VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Chaithanya Sudhan (1BM23CS073)

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

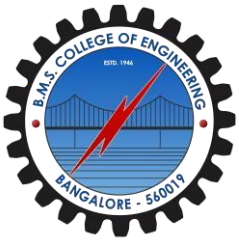
Sep-2024 to Jan-2025

B.M.S. College of Engineering

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering



CERTIFICATE

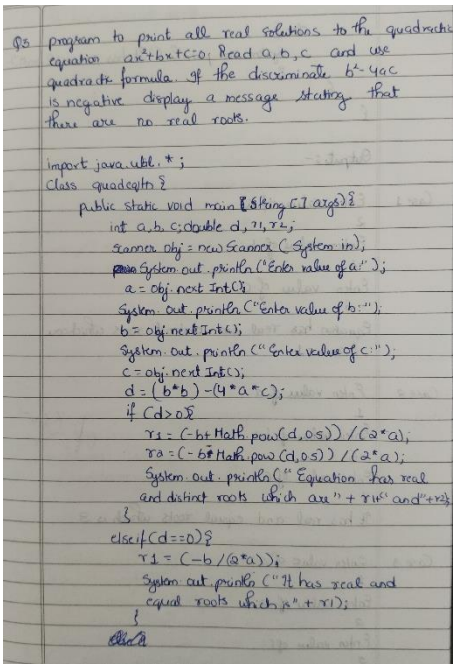
This is to certify that the Lab work entitled "Object Oriented Java Programming (23CS3PCOOJ)" carried out by **Chaithanya Sudhan (1BM23CS073),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

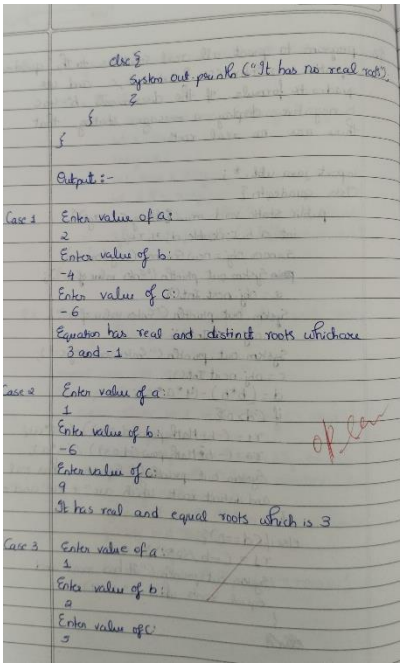
Lab faculty Incharge Name Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

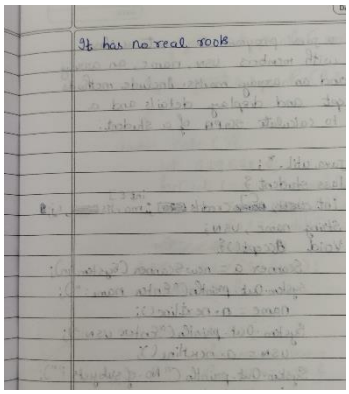
Index

Github Link: https://github.com/chaithanyasudhan/java-LAB

Program 1:
Quadratic Equations







CODE:

```
import java.util.*;

public class Quadeqtn {
  public static void main(String[] args) {
    int a,b, c; double d, r1, r2;
    Scanner obj = new Scanner(System.in);
    System.out.println("Enter value of a:" );
    a = obj.nextInt();
    System.out.println("Enter value of b:");
    b = obj.nextInt();
    System.out.println("Enter value of c:");
    c= obj.nextInt();
    d=(b*b)-(4*a*c);
    if(d>0){
     r1=(-b+ Math.pow(d,0.5))/(2*a);
     r2=(-b-Math.pow(d,0.5)) /(2*a);
```

