VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on
OBJECT ORIENTED JAVA PROGRAMMING LAB

Submitted by

RACHANA A (1BM21CS154)

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 (21CS3PCOOJ)" carried out by RACHANA A **(1BM21CS154),** 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 a Object Oriented Java Programming (21CS3PCOOJ) work prescribed for the said degree.

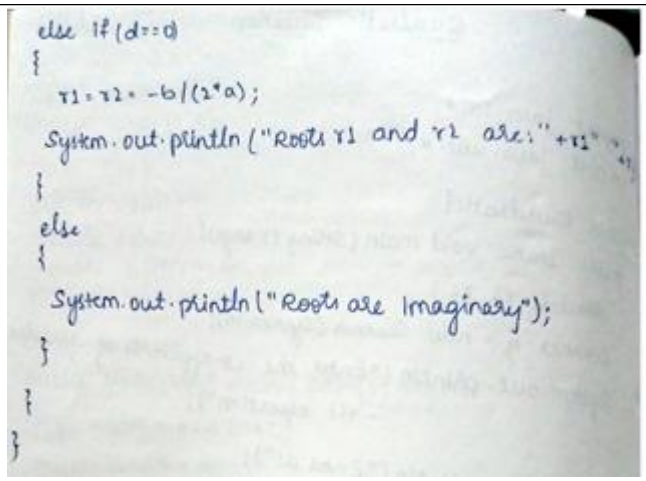
Basavaraj Jakkali
Associate professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

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.

Program-

```
import java.io.*;
import java.util.*;

class Quadratic{
public static void main (String [] args){
  double r1, r2 ;
  Scanner q = new Scanner (System.in);
  System.out.println ("Enter the co-efficients of quadra
                      -tic equation");

  System.out.println ("Enter a:");
   int a = q.nextInt();
  System.out.println ("Enter b:");
   int b= q.nextInt();
  System.out.println ("Enter c:");
   int c = q.nextInt();
   if (a==0){
  System.out.println ("Enter valid Inputs");
  }
  double d= Math.pow(b,2) - (4*a*c);
   if (d>0)
  {
    r1 = (-b + Math.sqrt (d)) / (2*a);
    r2 = (-b - Math.sqrt (d)) / (2*a);
   System.out.println ("Roots r1 and r2 are:"
                       +r1 "  " +r2);
  }
```

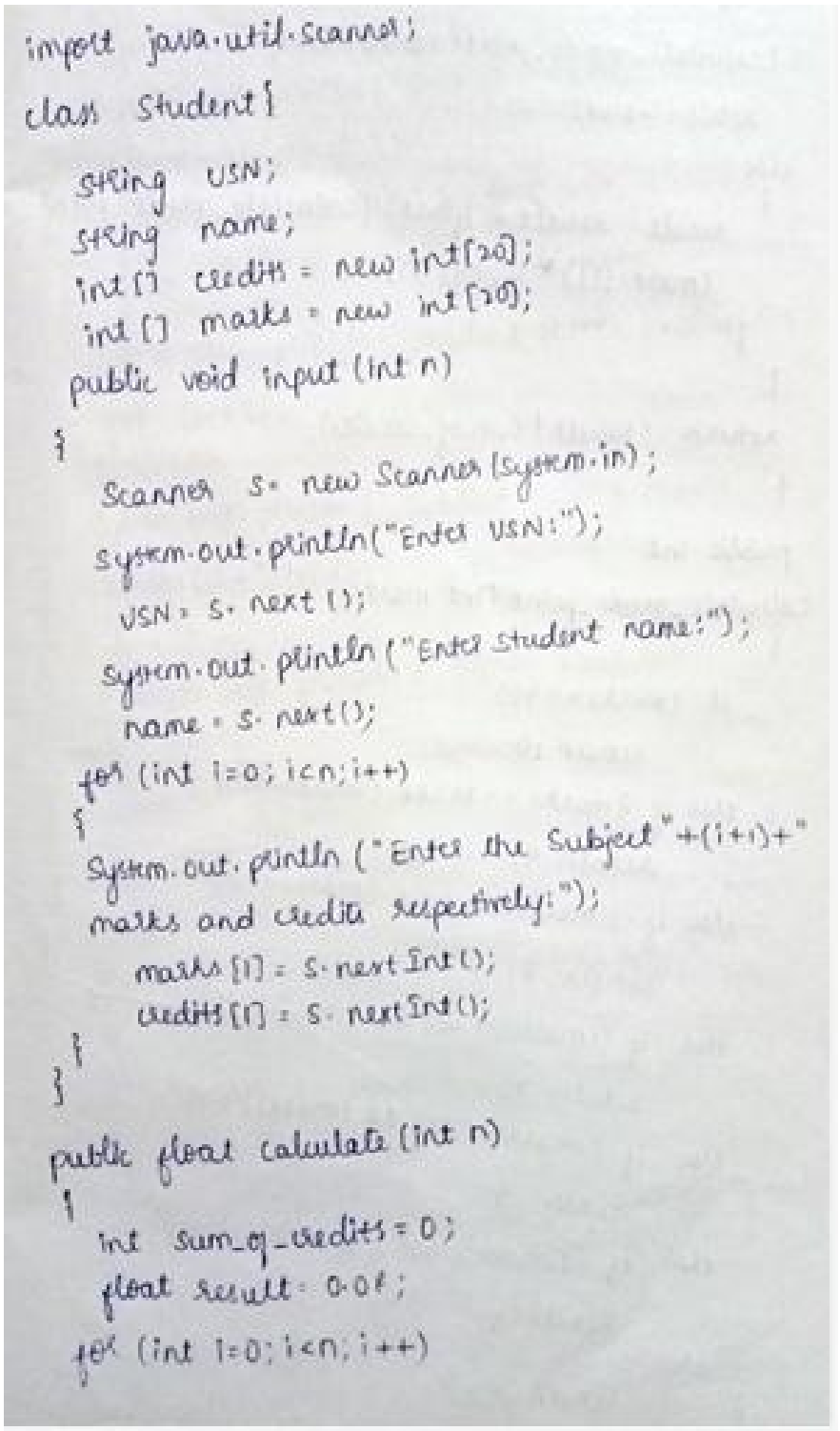


```
else if(d==0)
{
  r1=r2=-b/(2*a);
 System.out.println("Roots r1 and r2 are:"+r1+" ");
}
else
{
 System.out.println("Roots are Imaginary");
}
}
}
```

