

LAB RECORD (OOJ LAB 2020)

NAME: TRISHA LAKHANI

USN:1BM19CS214

SECTION:3 D

ACADEMIC YEAR: AUG-JAN 2020

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

Lab Program

① Quadratic equation

```
import java.util.Scanner;  
class quadratic {  
    public static void main (String [] args) {  
        int a, b, c;
```

```
        double d, r1, r2;
```

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

```
        System.out.println ("Enter the values of a, b, c");
```

```
        a = in.nextInt();
```

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

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

```
        d = (b * b) - (4 * a * c);
```

```
        if (d > 0)
```

```
{
```

```
        System.out.println ("Roots are real and unequal");
```

```
        r1 = (-b + Math.sqrt(d)) / (2 * a);
```

```
        r2 = (-b - Math.sqrt(d)) / (2 * a);
```

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

```
}
```

```
else if (d == 0)
```

```
{
```

```
        System.out.println ("Roots are real and equal");
```

```
        r1 = r2 = -b / (2 * a);
```

```
        System.out.println ("The root is " + r1);
```

```
}
```

```
else {
```

```
        System.out.println ("Roots are imaginary");
```

```
}
```

```
}
```

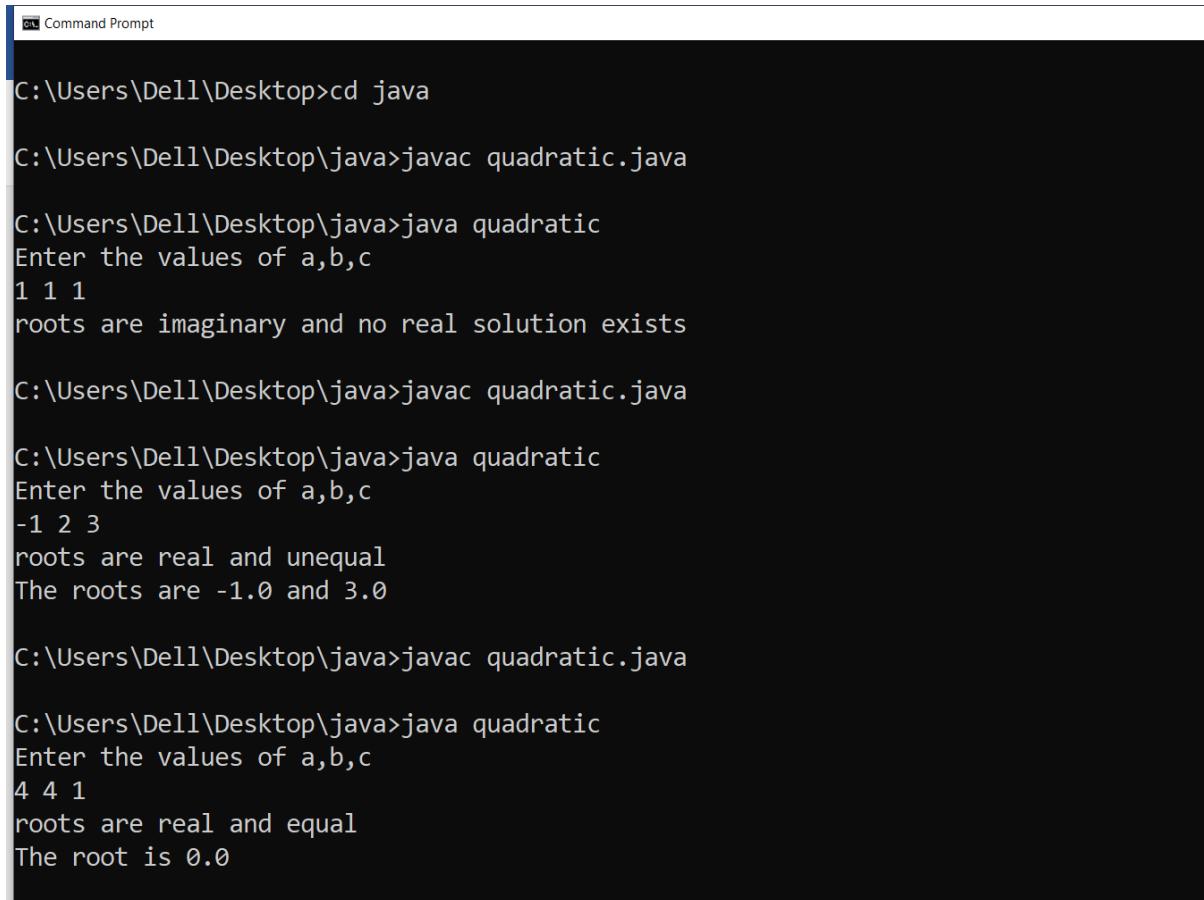
```
import java.util.Scanner;

public class quadratic{
    public static void main(String[] args){
        int a,b,c;
        double d,r1,r2;

        Scanner in=new Scanner(System.in);
        System.out.println("Enter the values of a,b,c");
        a=in.nextInt();
        b=in.nextInt();
        c=in.nextInt();
        d=(b*b)-(4*a*c);
        if(d>0)
        {
            System.out.println("roots are real and unequal");
            r1=(-b+Math.sqrt(d))/(2*a);
            r2=(-b-Math.sqrt(d))/(2*a);
            System.out.println("The roots are " + r1 + " and " + r2);
        }
        else if(d == 0)
        {
            System.out.println("roots are real and equal");
            r1 = r2 = -b / (2 * a);
            System.out.println("The root is " + r1);
        }
    }
}
```

```
{  
    System.out.println("roots are imaginary and no real solution exists");  
  
}  
  
}  
}
```

OUTPUT



```
Command Prompt  
C:\Users\Dell\Desktop>cd java  
C:\Users\Dell\Desktop\java>javac quadratic.java  
C:\Users\Dell\Desktop\java>java quadratic  
Enter the values of a,b,c  
1 1 1  
roots are imaginary and no real solution exists  
  
C:\Users\Dell\Desktop\java>javac quadratic.java  
C:\Users\Dell\Desktop\java>java quadratic  
Enter the values of a,b,c  
-1 2 3  
roots are real and unequal  
The roots are -1.0 and 3.0  
  
C:\Users\Dell\Desktop\java>javac quadratic.java  
C:\Users\Dell\Desktop\java>java quadratic  
Enter the values of a,b,c  
4 4 1  
roots are real and equal  
The root is 0.0
```

