b;
```

```
public integerdivision1() {  
    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 TextField(12);
n2 = new TextField(8);
res = new TextField(10);
b = new Button("DIVIDE");
add(ln1);
add(n1);
add(ln2);
add(n2);
add(b);
add(lres);
add(res);
b.addActionListener(this);
addWindowListener(new WindowAdapter1());
}

public void actionPerformed(ActionEvent ae)
```

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

```
msg="NUMBERFORMAT EXCEPTION";  
dia d=new dia(this,"EXCEPTION");  
d.setVisible(true);
```

```
}
```

```
catch(ArithmeticException a){
```

```
msg="ARITHMETIC EXCEPTION";  
dia d=new dia(this,"EXCEPTION");
```

```
d.setVisible(true);

}

}

}

public static void main(String args[]) {
    integerdivision1 i = new integerdivision1();
    i.setSize(new Dimension(300, 300));
    i.setTitle("INTEGER DIVISION OF TWO NUMBERS");
    i.setVisible(true);
}

class WindowAdapter1 extends WindowAdapter {
    public void windowClosing(WindowEvent we) {
        System.exit(0);
    }
}
```

}

OBSERVATION:

Name of the Experiment : Week13_Lab10
Experiment No. :

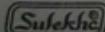
Date :
Page No. :

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class IntegerDivision extends Frame implements
    ActionListener {
    TextField n1, n2, res;
    Label ln1, ln2, lres;
    Button b;
    public IntegerDivision() {
        setLayout(new FlowLayout());
        Label ln1 = new Label("NUMBER1", Label.RIGHT);
        Label ln2 = new Label("NUMBER2", Label.RIGHT);
        Label lres = new Label("RESULT", Label.RIGHT);
        n1 = new TextField(12);
        n2 = new TextField(8);
        res = new TextField(10);
        b = new Button("DIVIDE");
        add(ln1);
        add(n1);
        add(ln2);
        add(n2);
        add(b);
        add(lres);
        add(res);
    }
}
```

Name of the Experiment :
Experiment No. :

Date :
Page No. :

```
b. addActionListener(this);  
addWindowListener(new WindowAdapter() {  
});  
public void actionPerformed(ActionEvent ae) {  
    if (ae.getSource() == b)  
    {  
        try {  
            int num1 = Integer.parseInt(t1.getText());  
            int num2 = Integer.parseInt(t2.getText());  
            int num3 = num1 / num2;  
            res.setText(String.valueOf(num3));  
        } catch (NumberFormatException ne) {  
            JOptionPane.showMessageDialog(this, ne, "ERROR",  
                JOptionPane.ERROR_MESSAGE);  
        }  
        catch (ArithmeticException 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);  
}
```



Name of the Experiment :
Experiment No. :

Date :
Page No.:

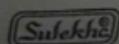
class WindowAdapter1 extends WindowAdapter {
public void windowClosing(WindowEvent we) {
System.exit(0);

3

3

3.

Signature.....

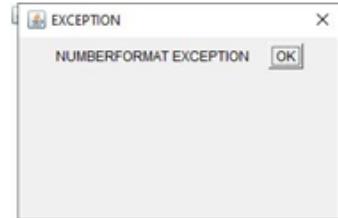


King Size Practical Book

OUTPUT:

INTEGER DIVISION OF TWO NUMBERS

NUMBER 1 NUMBER 2 DIVIDE



NUMBER 1 NUMBER 2 DIVIDE



NUMBER 1 NUMBER 2 DIVIDE