OUTPUT:

```
Enter the co-efficients of quadratic equation
Enter a:
10
Enter b:
2
Enter c:
5
Roots are imaginary
```

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 Student{
    String USN;
    String name;
    int[] credits = new int[20];
    int[] marks = new int[20];
    public void input(int n)
    {
        Scanner s= new Scanner(System.in);
        System.out.println("Enter USN:");
        USN = s.next();
        System.out.println("Enter student name:");
        name = s.next();
        for(int i=0;i<n;i++)
        {
        System.out.println("Enter the Subject"+(i+1)+"
        marks and credits respectively:");
            marks[i] = s.nextInt();
            credits[i] = s.nextInt();
        }
    }
    public float calculate(int n)
    {
        int sum_of_credits = 0;
        float result = 0.0f;
        for(int i=0;i<n;i++)
```

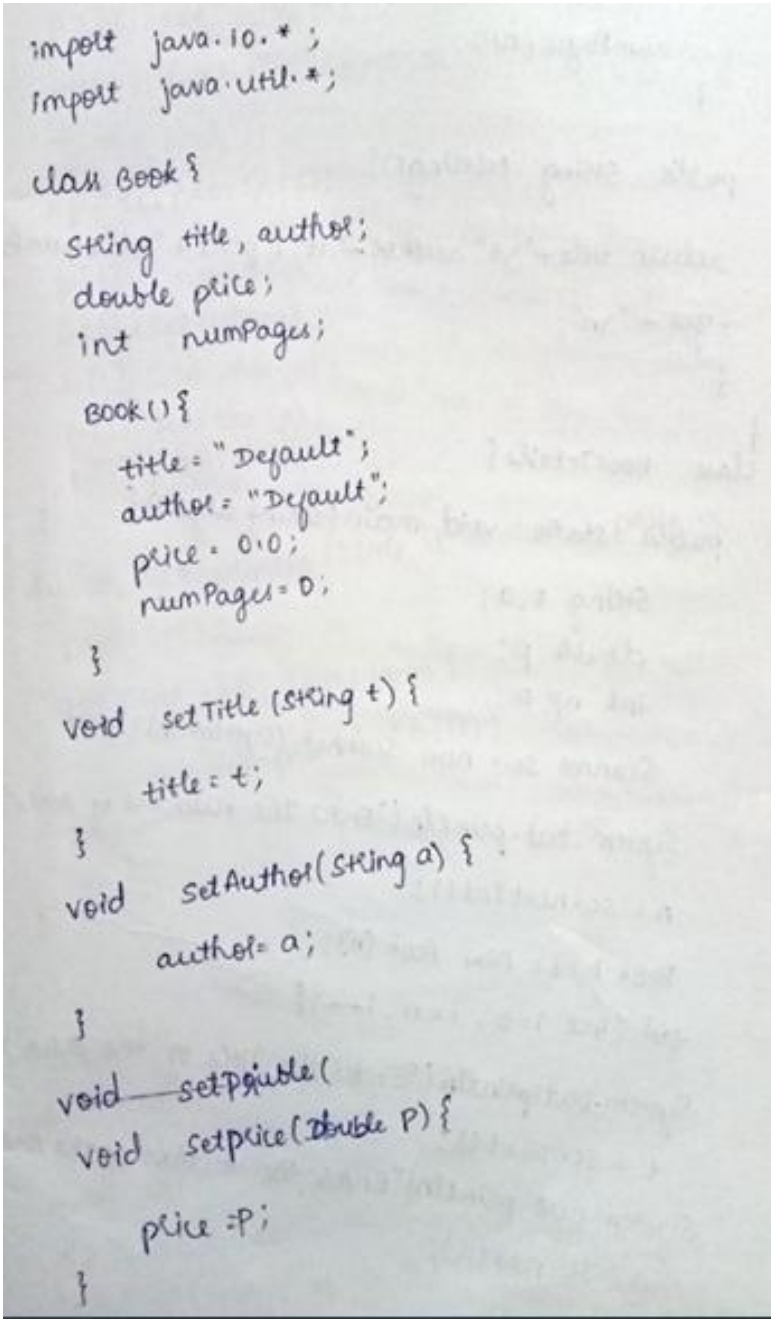
```
}
sum_of_credits += credits[i];
if (calculate_grade_point(marks[i]) == -1)
    return -1.0f;
else
{
    result = result + (float) (calculate_grade_point
    (marks[i]) * credits[i]);
}
}
return (result / sum_of_credits);
}
public int
calculate_grade_point (int marks)
{
    if (marks >= 90)
        return 10;
    else if ((marks >= 80) && (marks < 90))
        return 9;
    else if ((marks >= 70) && (marks < 80))
        return 8;
    else if ((marks >= 60) && (marks < 70))
        return 7;
    else if ((marks >= 50) && (marks < 60))
        return 6;
    else if ((marks >= 40) && (marks < 50))
        return 5;
    else
        return -1;
```

