VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**“JnanaSangama”, Belgaum -590014, Karnataka.**



# LAB REPORT

**on**

**OBJECT ORIENTED JAVA PROGRAMMING (22CS3PCOOJ)**

***Submitted by***

# Vrishank J Vasist

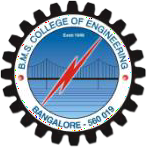
# (1BM21CS246)

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

***in***

# COMPUTER SCIENCE AND ENGINEERING



**B.M.S. COLLEGE OF ENGINEERING**

**(Autonomous Institution under VTU)**

# BENGALURU-560019

**October-2022 to Feb-2023**

**B. M. S. College of Engineering,**

**Bull Temple Road, Bangalore 560019**

(Affiliated To Visvesvaraya Technological University, Belgaum)

**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled OBJECT ORIENTED JAVA PROGRAMMING “ (22CS3PCOOJ)” carried out by **VRISHANK J VASIST (1BM21CS246)** who is

bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of OBJECT ORIENTED JAVA PROGRAMMING (22CS3PCOOJ) work prescribed for the said degree.

Name of the Lab-Incharge **Dr. Jyothi S Nayak**

**Basavaraj Jakkali**

Associate professo Professor and Head

Department of CSE Department of CSE

BMSCE, Bengaluru BMSCE, Bengaluru

# INDEX

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl No.** | **Date** | **Experiment Title** | **Page No.** |
| 1. | 17/11/2022 | Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 - 4ac is negative, display a message stating  that there are no real solutions. | 6 |
| 2. | 24/11/2022 | 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. | 8 |
| 3. | 01/12/2022 | 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. | 11 |
| 4. | 08/12/2022 | 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. | 14 |
| 5. | 29/12/2022 | 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 Cur- | 17 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:   1. Accept deposit from customer and update the balance. 2. Display the balance. 3. Compute and deposit interest 4. Permit withdrawal and update the balance Check for the minimum balance, impose   penalty if necessary and update the balance. |  |
| 6. | 12/01/2023 | 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. | 23 |
| **7.** | 05/01/2023 | 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. | 26 |
| 8. | 12/01/2023 | 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 | 30 |
| 9. | 19/01/2023 | 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. | 32 |
| 10. | 26/01/2023 | Demonstrate Inter process Communication and deadlock |  |

**EXPERIMENT-1**

**1.Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a message stating that there are no real solutions.**

import java.util.Scanner;

class Quad

{

double d, r1, r2;

Quad(double a, double b, double c)

{

if(a==0)

{

System.out.println("Coefficient 'a' can not be zero for this.");

}

else

{

d = b\*b-4\*a\*c;

if(d>0)

{

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

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

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

System.out.println("First root is: " + r1 +"\nSecond Root is: "

+r2);

}

else if(d==0)

{

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

r1 = -b/(2\*a);

System.out.println("The roots are both: " + r1);

}

else

{

System.out.println("The roots are imaginary and distinct.");

r1 = -b/(2\*a);

r2 = (Math.sqrt(Math.abs(d)))/(2\*a);

System.out.println("First root is:" + r1 + "+i" + r2);

System.out.println("Second root is:" + r1 + "-i" + r2);

}

}

}

}

class Quadratic\_Equations

{

public static void main(String args[])

{

System.out.println("Enter your coefficients:\n");