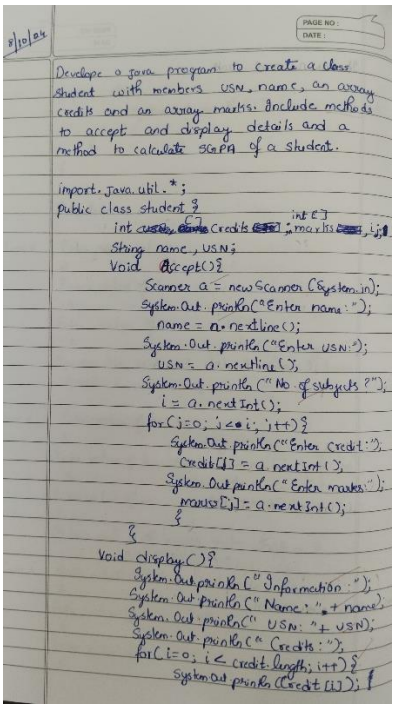
```
 System.out.println("Equation has real and distinct roots which are " + r1+" and"+r2);}
else if(d==0){
 r1=(-b/2*a);
 System.out.println("It has real and equal roots which is" + r1);}
else{
 System.out.println("it has no real roots");}}}
```

OUTPUT:

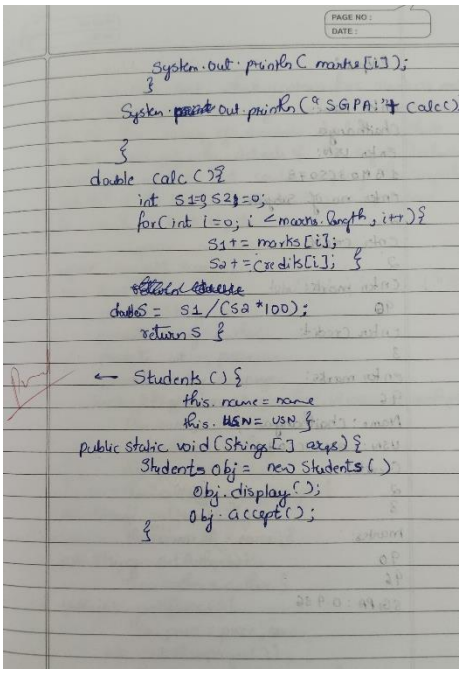
```
C:\Users\chait\OneDrive\Documents\sem 3\java\practiced prgms>javac Quadeqtn.java

C:\Users\chait\OneDrive\Documents\sem 3\java\practiced prgms>java Quadeqtn
Enter value of a:
2
Enter value of b:
-4
Enter value of c:
-6
Equation has real and distinct roots which are 3.0 and-1.0

C:\Users\chait\OneDrive\Documents\sem 3\java\practiced prgms>java Quadeqtn
Enter value of a:
1
Enter value of b:
-6
Enter value of c:
9
It has real and equal roots which is3.0

C:\Users\chait\OneDrive\Documents\sem 3\java\practiced prgms>java Quadeqtn
Enter value of a:
1
Enter value of b:
2
Enter value of c:
5
it has no real roots
```

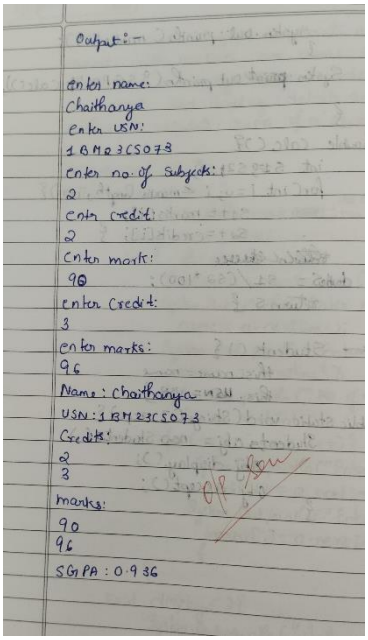
Program 2

Student SGPA


Develope 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.*;
public class student {
    int [] credits; int [] marks; int i,j;
    String name, USN;
    Void accept(){
        Scanner a = new Scanner(System.in);
        System.out.println("Enter name:");
        name = a.nextline();
        System.out.println("Enter USN:");
        USN = a.nextline();
        System.out.println("No of subjects?");
        i = a.nextInt();
        for(j=0; i<0 i; i++){
            System.out.println("Enter credit:");
            Credit[j] = a.nextInt();
            System.out.println("Enter marks:");
            marks[j] = a.nextInt();
        }
    Void display(){
        System.out.println("Information:");
        System.out.println("Name:"+name);
        System.out.println("USN:"+USN);
        System.out.println("Credits:");
        for(i=0; i< credit.length; i++){
            System.out.println(Credit[i]);
```