```
public void display(int n, float result)
{
  System.out.println("\n");
  System.out.println("student details");
  System.out.println();
  System.out.println("student USN:" + USN);
  System.out.println("student name:" + name);
  System.out.println("student marks and credits");
  for (int i=0; i<n; i++)
  {
  System.out.println("subject ---->\t marks: "+
  marks[i] +" credit: " + credits[i]);
  }
  System.out.println("SGPA:" +result);
  }
}
public class SgpaCalc
{
  public static void main(String[] args)
  {
      Scanner s = new Scanner(System.in);
      Student s1 = new Student();
      System.out.println("Enter the number of subje
                          cts:");
       int n= s.nextInt();
        s1.input(n);
```

```
    float result = S1.calculate(n);
    if (result==-1.0f)
    {
     System.out.println();
    System.out.println("The Student has failed in a
      Subject. SGPA cannot be calculated:");
      System.exit(0);
    }
    S1.display(n, result);
  }
}
```

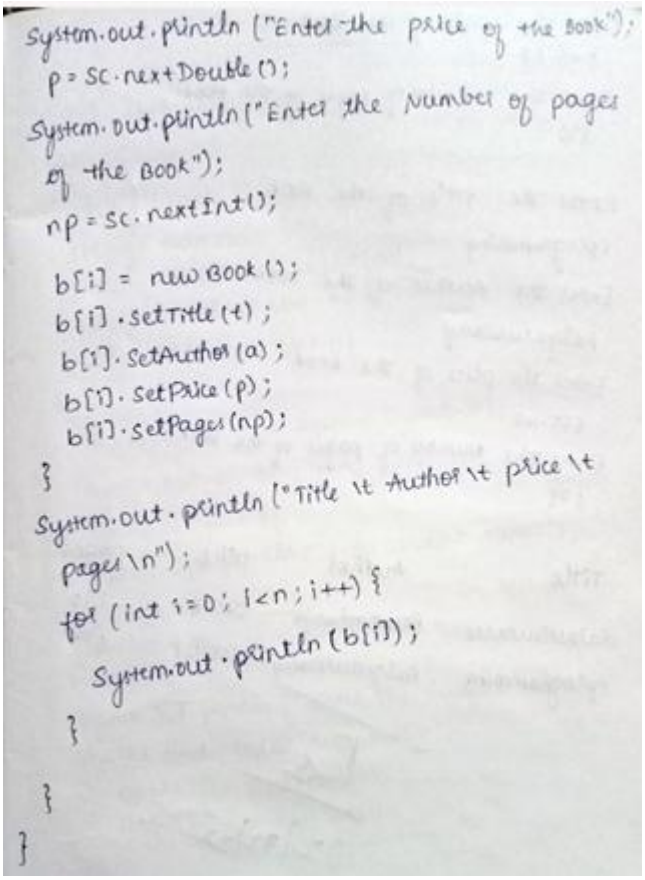
```
y_86fe863b\bin   Lab_02_SGPA
Enter the number of subjects: 2
Enter Student USN: 1
Enter Student Name: a
Enter the Subject 1 marks and credits respectively: 88 2
Enter the Subject 2 marks and credits respectively: 90 3