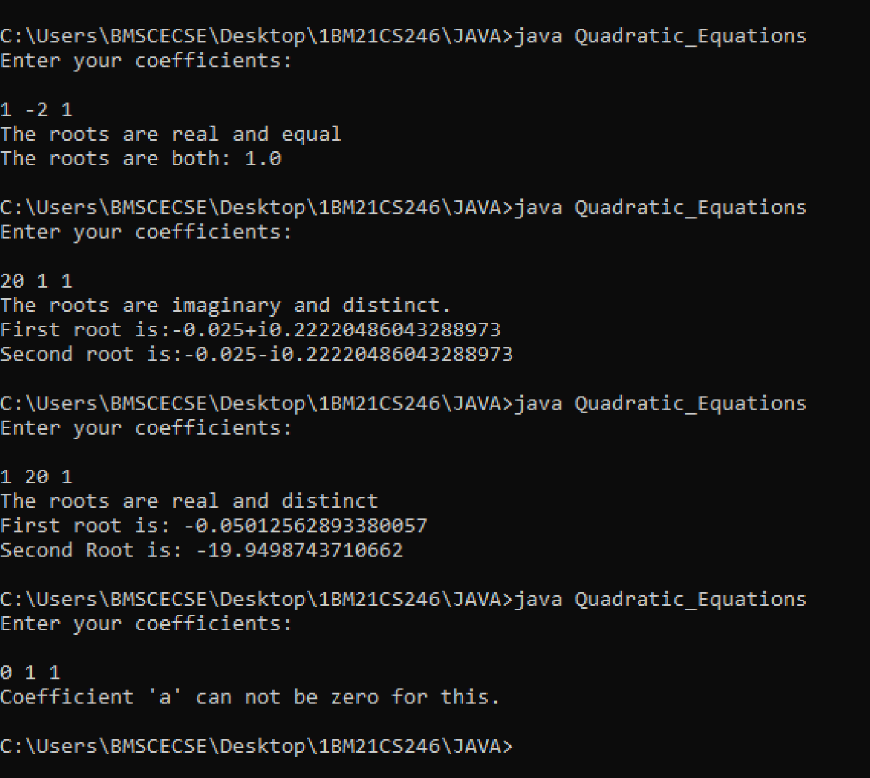
Scanner ss = new Scanner(System.in);

Quad One = new Quad(ss.nextInt(), ss.nextInt(), ss.nextInt());

}

}

­Output:



# EXPERIMENT-2

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

import java.util.Scanner;

class SGPA{

double cgpa;

void sgpa(String u, int c[], int m[])

{

System.out.println("USN:\t"+u);

cgpa = (double)(c[0]\*m[0] + c[0]\*m[1] + c[1]\*m[2] + c[1]\*m[3] + c[1]\*m[4] + c[1]\*m[5] + c[2]\*m[6])/18;

System.out.println("SGPA:\t"+ cgpa);

}

}

class Student{

public static void main(String args[])

{

int credits[] = new int[3];

credits[0] = 1;

credits[1] = 3;

credits[2] = 4;

int marks[] = new int[7];

Scanner ss = new Scanner(System.in);

SGPA one = new SGPA();

System.out.println("Enter the name please:\t");

String name = ss.next();

System.out.println("Please enter "+name+"'s USN:\t");

String usn = ss.next();

System.out.println("Enter the marks of the subjects IDT, English, Maths, BEE, ECM, EVI, Physics in that order\n");

for(int i=0;i<7;i++)

{

marks[i] = ss.nextInt();

}

for(int i=0;i<7;i++)

{

if(marks[i]==100)

{

marks[i]=10;

}

else if(marks[i]<40)

{

marks[i] = 0;

}

else

{

for(int j=10; j>3;j--)

{

if(j\*10-10 <= marks[i] && marks[i]< j\*10)

{

marks[i] = j;

}

}

}

}

System.out.println("Would you like to view "+name+"'s Informations? 1/0");

if(ss.nextInt() != 0)