Lab Program 2: Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

Trisha Lekhani
1B19CS214

(2)

Student SGPA

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

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

    void showDetails () {
        System.out.println ("Student USN : " + USN);
        System.out.println ("Student Name : " + Name);
        for (int i=0; i<5; i++) {
            System.out.println ("Subject " + (i+1) + " - Credits : " + credits [i] +
                " and Marks : " + marks [i]);
        }
    }
}
```

```
System.out.println (" SGPA of " + Name + " is: " + (float)(SGPA/totalCredi  
} s));  
  
void calcSGPA() {  
  
    for (int i= 0; i<5; i++) {  
        if (marks[i] > 100) {  
            System.out.println (" error: Marks are above 100");  
            return;  
        }  
        else if (marks[i] >= 90) {  
            points[i] = 10;  
        }  
        else if (marks[i] >= 80) {  
            points[i] = 9; }  
        else if (marks[i] >= 70) {  
            points[i] = 8; }  
        } else if (marks[i] >= 60) {  
            points[i] = 7; }  
        } else if (marks[i] >= 50) {  
            points[i] = 6; }  
        } else if (marks[i] >= 40) {  
            points[i] = 5; }  
        } else {  
            points[i] = 0; }  
        SGPA += (points[i] * credits[i]);  
    }  
}
```

```
public class Lab2 {  
    public static void main (String args []) {  
        Student1 stu1 = new Student1 ();  
        stu1. get Details ();  
        stu1. calc SGPA ();  
        stu1. show Details ();
```

3

}

```
import java.util.Scanner;

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

    void getDetails(){
        System.out.println("Enter student USN: ");
        USN = in.nextLine();
        System.out.println("Enter student Name: ");
        Name = in.nextLine();
        for(int i=0;i<5;i++){
            System.out.println("Enter Credits for Subject " + (i+1) + ": ");
            credits[i] = in.nextInt();
            totalCredits += credits[i];
            System.out.println("Enter Marks for Subject " + (i+1) + ": ");
            marks[i] = in.nextFloat();
        }
    }

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

```

System.out.println("Student name: " + Name);
for(int i=0;i<5;i++){
    System.out.println("Subject " + (i+1) + " - Credits: " + credits[i] + " and Marks: " + marks[i]);
}
System.out.println("SGPA of " + Name + " is: " + (float)(SGPA/totalCredits));
}

void calcSGPA(){
    for(int i = 0;i<5;i++){
        if(marks[i] > 100){
            System.out.println("Error: Marks are above 100");
            return;
        }else if(marks[i] >= 90){
            gradepoints[i] = 10;
        }else if(marks[i] >= 80){
            gradepoints[i] = 9;
        }else if(marks[i] >= 70){
            gradepoints[i] = 8;
        }else if(marks[i] >= 60){
            gradepoints[i] = 7;
        }else if(marks[i] >= 50){
            gradepoints[i] = 6;
        }else if(marks[i] >= 40){
            gradepoints[i] = 5;
        }else{
            gradepoints[i] = 0;
        }
    }
}

```

```
    SGPA = SGPA+(gradepoints[i]*credits[i]);  
}  
}  
  
}
```

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

OUTPUT

```
C:\Users\DELL\Desktop>javac Lab2.java
C:\Users\DELL\Desktop>java Lab2
Enter student USN:
1bm19cs214
Enter student Name:
trisha
Enter Credits for Subject 1:
2
Enter Marks for Subject 1:
70
Enter Credits for Subject 2:
4
Enter Marks for Subject 2:
80
Enter Credits for Subject 3:
4
Enter Marks for Subject 3:
75
Enter Credits for Subject 4:
3
Enter Marks for Subject 4:
90
Enter Credits for Subject 5:
4
Enter Marks for Subject 5:
89
Student USN: 1bm19cs214
Student name: trisha
Subject 1 - Credits: 2 and Marks: 70.0
Subject 2 - Credits: 4 and Marks: 80.0
Subject 3 - Credits: 4 and Marks: 75.0
Subject 4 - Credits: 3 and Marks: 90.0
Subject 5 - Credits: 4 and Marks: 89.0
SGPA of trisha is: 8.823529
C:\Users\DELL\Desktop>java Lab2
```

LAB PROGRAM 3: Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

Lab program: Books

③ import java.util.Scanner;

class Book

{

 public String name;

 public String author;

 public double price;

 public int num-pages;

 Book()

{

 name = "xyz";

 author = "abc";

 price = 0.0;

 num-pages = 0;

}

 void getdata()

{

 Scanner sc = new Scanner(System.in);

 System.out.println("enter the name of the book");

 name = sc.nextLine();

 System.out.println("enter the price of the book");

 price = sc.nextDouble();

 System.out.println("enter the author of book");

 author = sc.nextLine();

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

 num-pages = sc.nextInt();

}

 public String toString()

 return ("Book: " + name + "\nAuthor: " + author + "\nPrice: Rs" +

 price + "\nNo. of pages: " + num-pages);