```
        System.out.println(marks[i]);
    }
    System.out.println("SGPA:"+calc());
}
double calc(){
    int S1=0,S2=0;
    for(int i=0; i<marks.length; i++){
        S1+= marks[i];
        S2+=credits[i]; }
    double S = S1/(S2*100);
    return S }
← Students(){
    this.name = name
    this.USN = USN }
public static void(String[] args){
    Students obj = new Students()
    obj.display();
    obj.accept();
}
```


```
Output:-
enter name:
Chaithanya
Enter USN:
1BM23CS078
enter no. of subjects:
2
enter credit:
2
Enter marks:
90
enter credit:
3
enter marks:
96
Name: Chaithanya
USN: 1BM23CS078
Credits:
2
3
marks:
90
96
SGPA: 0.936
```

CODE:

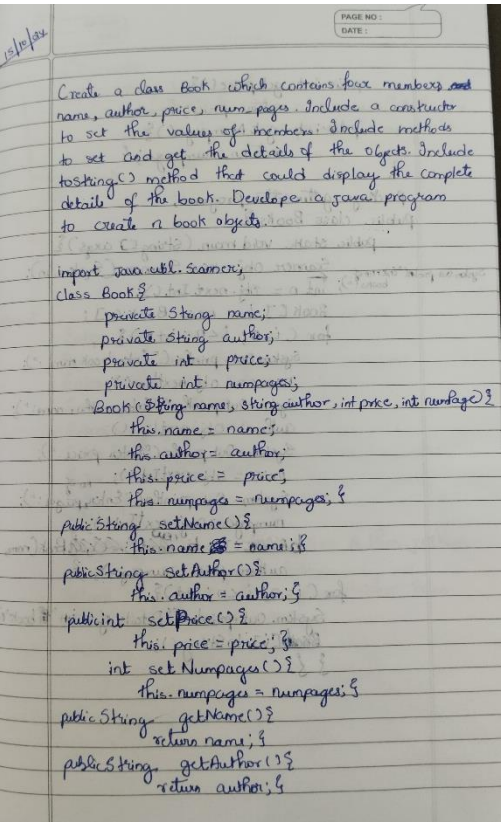
```
import java.util.*;
public class Student{
  int []credits;
  int []marks;
  int i,j;
  String name,USN;
  void accept(){
    Scanner a =new Scanner(System.in);
    System.out.println("enter name:");
    name=a.nextLine();
    System.out.println("enter USN:");
    USN=a.nextLine();
    System.out.println("enter no of subjects:");
    i=a.nextInt();
    marks=new int[i];
    credits=new int[i];
    for(j=0;j<i;j++){
      System.out.println("enter credit:");
      credits[j]=a.nextInt();
      System.out.println("enter marks:");
      marks[j]=a.nextInt();
      } }
   void display(){
     System.out.println("Name:"+name);
     System.out.println("USN:"+USN);
     System.out.println("Credits:");
     for(i=0;i<credits.length;i++){
       System.out.println(credits[i]);}
```