{

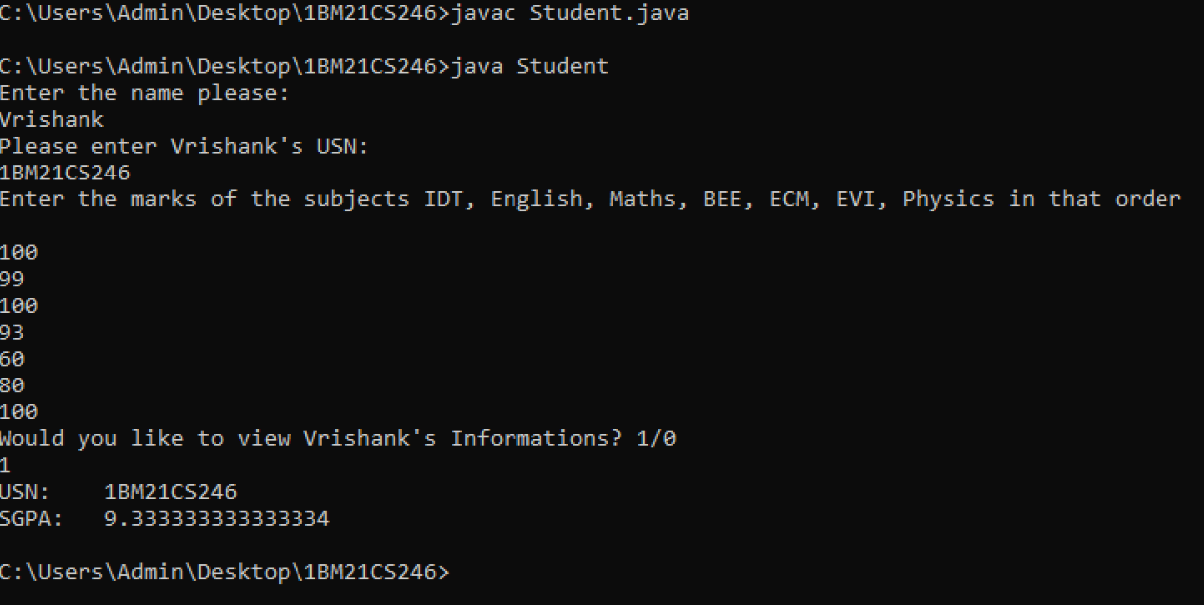
one.sgpa(usn, credits, marks);

}

}

}

Output:



# EXPERIMENT-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.**

import java.util.Scanner;

class Info{

String title; String author; int price; int num\_pages;

Info()

{

title = new String();

author = new String();

Scanner ss = new Scanner(System.in);

System.out.println("Please enter the name of the book:\t");

title = ss.next();

System.out.println("Please enter the name of the author:\t");

author = ss.next();

System.out.println("What are its prices and number of pages respectively:\t");

price = ss.nextInt(); num\_pages = ss.nextInt();

}

@Override

public String toString()

{

return "Book Title:\t"+title +"\nWritten by:\t"+author +"\nCost:\t"+price +"\nPages:\t"+num\_pages;

}

}

class Author

{

public static void main(String args[])

{

int total;

System.out.println("How many books do you want to record: ");

try (Scanner two = new Scanner(System.in))

{

total = two.nextInt();

Info one[] = new Info[total];

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

{

one[i] = new Info();

}

System.out.println("Do you want to check the information recorded? 0 to exit.");

if(two.nextInt() != 0)