}

}

```
class lab4
{
    public static void main (String ss[])
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter the number of books :");
        int n= sc.nextInt ();
        Book b[] = new Book[n];
        System.out.println ("Enter the book details");
        for (int i=0; i<n; i++)
        {
            System.out.println ("Book " + (i+1));
            b[i]= new Book();
            b[i].getdata();
        }
        System.out.println ("Book Details");
        for (int i=0; i<n; i++)
        {
            System.out.println ("Book " + (i+1));
            System.out.println (b[i]);
            System.out.println ("-----");
        }
    }
}
```

```
import java.util.*;
class Book
{
    public String name;
    public String author;
    public double price;
    public int num_pages;

    Book()
    {
        name="xyz";
        author="abc";
        price= 0.0;
        num_pages=0;
    }

    void getdata()
    {
        Scanner sc= new Scanner(System.in);
        System.out.println("enter the name of book");
        name = sc.nextLine();
        System.out.println("enter the name of author");
        author = sc.nextLine();
        System.out.println("enter the price of book");
        price = sc.nextDouble();
        System.out.println("enter the number of pages");
        num_pages = sc.nextInt();
    }

    public String toString()
```

```
{  
    return("Book: "+name+"\nAuthor: "+author+"\nPrice: Rs "+price+"\nNo.of pages:  
"+num_pages);  
}  
}  
  
class lab3  
{  
    public static void main(String ss[])  
    {  
        Scanner sc=new Scanner(System.in);  
        System.out.println("Enter the number of books:");  
        int n=sc.nextInt();  
        Book b[]=new Book[n];  
        int i;  
        System.out.println("Enter the book details");  
        for(i=0;i<n;i++)  
        {  
            System.out.println("Book "+(i+1));  
            b[i]=new Book();  
            b[i].getdata();  
        }  
        System.out.println(" Book details....");  
        for(i=0;i<n;i++)  
        {  
            System.out.println("Book "+(i+1));  
            System.out.println(b[i]);  
            System.out.println("-----");  
        }  
    }  
}
```

```
 }  
 }
```

OUTPUT

```
 Command Prompt  
Enter the number of books:  
2  
Enter the book details  
Book 1  
enter the name of book  
math  
enter the name of author  
abc  
enter the price of book  
400  
enter the number of pages  
200  
Book 2  
enter the name of book  
evs  
enter the name of author  
xyz  
enter the price of book  
300  
enter the number of pages  
150  
Book details....  
Book 1  
Book: math  
Author: abc  
Price: Rs 400.0  
No.of pages: 200
```

```
 Book details....  
Book 1  
Book: math  
Author: abc  
Price: Rs 400.0  
No.of pages: 200  
-----  
Book 2  
Book: evs  
Author: xyz  
Price: Rs 300.0  
No.of pages: 150  
-----
```

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.

Tousha
16M1AC5214.

Lab Program 4: Shape.

```
import java.util.*;
abstract class Shape
{
    int a;
    int b;
    abstract void printArea();
}

class Rectangle extends Shape
{
    Rectangle ( int x, int y )
    {
        a = x;
        b = y;
    }

    void printArea()
    {
        System.out.println (" Area is " + (a*b));
    }
}

class Triangle extends Shape
{
    Triangle ( int x, int y )
    {
        a = x;
        b = y;
    }

    void printArea()
    {
        System.out.println (" Area is " + (a*b * 0.5));
    }
}
```

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

```
    void printArea()  
{  
    System.out.println("Area is " + (a * a * 3.14));  
}
```

```
class lab4  
{  
    public static void main ( String args [] )  
{
```

```
    int l, b, ba, h, ra;
```

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

```
    System.out.println("Enter the length and  
breadth of rectangle");
```

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

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

```
    Rectangle r = new Rectangle (l, b);
```

```
    r.printArea();
```

```
System.out.println("Enter the base and height of triangle");
```

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

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

```
Triangle t = new Triangle (ba, h);
```

```
t.printArea();
```

```
System.out.println("Enter the radius of circle");
```

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

```
Circle c = new Circle (ra);
```

```
}
```

```
import java.util.*;

abstract class Shape

{

    int a;

    int b;

    abstract void printArea();

}

class Rectangle extends Shape

{

    Rectangle(int x, int y)

    {

        a=x;

        b=y;

    }

    void printArea()

    {

        System.out.println("Area is "+(a*b));

    }

}

class Triangle extends Shape

{

    Triangle(int x, int y)

    {

        a=x;

        b=y;

    }

}
```

```

void printArea()
{
    System.out.println("Area is "+(a*b*0.5));
}

}

class Circle extends Shape
{

    Circle(int x)
    {
        a=x;
    }

    void printArea()
    {
        System.out.println("Area is "+(a*a*3.14));
    }
}

class lab5
{
    public static void main(String ss[])
    {
        int l,b,ba,h,ra;
        Scanner sc = new Scanner(System.in);

        System.out.println("enter the length and breadth of rectangle");
        l= sc.nextInt();
        b= sc.nextInt();
        Rectangle r= new Rectangle(l,b);
        r.printArea();
    }
}

```

```

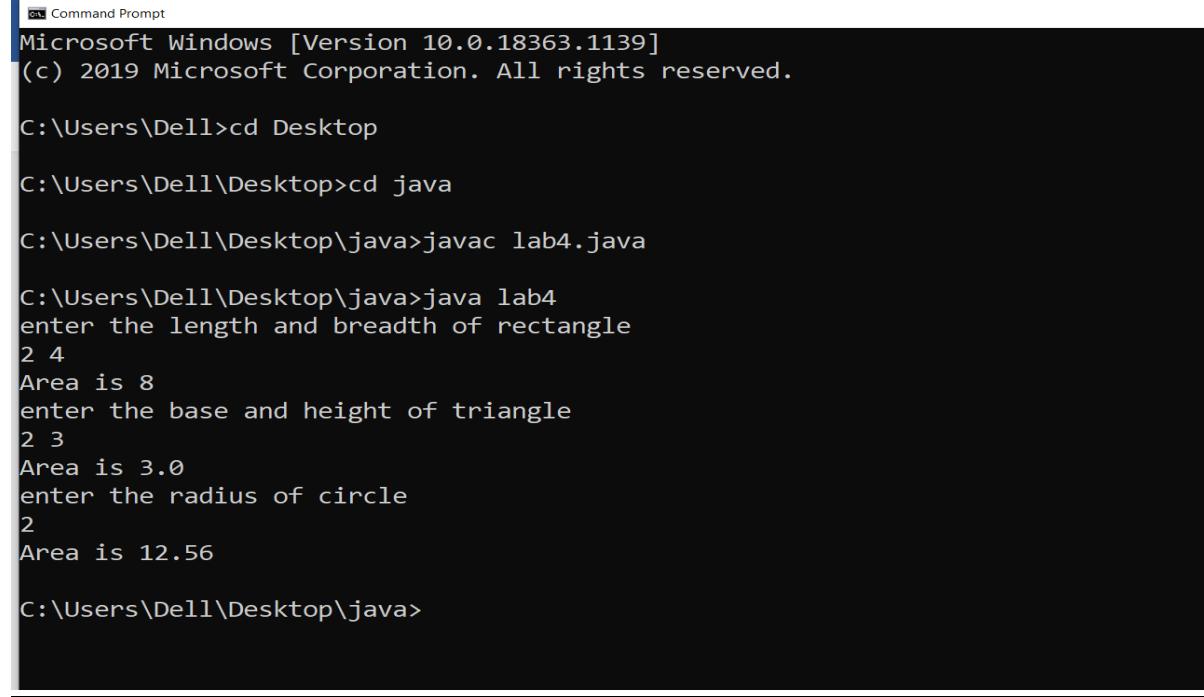
System.out.println("enter the base and height of triangle");
ba= sc.nextInt();
h= sc.nextInt();
Triangle t = new Triangle(ba,h);
t.printArea();

System.out.println("enter the radius of circle");
ra= sc.nextInt();
Circle c = new Circle(ra);
c.printArea();
}

}