```
    System.out.println("Marks:");
    for(i=0;i<marks.length;i++){
       System.out.println(marks[i]);}
    System.out.println("SGPA:"+calc());}
 double calc(){
    int s1=0,s2=0;
    for(int i=0;i<marks.length;i++){
        s1+=marks[i]*credits[i];
        s2+=credits[i];}
    return (double) s1/(s2*100);}
 Student(){
    this.name=name;
    this.USN=USN;}
 public static void main(String[] args){
     Student obj=new Student();
     obj.accept();
     obj.display();
  }}
```

OUTPUT:

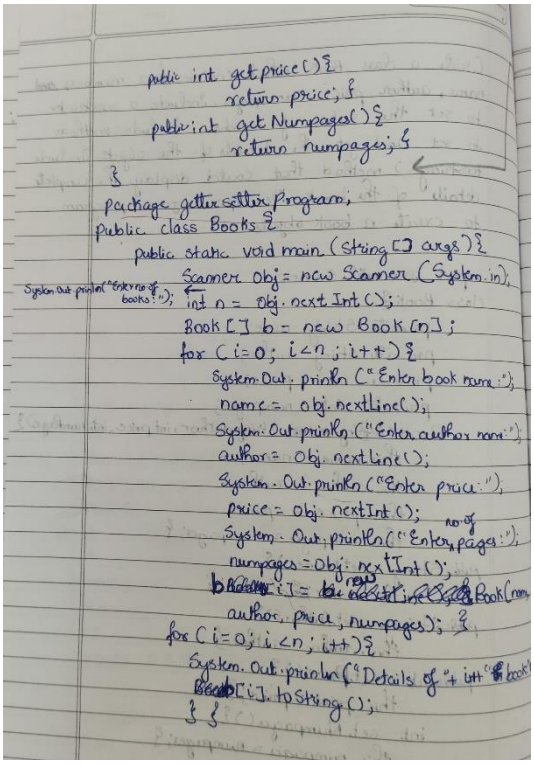
```
C:\Users\chait\OneDrive\Documents\sem 3\java\practiced prgms>javac Student.java

C:\Users\chait\OneDrive\Documents\sem 3\java\practiced prgms>java Student
enter name:
Chaithanya
enter USN:
1BM23CS073
enter no of subjects:
2
enter credit:
2
enter marks:
90
enter credit:
3
enter marks:
96
Name:Chaithanya
USN:1BM23CS073
Credits:
2
3
Marks:
90
96
SGPA:0.936
```

Program 3
Book Details

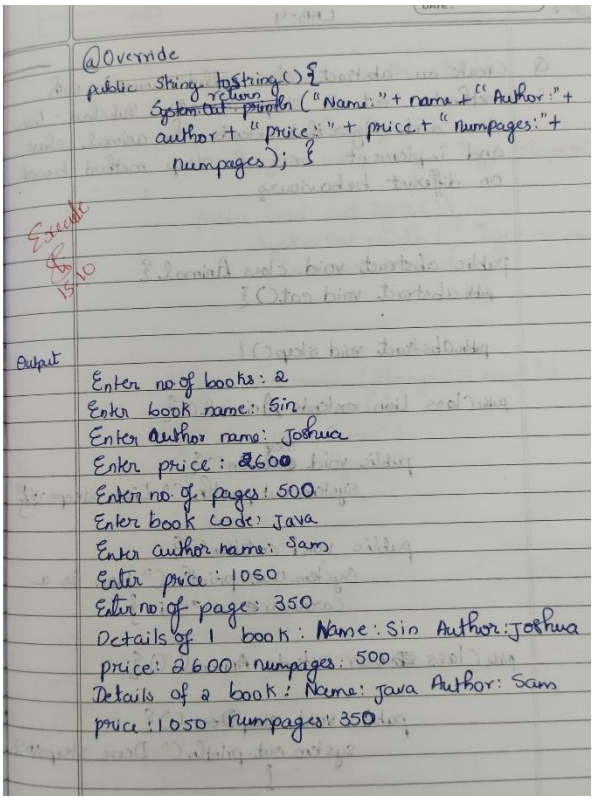


Create a class Book which contains four members name, author, price, num_pages. Include a constructor to set the values of members. Include methods to set and get the details of the objects. Include toString() method that could display the complete details of the book. Develope a java program to create n book objects.