Student Details

Student USN: 1
Student Name: a
Student Marks and Credits
Subject 1 -->   Marks: 88 Credits: 2
Subject 1 -->   Marks: 90 Credits: 3
SGPA: 9.6
```

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.io.*;
import java.util.*;

class Book {
  String title, author;
  double price;
  int numPages;

  Book() {
    title = "Default";
    author = "Default";
    price = 0.0;
    numPages = 0;
  }
  void setTitle(String t) {
    title = t;
  }
  void setAuthor(String a) {
    author = a;
  }
  void setPrice(double P) {
    price = P;
  }
```

```
void setPages (int np) {
    numPages = np;
  }
  public String toString(){
  return title + "\t" author + "\t" + price + "\t" + numPa
  -ges + "\n";
  }
}
class BookDetails {
  public static void main(String args[]) {
      String t,a;
      double p;
      int np, n;
      Scanner sc = new Scanner (System.in);
    System.out.println("Enter the number of Book
    n = sc.nextInt();
    Book b[] = new Book [n];
    for (int i=0; i<n; i++){
    System.out.println("Enter the Title of the Book"
     t = sc.next();
    System.out.println("Enter the Author of the Book
     a = sc.next();
```



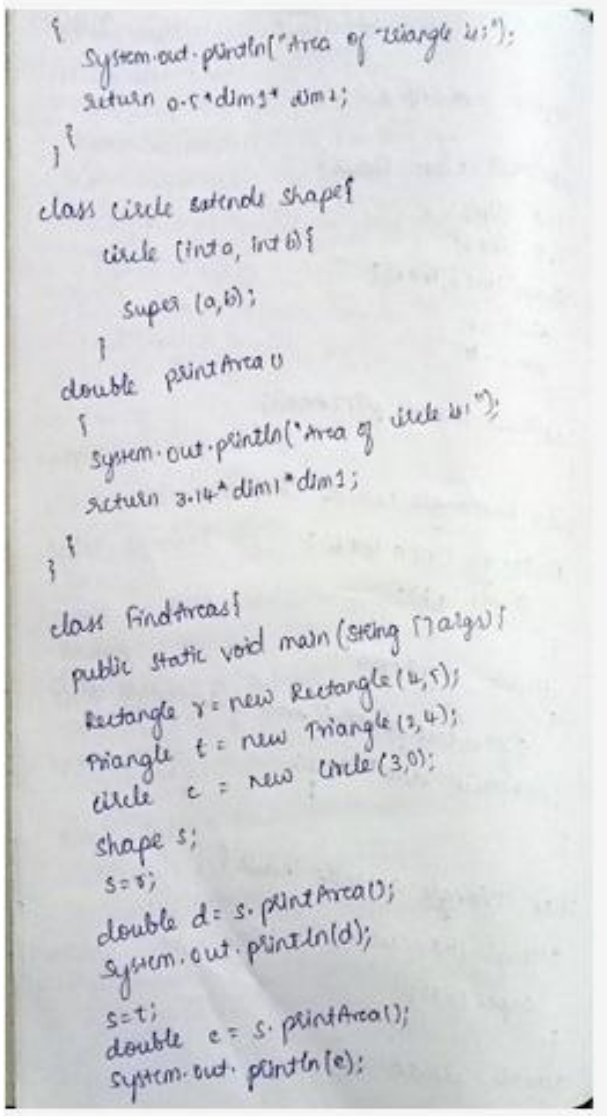
```
System.out.println ("Enter the price of the Book");
 p = sc.nextDouble();
System.out.println("Enter the Number of pages
 of the Book");
 np = sc.nextInt();

 b[i] = new Book();
 b[i].setTitle(t);
 b[i].SetAuthor(a);
 b[i].setPrice(p);
 b[i].setPages(np);
}
System.out.println("Title \t Author \t price \t
 pages \n");
 for (int i=0; i<n; i++) {
   System.out.println(b[i]);
 }

}
}
```

OUTPUT:

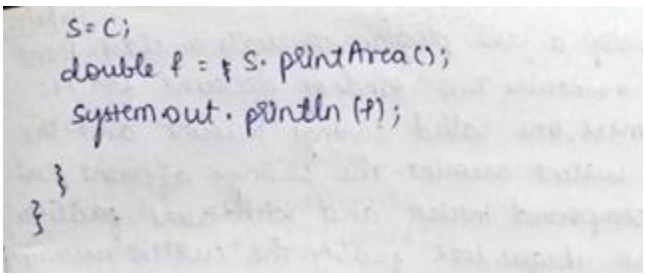
```
Enter the Price of  the Books
100
Enter the Number of pages of  the Books
395
Enter the Title of  the Books
k
Enter the Author of  the Books
gg
Enter the Price of  the Books
398
Enter the Number of pages of  the Books
34
Title    Author          Price   Pages

a       bb      100.0   395

k       gg      398.0   34
```

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.

```
import java.util.*;

abstract class Shape{
 int dim1;
 int dim2;
 Shape (int a, int b){
  dim1 = a;
  dim2 = b;
 }
 abstract double printArea();
}
class Rectangle extends Shape{
 Rectangle (int a, int b){
   super (a,b);
 }
 double printArea()
 {
   System.out.println("Area of rectangle is:");
   return dim1*dim2;
 }
}
class Triangle extends Shape{
 Triangle (int a, int b){
   super (a,b);
 }
 double printArea()
```



```
{
  System.out.println("Area of Triangle is:");
  return 0.5*dim1*dim2;
}
}
class Circle extends Shape{
    Circle (int a, int b){
      super (a,b);
    }
 double printArea()
  {
   System.out.println("Area of circle is:");
   return 3.14*dim1*dim1;
}
}
class FindAreas{
 public static void main (String [] args){
  Rectangle r = new Rectangle(4,5);
  Triangle t = new Triangle(2,4);
  Circle c = new Circle(3,0);
  Shape s;
   s=r;
   double d= s.printArea();
   System.out.println(d);
   s=t;
   double e = s.printArea();
   System.out.println(e);
```



```
   s=c;
   double f = s.printArea();
   System.out.println(f);
  }
}
```

OUTPUT:

```
Area of rectangle is:
20.0
Area of triangle is:
4.0
Area of circle is:
28.259999999999998
```

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:

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

```
import java.util.Scanner;

class Account {
    String customerName;
    int accountNumber;
    String typeOfAccount;
    double balance;

    Account (String customerName, int accountNumb
     String typeOfAccount) {
     this.customerName = customerName;
     this.accountNumber = accountNumber;
     this.typeOfAccount = typeOfAccount;
    }
}
class SavAcc extends Account {
    Scanner sc = new Scanner (System.in);
    SavAcc (String customerName, int accountNumb
String typeOfAccount) {
     super (customer name, accountNumber, typeOfAccount)

    }
    void details() {
     System.out.println();
System.out.println ("Name: " + customerName);
System.out.println ("Acc no.: " + accountNumber);
System.out.println ("Type: " + typeOfAccount);
System.out.println ("Balance: " + balance);
```