```

OUTPUT



```

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

C:\Users\Dell>cd Desktop

C:\Users\Dell\Desktop>cd java

C:\Users\Dell\Desktop\java>javac lab4.java

C:\Users\Dell\Desktop\java>java lab4
enter the length and breadth of rectangle
2 4
Area is 8
enter the base and height of triangle
2 3
Area is 3.0
enter the radius of circle
2
Area is 12.56

C:\Users\Dell\Desktop\java>

```

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:

- a) Accept deposit from customer and update the balance.
- b) Display the balance.
- c) Compute and deposit interest
- d) Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

Lab Program : Bank

⑥

```

import java.util.*;
import java.lang.Math;
class Account
{
    String name;
    int acctno;
    char type;
    double def;
    boolean cheq;
    double balance;

    void get (char c)
    {
        type = c;
        if (c == 'S' || c == 's')
            cheq = false;
        else cheq = true;
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter your name");
        name = sc.nextLine ();
        System.out.println ("Enter the account number");
        acctno = sc.nextInt ();
        System.out.println ("Enter the current available
                           balance in your account");
        balance = sc.nextDouble ();
    }

    void putd ()
    {
        System.out.println ("Account details");
        System.out.println ("Name: " + name);
        System.out.println ("Account Number: " + acctno);
        System.out.println ("balance: " + balance);
    }
}

```

```
void dep ()  
{ Scanner sc = new Scanner (System.in);  
System.out.println ("Enter the amount to be deposited");  
dep = sc.nextDouble();  
balance = balance + dep;  
System.out.println ("Amount has been deposited and  
balance has been updated");  
}
```

```
void display ()  
{
```

```
System.out.println ("Balance amount is "+balance);  
}
```

```
void check ()
```

```
{ if ( cheq == false )
```

```
System.out.println ("Check book facility is not available");  
else
```

```
System.out.println ("Cheque book facility is available");  
}
```

```
}
```

```
class Saving extends Account {
```

```
double rat;
```

```
double s-with;
```

```
int n;
```

```
int ch;
```

```
double amt;
```

```
double term;
```

```
double pr;
```

```
void ci ()  
{ Scanner sc = new Scanner (System.in);  
    System.out.println("Enter principle deposit amount");  
    pr = sc.nextDouble();  
    System.out.println("Enter rate of interest");  
    rate = sc.nextDouble();  
    System.out.println("Enter the term (years)");  
    term = sc.nextDouble();  
    System.out.println("Enter the number of times interest is  
compounded annually");  
    n = sc.nextInt();  
    amt = pr * Math.pow((1 + (rate / 100)), (n * term));  
    balance += amt;  
    System.out.println("Interest is compounded and  
deposited; balance is updated");  
}
```

```
void withdraw ()  
{ Scanner sc = new Scanner (System.in);  
    System.out.println("Enter the amount of money to be withdrawn");  
    s_with = sc.nextDouble();  
    if (s_with > balance)  
        System.out.println("Insufficient balance");  
    else  
    { balance = balance - s_with;  
        System.out.println("Money has been withdrawn and  
balance has been updated");  
    }  
}
```

```

class Current extends Account
{
    double c-with;
    double pen;
    double min;

    Current()
    {
        pen = 100;
        min = 500;
    }

    void withdraw()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the amount to be withdrawn");
        c-with = sc.nextDouble();
        if (c-with > balance)
        {
            System.out.println("Insufficient balance");
            return;
        }
        else
        {
            balance = balance - c-with;
            System.out.print("Amount has been withdrawn and balance
                           has been updated ");
        }
        if (balance < min)
        {
            System.out.println("Balance is below the min threshold.
                               Service penalty charge = 100/-");
            if (balance < pen)
                System.out.println("Due to insufficient funds, penalty charge
                               will be deducted from account after replenishing.
                               current balance is " + balance);
            else
                balance = balance - pen;
        }
    }
}

```

System.out.println("Penalty charge has been deducted
from account balance. Current balance is "+balance);

}

```
class lab5
{
    public static void main (String args[])
    {
        int ch, chh;
        Scanner sc = new Scanner (System.in);
        System.out.println ("--- Welcome ---");
        System.out.println ("Savings account or current account ?
1- Savings ; 2- Current ");
        int ch = sc.nextInt();
        if (ch == 1)
    }
```

Saving s = new Saving();

s.get ('S');

do {

System.out.println ('1. Deposit money \n 2. Calculate
compound interest \n 3. withdraw money \n 4. Display
balance \n 5. Cheque book facility \n 6. Exit ');
System.out.println ("Enter your choice"),

chh = sc.nextInt();

switch (chh)

{

case 1: s.dep();
break;

case 2 : s.ci();
break;

case 3 : s.with-s();
break;

case 4 : s.display();
break;

case 5 : s.check();
break;

case 6 : break;

default : System.out.println("Wrong option");
break;

} } while (ch != 6);

}
else if (ch == 2)
{ Current cr = new Current();
cr.get('c');
do {

System.out.println("1. Deposit money\n2. Chequebook facility\n3.
Withdraw money\n4. Display balance\n5. Exit");

cch = sc.nextInt();

switch (cch)