```
import java.util.Scanner;
class Book{
    private String name;
    private String author;
    private int price;
    private int numpages;
    Book(String name, String author, int price, int numPage){
        this.name = name;
        this.author = author;
        this.price = price;
        this.numpages = numpages;}
public String setName(){
        this.name = name;}
public String setAuthor(){
        this.author = author;}
public int setPrice(){
        this.price = price;}
    int setNumpages(){
        this.numpages = numpages;}
public String getName(){
        return name;}
public String getAuthor(){
        return author;}
```



```
    public int getPrice(){
        return price;}
    public int getNumpages(){
        return numpages;}
}
package getterSetterProgram;
public class Books{
    public static void main(String[] args){
        Scanner obj = new Scanner(System.in);
System.out.println("Enter no of books:");
        int n = obj.nextInt();
        Book[] b = new Book[n];
        for(i=0; i<n; i++){
            System.out.println("Enter book name:");
            name = obj.nextLine();
            System.out.println("Enter author name:");
            author = obj.nextLine();
            System.out.println("Enter price:");
            price = obj.nextInt();
            System.out.println("Enter pages:");
            numpages = obj.nextInt();
            b[i] = new Book(name,
            author, price, numpages);}
        for(i=0; i<n; i++){
            System.out.println("Details of "+i+" book");
            b[i].toString();
        }}
```



```
@Override
public String toString(){
    return
    System.out.println("Name:"+name+"Author:"+
    author+"price:"+price+"numpages:"+
    numpages);}
```

Execute
15/10

Output

```
Enter no of books: 2
Enter book name: Sin
Enter author name: Joshua
Enter price: 2600
Enter no. of pages: 500
Enter book code: Java
Enter author name: Sam
Enter price: 1050
Enter no of page: 350
Details of 1 book: Name: Sin Author: Joshua
price: 2600 numpages: 500
Details of 2 book: Name: Java Author: Sam
price: 1050 numpages: 350
```

CODE

```
import java.util.Scanner;

class Book {
    private String name;
    private String author;
    private int price;
    private int numPages;

    Book(String name, String
author, int price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public void setName(String
name) {
        this.name = name;
    }

    public void setAuthor(String
author) {
        this.author = author;
    }

    public void setPrice(int price) {
        this.price = price;
    }

    public void setNumPages(int
numPages) {
        this.numPages = numPages;
    }

    public String getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }

    public int getPrice() {
        return price;
    }

    public int getNumPages() {
        return numPages;
    }
```

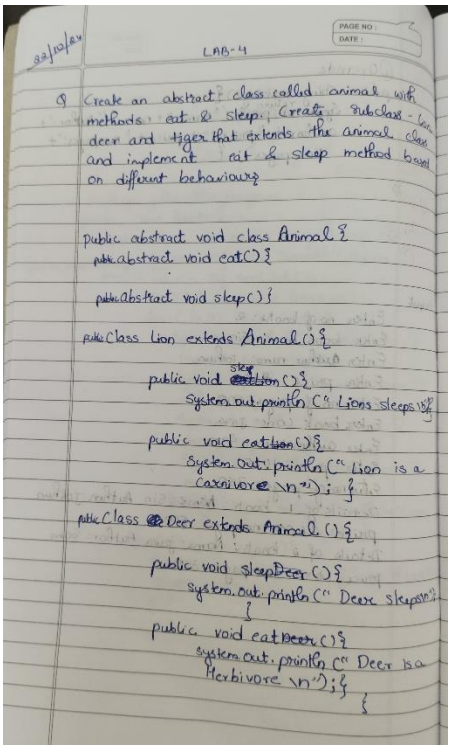
```
    @Override
    public String toString() {
        return "Name: " + name + ",
Author: " + author + ", Price: " +
price + ", Pages: " + numPages;
    }
}

public class Books {
    public static void main(String[]
args) {
        Scanner obj = new
Scanner(System.in);

        System.out.println("Enter the
number of books:");
        int n = obj.nextInt();
        obj.nextLine();