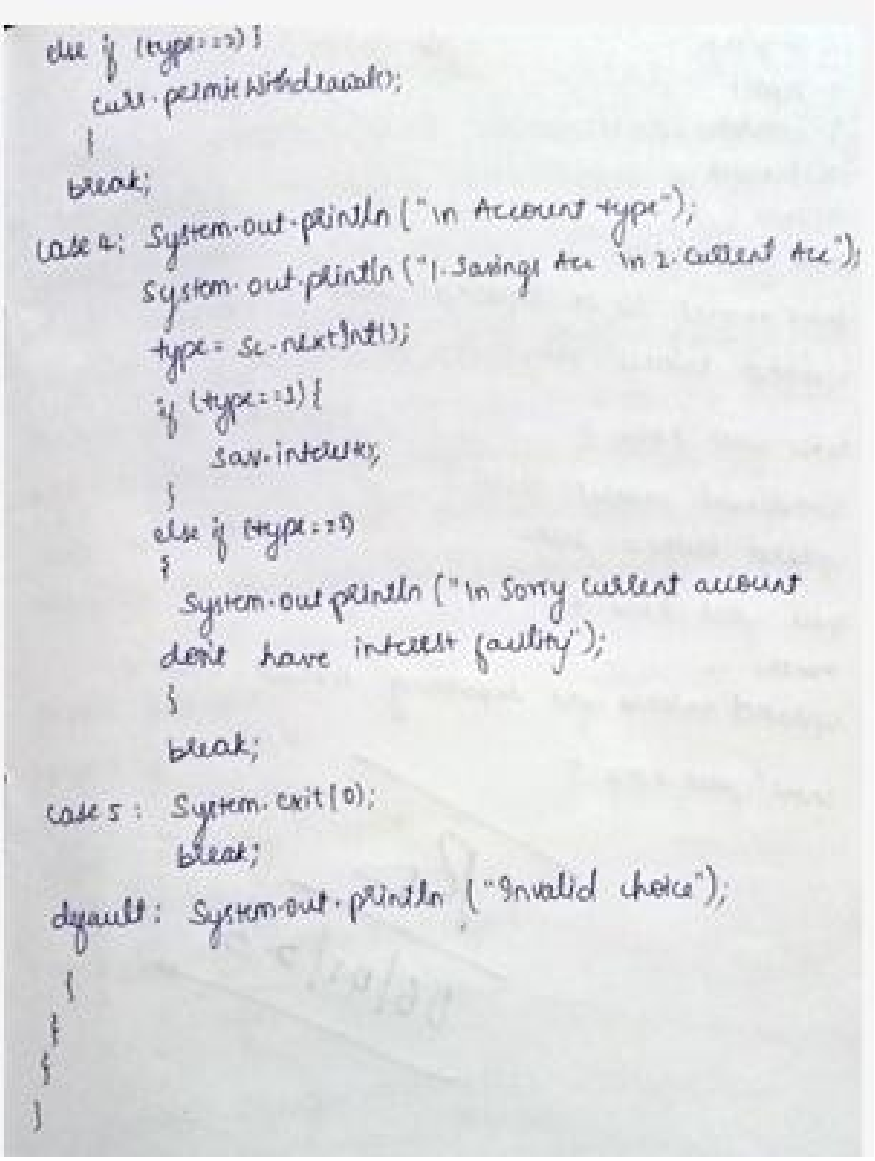
```
System.out.println("Minimum Balance: NO minimum
Balance for Savings Account");
}

void acceptDeposit(){
   System.out.println();
  System.out.println("Enter amount to be depos:
                      -ited");

   double deposit = sc.nextDouble();
    balance += deposit;
 System.out.println("updated Balance:" + balance);
}

void permitWithdrawal(){
   System.out.println();
  System.out.println("withdrawal amount");
   double withdraw = sc.nextDouble();
    if (balance == 0){
    System.out.println("zero balance");
      return;
    }
     balance -= withdraw;
System.out.println("Updated Balance: " + balo
                                    .nce);
}
```

```
void interest(){
  System.out.println("months");
  double months= sc.nextInt();
  balance = balance +( balance * 0.10 * (months/12));
 System.out.println("updated Balance after
  depositing interest : " +balance);
   }
}
class CurAcc extends Account {
     double minBalance;
    Scanner sc = new Scanner(System.in);
 CurAcc( String customerName, int accountNumber,
 String typeOfAccount, double minBalance) {
  super(customerName, accountNumber, typeOfAccount);
  this.minBalance = minBalance;
}
void details(){
 System.out.println();
System.out.println("Name: " + customerName);
System.out.println("Acc no.: " + accountNumber);
System.out.println("Type: " +typeOfAccount);
System.out.println("Balance: " + balance);
System.out.println("Minimum Balance: " + minBalance);
}
```

```
void acceptDeposit(){
System.out.println();
System.out.println("Enter amount to be deposited");
double deposit = sc.nextDouble();
 balance += deposit;
System.out.println("Update Balance: "+balance);
}

void permitWithdrawal(){
System.out.println();
System.out.println("Enter withdrawal amount");
double withdraw= sc.nextDouble();
 if (balance==0){
System.out.println("zero Balance");
 return;
}
balance -= withdraw;
 if (balance<withdraw)
{
   System.out.println();
 System.out.println("oops!! balance is less than
 minimum balance");
System.out.println("You have to pay penalty of
 Rs" +1000);
      balance -=1000;
System.out.println("updated balance after
deducting penalty: " + balance);
}       return;
```

```
        System.out.println("Updated Balance : " +
        balance);
    }
}

class Main{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int type;
        System.out.println("Enter account holder name:");
        String name = sc.nextLine();
        System.out.println("Enter SavingsAccount number:");
        int AccNo1 = sc.nextInt();
        SavAcc sav = new SavAcc(name, AccNo1, "SavingsAccount");
        System.out.print("Enter CurrentAccount number:");
        int AccNo2 = sc.nextInt();
System.out.println("Enter min balance to be maintained
                    in CurrentAccount: ");
        double min = sc.nextDouble();
        CurAcc cur = new CurAcc(name, AccNo2, "Current
        Account", min);
        while (true) {
```

```
System.out.println("\n1.Display\n2.Deposit\n3.Withdraw
                    \n4.Interest\n5.exit");
System.out.print("Enter your choice");
int ch = sc.nextInt();
Switch(ch)
{
  case 1 : System.out.println ("Select Account type");
           System.out.println ("1.Savings Acc\n2.Current A/c");
           int type = sc.nextInt();
           if (type == 1){
              sav.details();
           }
           else if (type == 2)
           {
              curr.details();
           }
           break;
  case 2 : System.out.println ("select Account type");
           System.out.println("1.Savings Acc\n2.Current A/c"
           type = sc.nextInt();
           if (type == 1){
             sav.acceptDeposit();
           }
           else if (type == 2){
             curr.acceptDeposit();
           }
           break;
  case 3 : System.out.println ("Select Account type");
           System.out.println ("1.Savings Acc\n2.Current A/c
           type = sc.nextInt();
           if (type == 1){
             sav.permitWithdrawal();
           }
```



```
else if (type==2){
    curr.permitWithdrawal();
}
break;
case 4: System.out.println("\n Account type");
        System.out.println("1.Savings Acc \n 2. Current Acc");
        type = sc.nextInt();
        if (type==1){
            sav.interest;
        }
        else if (type==2)
        {
         System.out.println("\n Sorry current account
        dont have interest facility");
        }
        break;
case 5 : System.exit(0);
        break;
default: System.out.println("Invalid choice");
}
}
}
```