{ case 1 : cr.dep();
break;

case 2 : cr.check();
break;

case 3 : cr.with-d();
break;

```
case 4 : cr. display();  
break;  
case 5 : break;  
default : System.out.println("Wrong option");  
break;  
}  
} while (cch != 5);  
}  
else  
System.out.println("Wrong!");  
}  
y.
```

```
import java.util.*;
import java.lang.Math;
class Account
{
    String name;
    int acctno;
    char type;
    double balance;
    double dep;
    boolean cheq;

    void get(char c)
    {
        type = c;
        if(c=='s' || c == 'S')
            cheq=false;
        else cheq=true;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter your name");
        name = sc.nextLine();
        System.out.println("Enter the account number");
        acctno = sc.nextInt();
        System.out.println("Enter the current available balance in your account");
        balance= sc.nextDouble();
    }

    void putd()
```

```
{  
    System.out.println("Account details");  
    System.out.println("Name: "+name);  
    System.out.println("Account number: "+acctno);  
    System.out.println("Account type :"+type);  
    System.out.println("balance: "+balance);  
}  
  
void dep()  
{  
    Scanner ss = new Scanner(System.in);  
    System.out.println("Enter the amount to be deposited");  
    dep= ss.nextDouble();  
    balance=balance +dep;  
    System.out.println("Amount has been deposited and balance has been updated");  
}  
  
void display()  
{  
    System.out.println("Balance amount is "+balance);  
}  
  
void check()  
{  
    if(cheq==false)  
        System.out.println("Cheque book facility is not available");  
    else  
        System.out.println("Cheque book facility is available");  
}
```

```
}
```

```
class Saving extends Account
```

```
{
```

```
    double rate;
```

```
    double s_with;
```

```
    int n;
```

```
    int ch;
```

```
    double amt;
```

```
    double term;
```

```
    double pr;
```

```
    void ci()
```

```
{
```

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

```
    System.out.println("Enter principal deposit amount");
```

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

```
    System.out.println("Enter the rate of interest");
```

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

```
    System.out.println("Enter the term(years)");
```

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

```
    System.out.println("Enter the number of times interest in compounded annually");
```

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

```
    amt = pr* Math.pow((1+(rate/100)),(n*term));
```

```
    balance+= amt;
```

```
System.out.println("Interest is compounded and deposited; balance is updated");

}

void with_s()
{

Scanner ss = new Scanner(System.in);

System.out.println("Enter the amount of money to be withdrawn");

s_with = ss.nextDouble();

if(s_with>balance)

System.out.println("Insufficient balance");

else

{

balance= balance - s_with;

System.out.println("Money has been withdrawn and balance has been updated");

}

}

}

class Current extends Account

{

double c_with;

double pen;

double min;

Current()

{
```

```

pen=100;
min=500;
}

void with_c()
{
Scanner xx = new Scanner(System.in);
System.out.println("Enter the amount to be withdrawn");
c_with= xx.nextDouble();
if(c_with>balance)
{
    System.out.println("Insufficient funds!");
    return;
}
else
{
    balance= balance- c_with;
    System.out.println("Amount has been withdrawn and balance has been updated");
}
if(balance<min)
{
    System.out.println("Balance is below the minimum threshold. Service penalty charge = 100/- .");
    if(balance<pen)
        System.out.println("Due to insufficient funds, penalty charge will be deducted from account
after replenishing. Current balance is "+balance);
    else
    {
        balance= balance-pen;
    }
}
}

```

```
        System.out.println("Penalty charge has been deducted from account balance. Current  
balance is "+balance);  
    }  
}  
}  
}
```

```
class bank  
{  
    public static void main(String sss[])  
    {  
        int cch, chh;  
        Scanner sx = new Scanner(System.in);  
        System.out.println("-----Welcome-----");  
        System.out.println("Savings account or current account? 1- Savings; 2- Current");  
        int ch= sx.nextInt();  
        if(ch==1)  
        {  
            Saving s = new Saving();  
            s.get('S');  
            do{  
                System.out.println("1. Deposit money\n2. Calculate compound interest\n3.  
Withdraw money\n4. Display balance\n5. Cheque book facility\n6. Exit");  
                System.out.println("Enter your choice");  
                chh= sx.nextInt();  
                switch(chh)  
                {  
                    case 1:  
                        s.dep();
```

```
        break;

    case 2:
        s.ci();
        break;

    case 3:
        s.with_s();
        break;

    case 4:
        s.display();
        break;

    case 5:
        s.check();
        break;

    case 6:
        break;

    default:
        System.out.println("Wrong option.");
        break;
    }

}while(chh!=6);

}

else if(ch==2)
```

```
{  
    Current cr = new Current();  
    cr.get('C');  
    do{  
        System.out.println("1. Deposit money\n2. Chequebook facility\n3. Withdraw  
money\n4. Display balance\n5. Exit");  
        cch= sx.nextInt();  
        switch(cch){  
            {  
                case 1:  
                    cr.dep();  
                    break;  
  
                case 2:  
                    cr.check();  
                    break;  
  
                case 3:  
                    cr.with_c();  
                    break;  
  
                case 4:  
                    cr.display();  
                    break;  
  
                case 5:  
                    break;  
  
            default:  
        }  
    }  
}
```

```

        System.out.println("Wrong option.");
        break;
    }
}while(cch!=5);

}

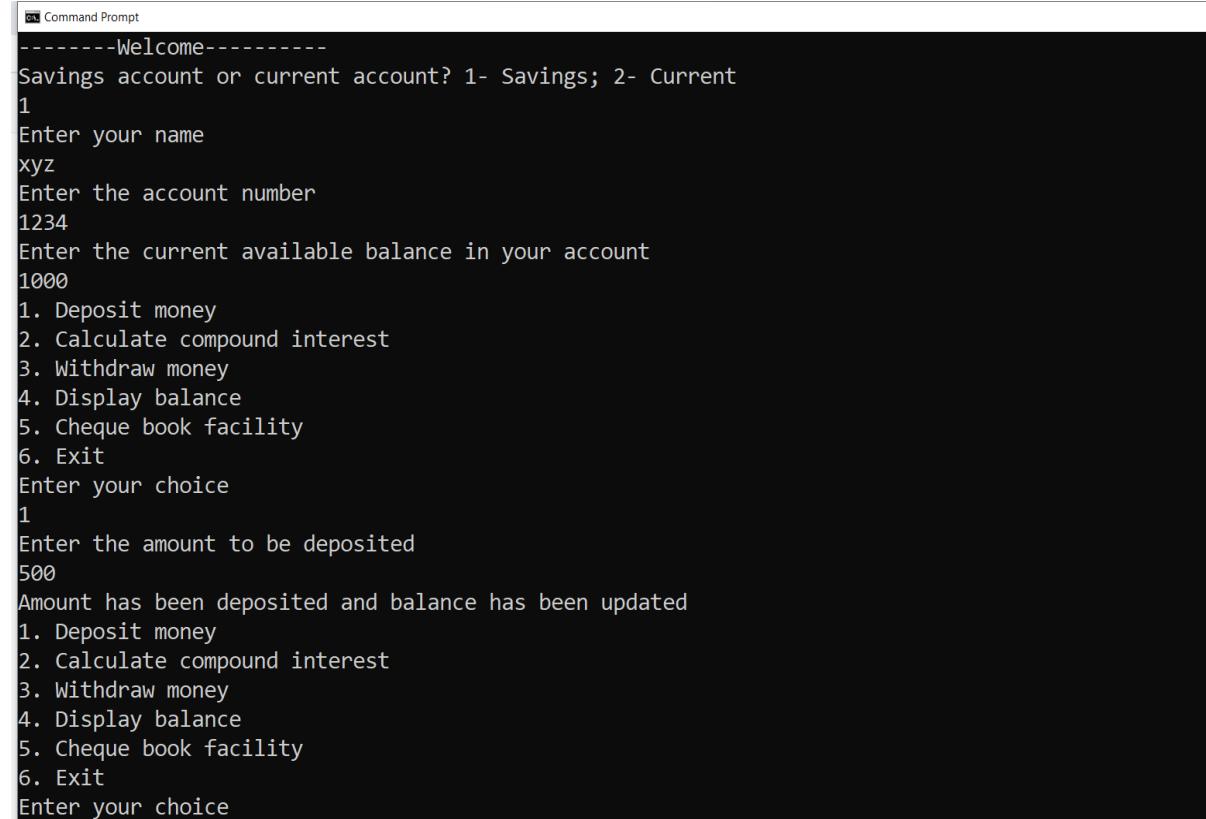
else System.out.println("Wrong!");

}

}