{

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

{

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

}

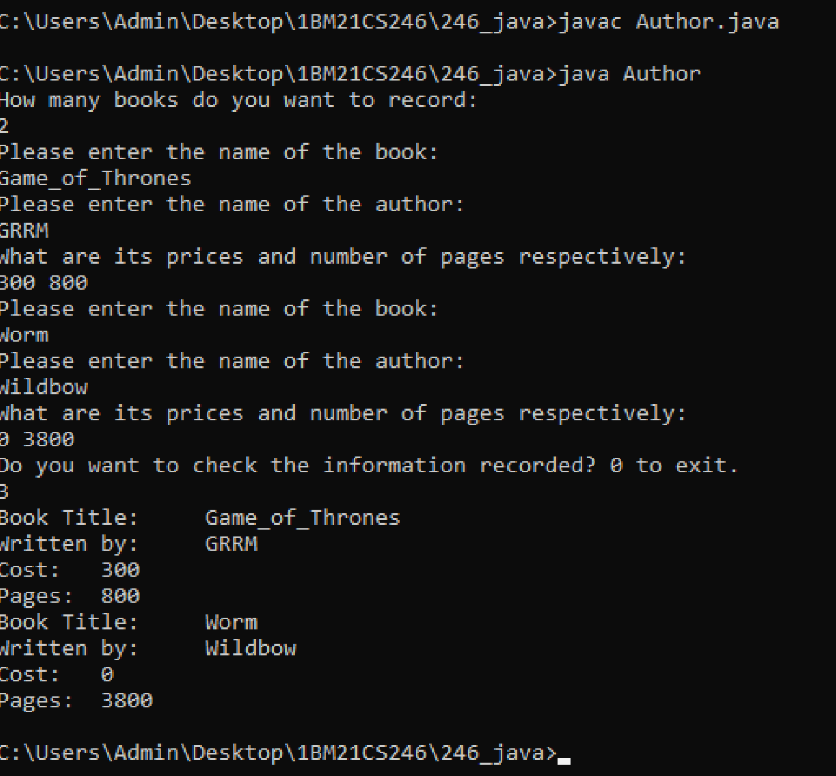
}

}

}

}

Output:



# EXPERIMENT 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.**

abstract class Shape

{

int one; int two;

void initial(int o, int t)

{

one = o; two = t;

}

abstract void printArea();

}

class Rectangle extends Shape

{

void printArea()

{

System.out.println("The area of the rectangle is:\t"+ one\*two);

}

}

class Triangle extends Shape

{

void printArea()

{

System.out.println("The area of the triangle is:\t"+ (double)0.5\*one\*two);

}

}

class Circle extends Shape

{

void printArea()

{

System.out.println("The area of the circle is:\t"+ (double) 3.14159\*one\*two);

}

}

class Areas

{

public static void main(String args[])

{

Triangle t1 = new Triangle();

Rectangle r1 = new Rectangle();

Circle c1 = new Circle();

t1.initial(50, 40); t1.printArea();

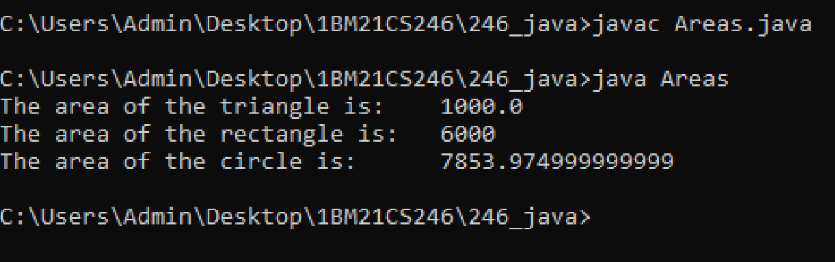
r1.initial(100, 60); r1.printArea();

c1.initial(50, 50); c1.printArea();

}

}

Output:



# EXPERIMENT 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 Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:**

1. **Accept deposit from customer and update the balance.**
2. **Display the balance.**
3. **Compute and deposit interest**
4. **Permit withdrawal and update the balance**

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

import java.util.Scanner;

class Account

{

String cust\_name, type\_of\_account;

int acc\_no;

double balance;

public void deposit(int choice)

{

if(choice == 0)

{

type\_of\_account = "Savings";

}

else

{

type\_of\_account = "Currency";

}

Scanner one = new Scanner(System.in);

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

cust\_name = one.next();

System.out.println("Enter acc\_no: ");

acc\_no = one.nextInt();

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

balance = one.nextDouble();

}

public void updateBal()

{

System.out.println("This is not available in the type of account you chose.");

}

public void withdraw()

{

System.out.println("This is not available in the type of account you chose.");

}

void checkBal()

{

System.out.println("This is not available in the type of account you chose.");

}

void computeInterest()

{

System.out.println("This is not available in the type of account you chose.");

}

void depositInterest()

{

System.out.println("This is not available in the type of account you chose.");

}

@Override

public String toString()

{

return "Name:\t" + this.cust\_name + "\nType of Account:\t"+ this.type\_of\_account+"\nAccount Number:\t"+ this.acc\_no+"\nAmount of Money in Account:\t"+ this.balance;