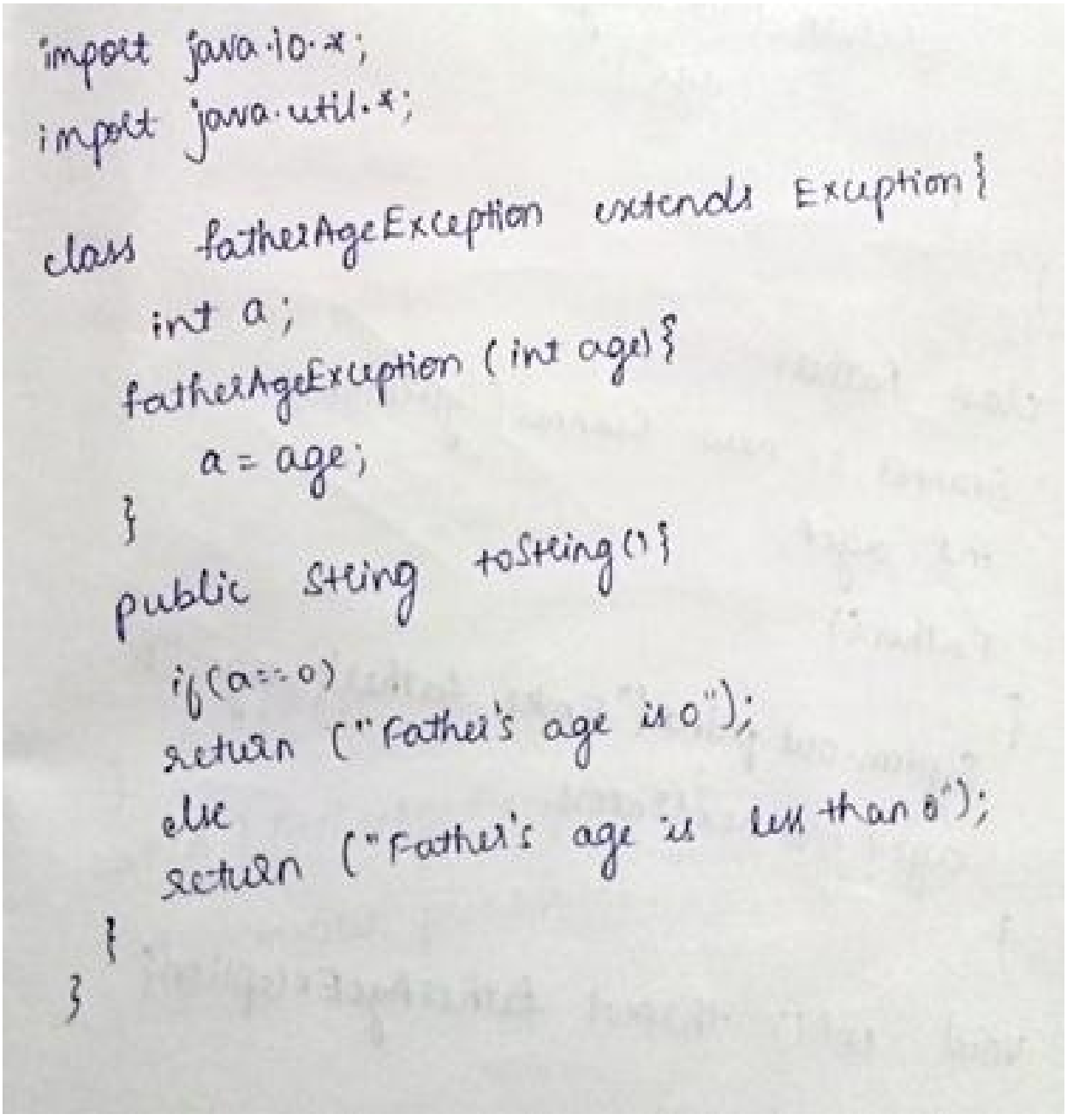
OUTPUT:

```
Enter account holder name: rajesh
Enter SavingsAccount number: 123456789
Enter CurrentAccount number: 345612784
Enter min balance to be maintained in CuurentAccount: 1000

1.Display
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 2
```

```
Account Type
1.Savings Acc
2.Current Acc
1

Enter amount to be deposited
20000
Updated Balance: 20000.0
```

```
Enter your Choice: 3

Account Type
1.Savings Acc
2.Current Acc
1

Withdrawal amount
2000
Updated Balance: 18000.0

1.Display
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 4

Account Type
1.Savings Acc
2.Current Acc
1
Months
12
Updated Balance after depositing interest: 19800.0
```