        Book[] b = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter
details for book " + (i + 1) + ":");
            System.out.println("Enter
book name:");
            String name =
obj.nextLine();
            System.out.println("Enter
author name:");
            String author =
obj.nextLine();
            System.out.println("Enter
price:");
            int price = obj.nextInt();
            System.out.println("Enter
number of pages:");
            int numPages =
obj.nextInt();
            obj.nextLine();
            b[i] = new Book(name,
author, price, numPages);
        }

        System.out.println("\nDetails
of all books:");
        for (int i = 0; i < n; i++) {
            System.out.println("Book "
+ (i + 1) + ": " + b[i].toString());
        }
    }
```

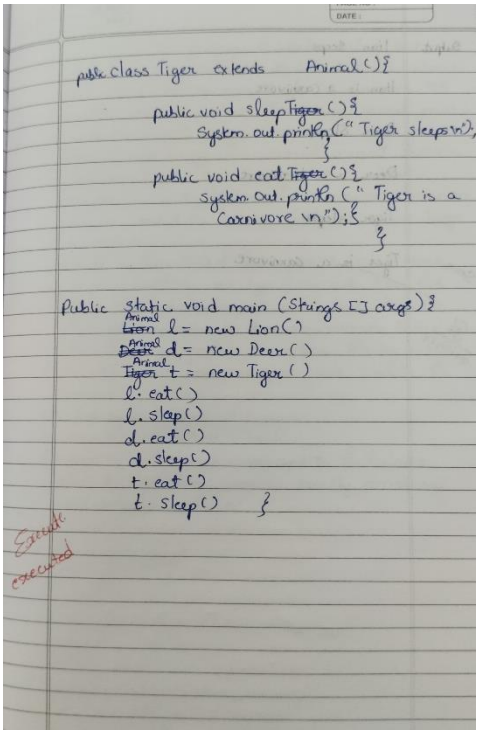
}

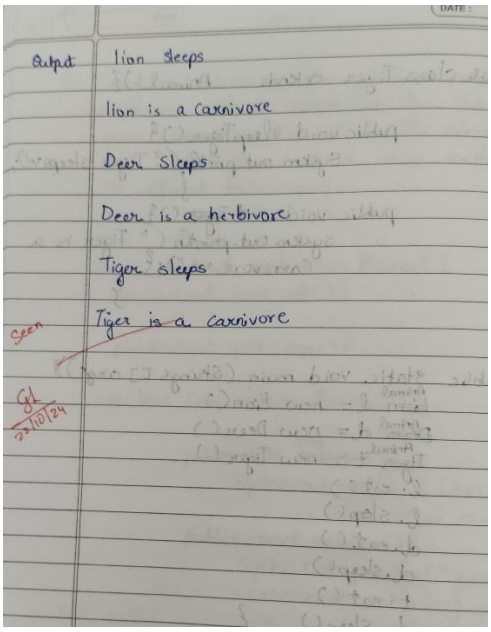
OUTPUT:

```
C:\Users\chait\OneDrive\Documents\sem 3\java\practiced prgms>javac Books
.java

C:\Users\chait\OneDrive\Documents\sem 3\java\practiced prgms>java Books
Enter the number of books:
2
Enter details for book 1:
Enter book name:
Sin
Enter author name:
Joshua
Enter price:
2600
Enter number of pages:
500
Enter details for book 2:
Enter book name:
Java
Enter author name:
Sam
Enter price:
1050
Enter number of pages:
350