}

}

class Current extends Account

{

double interest, cheque\_book;

Current()

{

System.out.println("You've chosen our special Account this gives you access to our Cheque Book which holds info about your last ten transactions.");

}

public void updateBal()

{

System.out.println("Enter amount of money you want to deposit");

Scanner three = new Scanner(System.in);

cheque\_book = three.nextDouble();

balance += cheque\_book;

System.out.println("New Balance:\t"+balance);

}

public void withdraw()

{

System.out.println("Enter amount of money you want to withdraw");

Scanner two = new Scanner(System.in);

cheque\_book = two.nextDouble();

balance -= cheque\_book;

System.out.println("New Balance:\t"+balance);

}

public void computeInterest()

{

if(balance<50000)

{

interest = 1000;

System.out.println("You have gone below our 50k limit and will be charged 1k for every month you don't pay to keep your account over the minimum limit.");

balance = balance - interest;

System.out.println("Your new balance is:\t"+balance);

return;

}

interest = 100;

balance = balance - interest;

System.out.println("Your new balance is:\t"+balance);

}

public void depositInterest(double s)

{

interest = s;

balance = balance + interest - 100;

System.out.println("New Balance:\t"+balance);

}

}

class Saving extends Account

{

Saving()

{

System.out.println("Since you have chose a savings account, you don't have any cheque book privilages.\nIf you wish to withdraw, please come irl.\nThere is no interest and your money increases by 50 every month\n");

}

double servicecharge;

public void checkBal()

{

servicecharge = 50;

balance +=servicecharge;

}

}

class Banking

{

public static void main(String args[])

{

int choice;

Account Account\_obj;

System.out.println("Please enter the type of account you want. 0 to choose Savings.");

Scanner type = new Scanner(System.in);

choice = type.nextInt();

if(choice==0)

{

Saving ref = new Saving();

Account\_obj = ref;

}

else

{

Current ref = new Current();

Account\_obj = ref;

}

Account\_obj.deposit(choice);

System.out.println("Do you want to \n1)Display your account info\n2)Do you want to update your balance?\n3)Do you want to withdraw?\n");

while(true)

{

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

Scanner main = new Scanner(System.in);

choice = main.nextInt();

switch(choice)

{

case 1 : System.out.println(Account\_obj);

break;

case 2 :Account\_obj.updateBal();

Account\_obj.computeInterest();

break;

case 3 :

Account\_obj.withdraw();

Account\_obj.checkBal();

Account\_obj.depositInterest();

break;

case 4: System.exit(0);

break;

default : System.out.println("Invalid");

break;