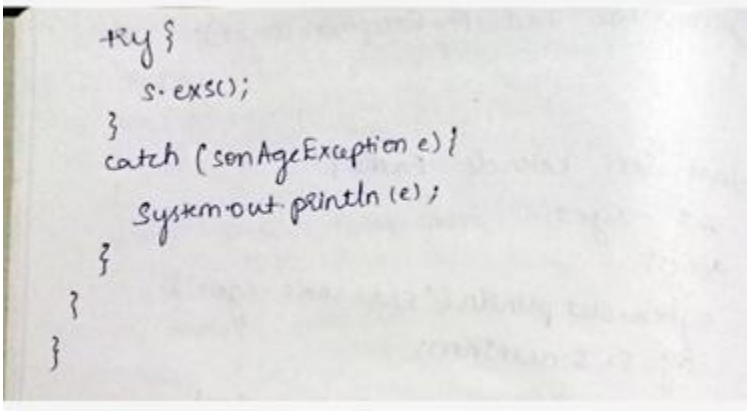
6.Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception Wrongage( ) when the input age<0. In Son 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.io.*;
import java.util.*;

class fatherAgeException extends Exception{
    int a;
    fatherAgeException (int age){
        a = age;
    }
    public string toString(){
        if(a==0)
        return ("father's age is 0");
        else
        return ("father's age is less than 0");
    }
}
```

```
class SonAgeException extends Exception{
    int a;
    SonAgeException (int age){
        a = age;
    }
    public String toString(){
        if (a<0)
        return("Son's age is less than 0");
        else if (a==0)
          return ("son's age is 0");
        else
          return ('Son's age is more than father's
                  age");
    }
}

class Father{
 Scanner s= new Scanner(System.in);
 int agef;
 Father()
{
  System.out.print("Enter father's age:");
   agef = s.nextInt();
}
Void exf() throws FatherAgeException{
    if (agef<=0)
```

```
        throw new fatherAgeException (age1);
    }
}

class Son extends Father{
    int ages;
    Son(){
    System.out.println("Enter son's age:");
      ages = s.nextInt();
    }
    void exs() throws sonAgeException{
        if (ages <=0 || ages>super.age1)
          throw new SonAgeException(ages);
        }
}

class WrongAge{
  public static void main (String [] args)
  {
     Son s= new Son();
     try{
       s.exs();
     }
     catch (fatherAgeException e){
       System.out.println(e)
     }
```



```
    try {
      s.exs();
    }
    catch (sonAgeExcaption e){
      System.out.println(e);
    }
  }
}
```

OUTPUT:

```
Enter Father's age: 35
Enter son's age: -1

Son's age is less than 0

C:\Users\SWETHA\Desktop>java WrongAgeException
Enter Father's age: 10
Enter son's age: 21

Son's age is more than father's age
```

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 SampleThread extends Thread {
    String name;
    Long time;
    SampleThread (String n, Long t){
      name = n;
      time = t;
    }
  public void run(){
    for (int i=0; i<5; i++){
     try{
        System.out.println(name);
        Thread.sleep(time);
      } Sop
     catch( InterruptedException e){
      System.out.println ("Thread is interrupted");
    }
    System.out.println( "Exiting thread:" +name);
  }
}
```

```
class MultithreadingDemo {
  public static void main (String []args){
  SampleThread S1 = new SampleThread (" BMS College of
                    Engineering", 10000);
  SampleThread S2 = new SampleThread ("CSE", 2000);

    S1.start();
    S2.start();

  }
}
```

OUTPUT:

```
C:\Users\BMSCECSEIL74\Desktop\152>javac ThreadDemo.java

C:\Users\BMSCECSEIL74\Desktop\152>java ThreadDemo
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
Exiting Thread[Cse,5,main]
BMS College of Engineering
BMS College of Engineering
Exiting Thread[Bms,5,main]
```