```

OUTPUT



The screenshot shows a Command Prompt window with the following interaction:

```

Command Prompt
-----Welcome-----
Savings account or current account? 1- Savings; 2- Current
1
Enter your name
xyz
Enter the account number
1234
Enter the current available balance in your account
1000
1. Deposit money
2. Calculate compound interest
3. Withdraw money
4. Display balance
5. Cheque book facility
6. Exit
Enter your choice
1
Enter the amount to be deposited
500
Amount has been deposited and balance has been updated
1. Deposit money
2. Calculate compound interest
3. Withdraw money
4. Display balance
5. Cheque book facility
6. Exit
Enter your choice

```

```
Enter your choice
2
Enter principal deposit amount
1500
Enter the rate of interest
10
Enter the term(years)
2
Enter the number of times interest is compounded annually
1
Interest is compounded and deposited; balance is updated
1. Deposit money
2. Calculate compound interest
3. Withdraw money
4. Display balance
5. Cheque book facility
6. Exit
Enter your choice
3
Enter the amount of money to be withdrawn
```

```
Enter the amount of money to be withdrawn
500
Money has been withdrawn and balance has been updated
1. Deposit money
2. Calculate compound interest
3. Withdraw money
4. Display balance
5. Cheque book facility
6. Exit
Enter your choice
5
Cheque book facility is not available
1. Deposit money
2. Calculate compound interest
3. Withdraw money
4. Display balance
5. Cheque book facility
6. Exit
Enter your choice
4
Balance amount is 2815.0
1. Deposit money
2. Calculate compound interest
3. Withdraw money
4. Display balance
```

```
Command Prompt
-----Welcome-----
Savings account or current account? 1- Savings; 2- Current
2
Enter your name
abc
Enter the account number
3214
Enter the current available balance in your account
7500
1. Deposit money
2. Chequebook facility
3. Withdraw money
4. Display balance
5. Exit
1
Enter the amount to be deposited
500
Amount has been deposited and balance has been updated
1. Deposit money
2. Chequebook facility
3. Withdraw money
4. Display balance
5. Exit
2
Cheque book facility is available
1. Deposit money
2. Chequebook facility
```

```
1. Deposit money
2. Chequebook facility
3. Withdraw money
4. Display balance
5. Exit
3
Enter the amount to be withdrawn
1000
Amount has been withdrawn and balance has been updated
1. Deposit money
2. Chequebook facility
3. Withdraw money
4. Display balance
5. Exit
4
Balance amount is 7000.0
1. Deposit money
2. Chequebook facility
3. Withdraw money
4. Display balance
5. Exit
5
```

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.

Trusha Lakharia
1BMINCS214

Lab Program 6 : Packages

```
package CIE;
import java.util.*;
public class personal
{
    public String name;
    public int sem;
    public String usn;

    public void read()
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter the name");
        name = sc.nextLine();
        System.out.println ("Enter the semester");
        sem = sc.nextInt();
        System.out.println ("Enter the USN");
        usn = sc.nextLine();
    }

    public void display()
    {
        System.out.println ("Student details:");
        System.out.println ("Name: " + name + "\n USN: " + usn +
                           "\n Semester: " + sem);
    }
}

package CIE;
import java.util.*;
public class internals extends personal
{
    public double cie[];
}
```

```
public void accept () {  
    CIE = new double [5];  
    Scanner sc = new Scanner (System.in);  
    for (int i=0; i<5; i++)  
    {  
        System.out.println ("CIE mark for course " + (i+1) + ":");  
        CIE [i] = sc.nextDouble();  
    }  
}  
  
package SEE;  
import java.util.*;  
import CIE.*;  
public class external extends personal  
{  
    public double see [];  
    public void get()  
    {  
        see = new double [5];  
        Scanner sc = new Scanner (System.in);  
        for (int i=0; i<5; i++)  
        {  
            System.out.println ("SEE marks for course " + (i+1) + ":";  
        }  
    }  
}
```

```

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

class Lab6
{
    public static void main (String args[])
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter the number of students");
        int n = sc.nextInt();
        CIE.internals[] = new CIE.internals[n];
        SEE.internals[] = new SEE.internals[n];
        int i, j;
        for (i = 0; i < n; i++)
        {
            System.out.println ("Student " + (i+1));
            in[i] = new CIE.internals();
            en[i] = new SEE.internals();
            in[i].read();
            System.out.println ("CIE Marks:");
            in[i].accept();
            System.out.println ("SEE Marks:");
            en[i].get();
            System.out.println ();
            in[i].display();
            for (j = 0; j < 5; j++)
                System.out.println ("Total marks for course " + (j+1) +
                    ":" + (in[i].cie[j] + (en[i].see[j]/2)));
        }
    }
}