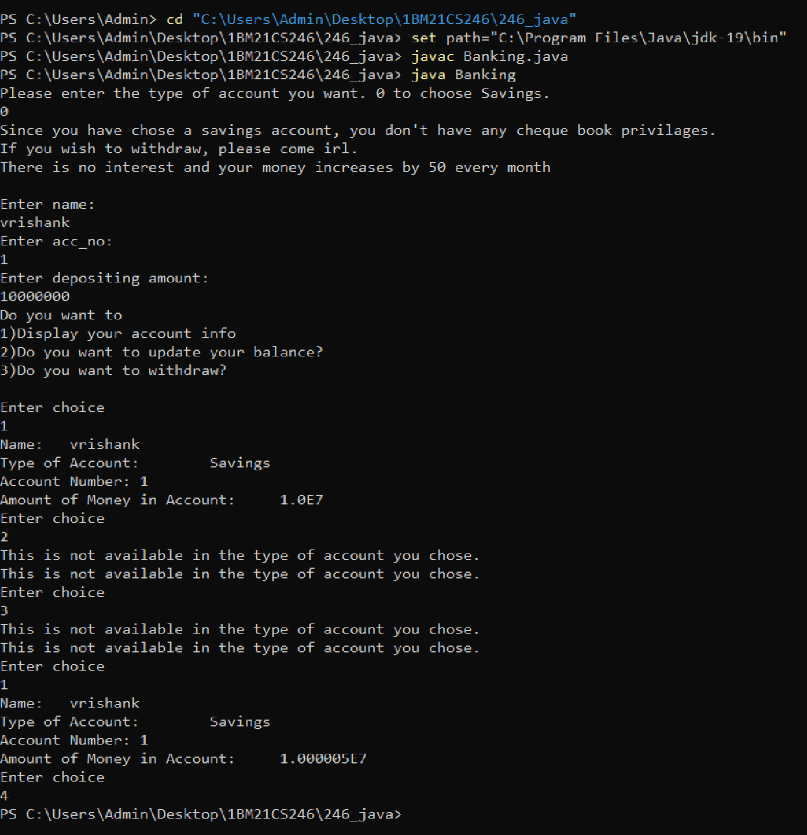
}

}

}

}

Output:



# EXPERIMENT 6

**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.**

import java.util.Scanner;

class Wrongage extends Exception

{ int detail;

Wrongage(int d)

{

detail=d;

}

public String toString()

{

return "Entered Wrong age is ["+detail+"]";

}

}

class Father {

int f;

Scanner in=new Scanner(System.in);

Father()

{

System.out.println("Enter father age ");

f=in.nextInt();

}

void checkage() throws Wrongage

{

if(f<0)

{

throw new Wrongage(f);

}

System.out.println("Father age positive");

}

}

class Son extends Father{

int s;

Scanner in=new Scanner(System.in);

Son()

{

super();

System.out.println("Enter son age ");

s=in.nextInt();

}

void checkages() throws Wrongage

{

super.checkage();

if(s<0)

{

throw new Wrongage(f);

}

System.out.println("Son age positive");

}

void checkage() throws Wrongage

{

if(s>f)

{

throw new Wrongage(s);

}

System.out.println("Father-Son age correct");

}

}

class Exceptionsssss{

public static void main(String args[])

{

int f,s;

Father fath=new Father();

Father r;

r=fath;

try{

r.checkage();

}

catch(Wrongage e){

System.out.println("Father age wrong"+e);

}

Son sn=new Son();

r=sn;

try{

sn.checkages();

r.checkage();

}

catch(Wrongage e){

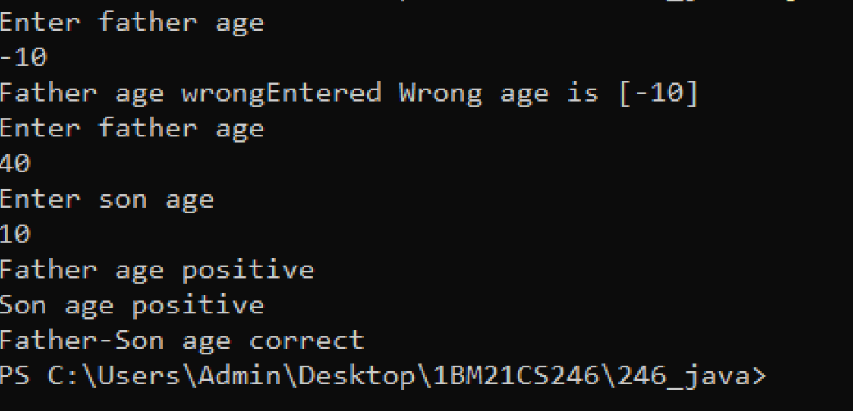
System.out.println("Son age wrong"+e);

}

}

}

Output:



# EXPERIMENT 7

**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.**

class Thread1 extends Thread{

public void run(){

while(true){

System.out.println("BMS College of eingineering");

try{

Thread.sleep(10000);

}

catch(InterruptedException e){

continue;

}

}

}

}

class Thread2 extends Thread{

public void run(){

while(true){

System.out.println("CSE");

try{

Thread.sleep(2000);

}

catch(InterruptedException e){

continue;