Details of all books:
Book 1: Name: Sin, Author: Joshua, Price: 2600, Pages: 500
Book 2: Name: Java, Author: Sam, Price: 1050, Pages: 350
```

Program 4
Animal







CODE

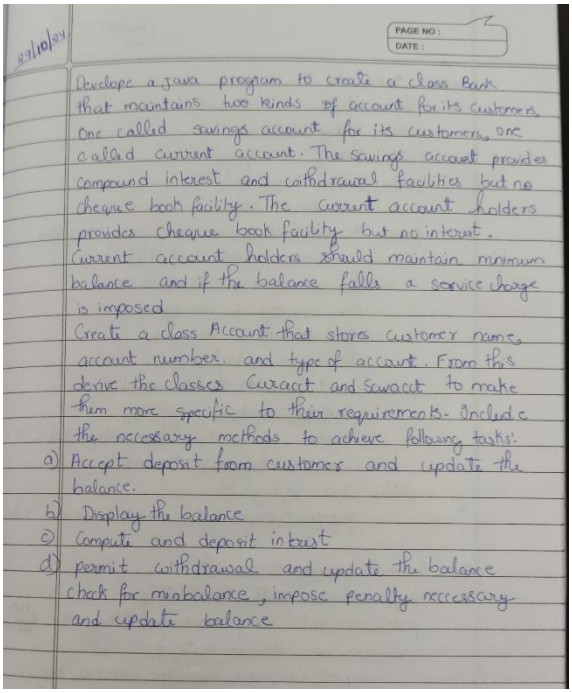
```
abstract class Animal{
  public abstract void sleep();
  public abstract void eat();}
```

```
 class Lion extends Animal{
    public void sleep(){
       System.out.println("lion
sleeps\n");}
    public void eat(){
       System.out.println("lion is
a carnivore \n");}}
 class Tiger extends Animal{
    public void sleep(){
       System.out.println("Tiger
sleeps\n");}
    public void eat(){
       System.out.println("Tiger
is a carnivore \n");}}
 class Deer extends Animal{
    public void sleep(){
       System.out.println("Deer
sleeps\n");}
    public void eat(){
       System.out.println("Deer is
a herbivore \n");}}
public abstract class Animal{
  public abstract void sleep();
  public abstract void eat();
  public static void
main(String[] args){
   Animal l=new Lion();
   Animal d=new Deer();
   Animal t=new Tiger();
   l.sleep();
   l.eat();
   d.sleep();
   d.eat();
   t.sleep();
   t.eat();
}}
```

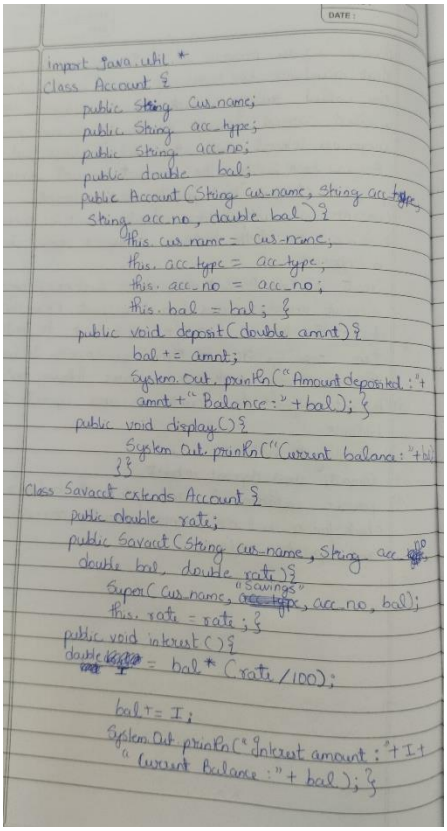
OUTPUT:

```
C:\Users\chait\OneDrive\Documents\sem 3\java\practiced prgms>javac Demo.
java

C:\Users\chait\OneDrive\Documents\sem 3\java\practiced prgms>java Demo
lion sleeps