```

```
package CIE;

import java.util.*;

public class personal

{

    public String name;

    public int sem;

    public String usn;

    public void read()

    {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the name");

        name = sc.next();

        System.out.println("Enter the semester");

        sem = sc.nextInt();

        System.out.println("Enter the USN");

        usn = sc.next();

    }

    public void display()

    {

        System.out.println("Student details: ");

        System.out.println("Name: "+name+"\nUSN: "+usn+"\nSem: "+sem);

    }

}
```

```
package CIE;

import java.util.*;

public class internals extends personal
```

```
{  
    public double cie[];  
  
    public void accept()  
    {  
        cie= new double[5];  
        Scanner sc = new Scanner(System.in);  
        for(int i=0;i<5;i++)  
        {  
            System.out.println("CIE mark for course "+(i+1)+" : ");  
            cie[i]= sc.nextDouble();  
        }  
    }  
}
```

```
package SEE;  
import java.util.*;  
import CIE.*;  
public class externals extends personal  
{  
    public double see[];  
  
    public void get()  
    {  
        see= new double[5];  
        Scanner sc = new Scanner(System.in);  
        for(int i=0;i<5;i++)  
        {
```

```

        System.out.println("SEE mark for course "+(i+1)+" : ");
        see[i]= sc.nextDouble();
    }

}

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

class LAB6
{
    public static void main(String args[])
    {
        Scanner sx = new Scanner(System.in);
        System.out.println("Enter the number of students");
        int n= sx.nextInt();
        CIE.internals in[]= new CIE.internals[n];
        SEE.externals en[]= new SEE.externals[n];
        int i,j;
        for(i=0;i<n;i++)
        {
            System.out.println("Student "+(i+1));
            in[i] = new CIE.internals();
            en[i] = new SEE.externals();
            in[i].read();
        }

        System.out.println("CIE MARKS:");
        in[i].accept();
    }
}
```

```

        System.out.println("SEE MARKS:");
        en[i].get();
        System.out.println();
        in[i].display();
        for(j=0;j<5;j++)
            System.out.println("Total Marks for course "+(j+1)+": "+(in[i].cie[j] +
(en[i].see[j]/2)));
    }
}
}

```

OUTPUT

```

C:\Users\DELL\Desktop>java LAB6
Enter the number of students
2
Student 1
Enter the name
TRISHA
Enter the semester
3
Enter the USN
214
CIE MARKS:
CIE mark for course 1 :
40
CIE mark for course 2 :
45
CIE mark for course 3 :
46
CIE mark for course 4 :
50
CIE mark for course 5 :
47
SEE MARKS:
SEE mark for course 1 :
80
SEE mark for course 2 :
90

```

```
 Command Prompt
SEE mark for course 2 :
90
SEE mark for course 3 :
85
SEE mark for course 4 :
75
SEE mark for course 5 :
90
```

```
Student details:
Name: TRISHA
USN: 214
Sem: 3
Total Marks for course 1: 80.0
Total Marks for course 2: 90.0
Total Marks for course 3: 88.5
Total Marks for course 4: 87.5
Total Marks for course 5: 92.0
Student 2
Enter the name
SAM
Enter the semester
3
Enter the USN
200
CIE MARKS:
CIE mark for course 1 :
```

```
 Command Prompt
CIE MARKS:
CIE mark for course 1 :
45
CIE mark for course 2 :
46
CIE mark for course 3 :
47
CIE mark for course 4 :
48
CIE mark for course 5 :
49
SEE MARKS:
SEE mark for course 1 :
90
SEE mark for course 2 :
91
SEE mark for course 3 :
92
SEE mark for course 4 :
93
SEE mark for course 5 :
94
```

Student details:

Name: SAM

USN: 200

Sem: 3

Total Marks for course 1: 90.0

Total Marks for course 2: 91.5

Total Marks for course 3: 93.0

Total Marks for course 4: 94.5

Total Marks for course 5: 96.0

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

Trisha

Lab Program 7 : Generics

```
class TwoGen < T, V > {
    T ob1;
    V ob2;

    TwoGen ( T o1, V o2 ) {
        ob1 = o1;
        ob2 = o2;
    }
```

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

```
T getOb1() {
    return ob1;
}
```

```
V getOb2() {
    return ob2;
}
```

```
class LAB7 {
```

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

```
TwoGen < Integer, String > tgObj = new TwoGen < Integer, String >
    ( 88, "Generics" );
```

```
tgObj.showTypes();  
  
int v = tgObj.getOb1();  
System.out.println("value: " + v);  
String str = tgObj.getOb2();  
System.out.println("value: " + str);  
}  
}
```

```
class TwoGen<T, V> {  
  
    T ob1;  
    V ob2;  
  
    TwoGen(T o1, V o2) {  
        ob1 = o1;  
        ob2 = o2;  
    }  
  
    void showTypes() {  
        System.out.println("Type of T is " + ob1.getClass().getName());  
        System.out.println("Type of V is " + ob2.getClass().getName());  
    }  
    T getob1() {
```

```
return ob1;
}

V getob2() {
    return ob2;
}

}

class LAB7 {
    public static void main(String args[]) {

        TwoGen<Integer, String> tgObj =new TwoGen<Integer, String>(101, "program 7");

        tgObj.showTypes();

        int v = tgObj.getob1();
        System.out.println("value: " + v);

        String str = tgObj.getob2();
        System.out.println("value: " + str);
    }
}
```