}

}

}

}

public class thread\_prog8 {

public static void main(String args[]){

System.out.println("Control-C to exit");

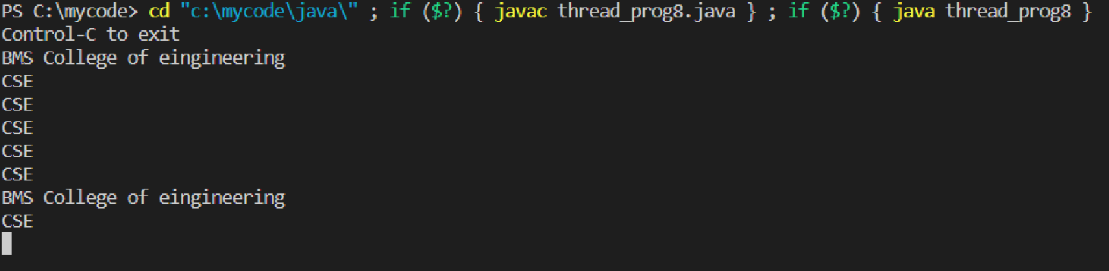
new Thread1().start();

new Thread2().start();

}

}

Output:



**EXPERIMENT 9**

**Demonstrate Inter process Communication and deadlock.**

class Q{

int n;

boolean valueSet = false;

synchronized int get(){

while(!valueSet){

try{

wait();

}

catch(InterruptedException e){

System.out.println("InterruptException caught");

}

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

valueSet = false;

notify();

return n;

}

return n;

}

synchronized void put(int n){

while(valueSet){

try{

wait();

}

catch(InterruptedException e){

System.out.println("InterruptException caught");

}

this.n = n;

valueSet = true;

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

notify();

}

}

}

class Producer implements Runnable{

Q q;

Producer(Q q){

this.q = q;

new Thread(this, "Producer").start();

}

public void run(){

int i=0;

while(true){

q.put(i++);

System.out.println(i);

}

}

}

class Consumer implements Runnable{

Q q;

Consumer(Q q){

this.q = q;

new Thread(this, "Consumer").start();

}

public void run(){

while(true){

q.get();

// System.out.println(q.get()); //to see the result

}

}

}

public class inter\_thread\_comms{

public static void main(String args[]){

Q q = new Q();

new Producer(q);

new Consumer(q);

System.out.println("Press Control-C to stop.");

}

}

**EXPERIMENT 8**

**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.**

//Student

package CIE;

public class Student1{

public String usn;

public String name;

public int sem;

}

//Internals

package CIE;

public class Internals extends Student1{

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

}

//External

package SEE;

import CIE.Student1;

public class External extends Student1{

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

}

//main

import java.util.Scanner;

import CIE.Internals;

import SEE.External;

public class lots\_of\_packages{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

int n, i, j;

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

n = sc.nextInt();

Internals[] inter = new Internals[n];

External[] exter = new External[n];

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

inter[i] = new Internals();

exter[i] = new External();

System.out.println("Enter the details of "+ (i+1)+"th Student");

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

inter[i].usn = sc.next();

exter[i].usn = inter[i].usn;

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

inter[i].name = sc.next();

exter[i].name = inter[i].name;

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

inter[i].sem = sc.nextInt();

exter[i].sem = inter[i].sem;

System.out.println("Enter the internal marks of 5 courses with 3 credits each: ");

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

inter[i].marks[j] = sc.nextInt();

System.out.println("Enter the External marks of 5 subjects with 3 credits each: ");

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

exter[i].marks[j] = sc.nextInt();

}

System.out.println("Details of students with their final marks are: ");

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

System.out.println("Student "+ (i+1)+ ": ");

System.out.println("USN: "+ inter[i].usn);

System.out.println("Name: " + inter[i].name);

System.out.println("Semester: "+ inter[i].sem);

System.out.println("Final Marks: ");

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

System.out.println((j+1)+ "th subject: "+ ((inter[i].marks[j]+exter[i].marks[j])/2));

}

}

}