OUTPUT

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

C:\Users\Dell>CD Desktop

C:\Users\Dell\Desktop>cd java

C:\Users\Dell\Desktop\java>javac LAB7.java

C:\Users\Dell\Desktop\java>java LAB7
Type of T is java.lang.Integer
Type of V is java.lang.String
value: 101
value: program 7

C:\Users\Dell\Desktop\java>
```

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 WrongAge() when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

Lab Program 8: Exceptions

```
import java.util.*;  
class Wrongage extends Exception {  
    int f, s;  
    Wrongage (int fage, int sage) {  
        f = fage;  
        s = sage;  
    }  
    public String toString () {  
        return "Please enter the correct ages as father's age can't be  
        less than or equal to the son's age";  
    }  
}
```

```
class NegativeAge extends Exception {  
    int n;  
    NegativeAge (int fage) {  
        n = fage;  
    }  
}
```

```
public String toString () {  
    return "Age can't be a negative value";  
}  
}
```

```
class Father {  
    int fage;  
    Scanner in = new Scanner (System.in);  
    Father () throws NegativeAge {  
    }
```

```
    System.out.println ("Enter the father's age");  
    fage = in.nextInt();  
    if (fage < 0) {  
        throw new NegativeAge (fage);  
    }
```

Tanish
IBM19CS216

}

}}

class Son extends Father
{ int sage,

Scanner in = new Scanner (System.in);

Son () throws NegativeAge, WrongAge {
super ();

System.out.println ("Enter the son's age");

sage = in.nextInt();

if (sage < 0)
{

throw new NegativeAge (sage);

}

if (sage >= sage) {

throw new WrongAge (sage, sage);

}

}

class LAB8 {

public static void main (String args []) {

try {

Son s = new Son ();

}

catch (NegativeAge n) {

System.out.println ("Exception: " + n);

}

catch (WrongAge) {

System.out.println ("Exception: " + w);

}

}

```
import java.util.*;

class WrongAge extends Exception{

int f,s;

WrongAge(int fage,int sage){

f=fage;

s=sage;

}

public String toString(){

return "Please enter the correct ages as father's age can't be less than or equal to the son's age./";

}

}

class NegativeAge extends Exception{

int x;

NegativeAge(int fage){

x=fage;

}

public String toString(){

return "Age can't be a negative value./";

}

}
```

```
class Father

{

int fage;

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

```
Father() throws NegativeAge
{
    System.out.println("Enter the father's age:");
    fage=in.nextInt();
    if(fage<0){
        throw new NegativeAge(fage);
    }
}
```

```
class Son extends Father
{
    int sage;
    Scanner in=new Scanner(System.in);
    Son() throws NegativeAge,WrongAge{
        super();
        System.out.println("Enter the son's age :");
        sage=in.nextInt();
        if(sage<0)
        {
            throw new NegativeAge(sage);
        }
        if(sage>=fage){
            throw new WrongAge(fage,sage);
        }
    }
}

class LAB8{
    public static void main(String args[]){

```

```
try{
Son s=new Son();
}
catch(NegativeAge n){
System.out.println("Exception:"+n);
}
catch(WrongAge w){
System.out.println("Exception:"+w);
}
}}
```

OUTPUT

```
C:\Users\DELL\Desktop>javac Lab8.java

C:\Users\DELL\Desktop>java LAB8
Enter the father's age:
-34
Exception:Age can't be a negative value.

C:\Users\DELL\Desktop>javac Lab8.java

C:\Users\DELL\Desktop>java LAB8
Enter the father's age:
40
Enter the son's age :
40
Exception:Please enter the correct ages as father's age can't be less than or equal to the son's age.

C:\Users\DELL\Desktop>javac Lab8.java

C:\Users\DELL\Desktop>java LAB8
Enter the father's age:
40
Enter the son's age :
15

C:\Users\DELL\Desktop>javac Lab8.java

C:\Users\DELL\Desktop>java LAB8
```

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.

Taisha

Lab Program 9 : Multithread

```
class NewThread implements Runnable
{
    private String name;
    private int interval;
    private Thread t;
    NewThread ( String threadname , int interval )
    {
        this.name = threadname;
        this.interval = interval;
        t = new Thread ( this, name );
        t.start ();
    }
    public void run ()
    {
        try {
            for ( int i = 5 ; i > 0 ; i -- )
            {
                System.out.println ( i + " : thread-->" + this.name );
                Thread.sleep ( this.interval );
            }
        }
        catch ( InterruptedException e ) {
            System.out.println ( name + " interrupted " );
        }
    }
    public static void main ( String args [] )
    {
        new NewThread ( "BMS College of Engineering" , 10000 );
        new NewThread ( "CSE" , 2000 );
    }
}
```

```
class NewThread implements Runnable
{
    private String name;
    private int interval;
    private Thread t;
    NewThread (String threadname,int interval)
    {
        this.name=threadname;
        this.interval=interval;
        t=new Thread(this.name);
        t.start();
    }
    public void run()
    {
        try{
            for(int i=5;i>0;i--)
            {
                System.out.println(i +":thread-->" +this.name);
                Thread.sleep(this.interval);
            }
        }
        catch(InterruptedException e){
            System.out.println(name+"interrupted");
        }
    }
}
class LAB9
{
    public static void main(String args[])
}
```

```
{  
    new NewThread("BMS College of Engineering",10000);  
    new NewThread("CSE",2000);  
}  
}
```

OUTPUT

```
C:\Users\Dell\Desktop\java>javac LAB9.java  
  
C:\Users\Dell\Desktop\java>java LAB9  
5:thread-->BMS College of Engineering  
5:thread-->CSE  
4:thread-->CSE  
3:thread-->CSE  
2:thread-->CSE  
1:thread-->CSE  
4:thread-->BMS College of Engineering  
3:thread-->BMS College of Engineering  
2:thread-->BMS College of Engineering  
1:thread-->BMS College of Engineering  
  
C:\Users\Dell\Desktop\java>
```