lion is a carnivore

Deer sleeps

Deer is a herbivore

Tiger sleeps

Tiger is a carnivore
```

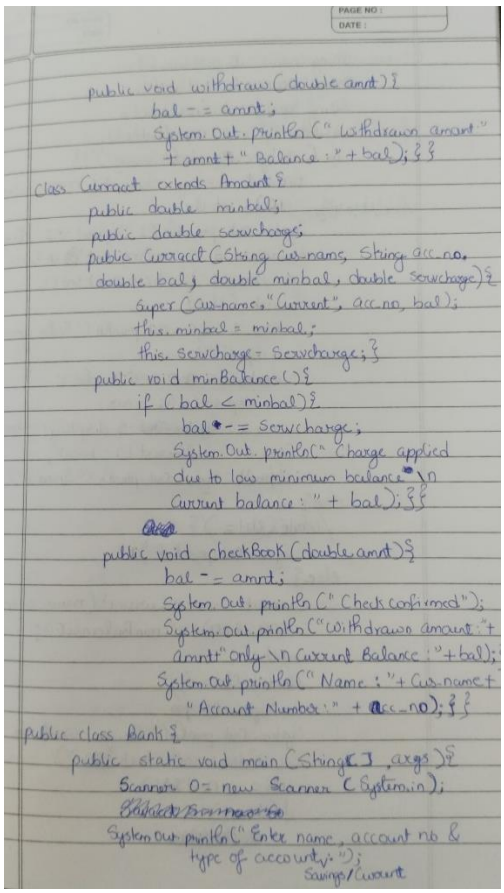
Program 5
Bank



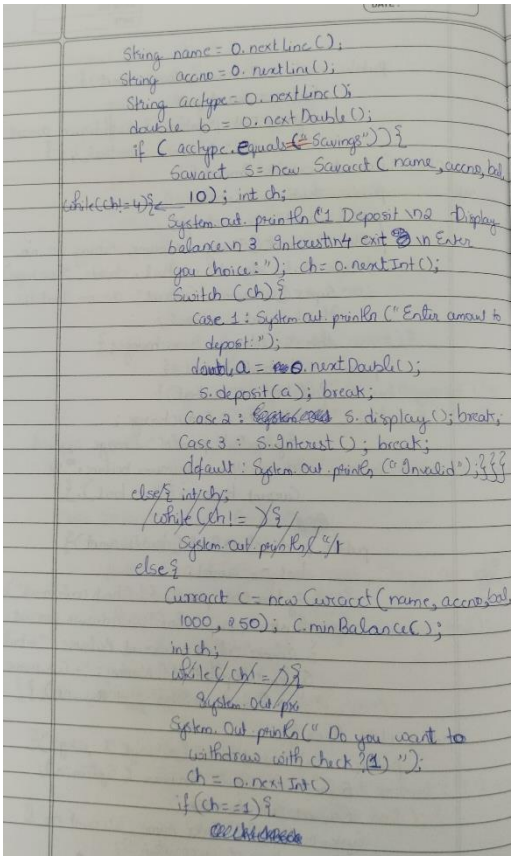
Develope a java program to create a class Bank that maintains two kinds of account for its customers, one called savings account for its customers, one called current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account holders provides cheque book facility but no interest. Current account holders should maintain minimum balance and if the balance falls a service charge is imposed
Create a class Account that stores customer name, account number and type of account. From this derive the classes Curacct and Savacct to make them more specific to their requirements. Include the necessary methods to achieve 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 minbalance, impose penalty necessary and update balance



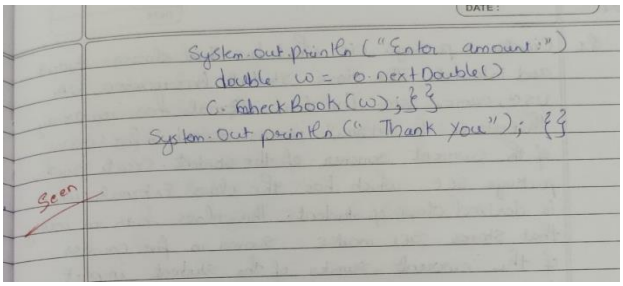
```
import java.util.*
class Account {
    public String cus_name;
    public String acc_type;
    public String acc_no;
    public double bal;
    public Account(String cus_name, String acc_type,
    String acc_no, double bal){
        this.cus_name = cus_name;
        this.acc_type = acc_type;
        this.acc_no = acc_no;
        this.bal = bal; }
    public void deposit(double amnt){
        bal += amnt;
        System.out.println("Amount deposited:"+
        amnt+"Balance:"+bal); }
    public void display(){
        System.out.println("Current balance:"+bal
    }}
class Savacct extends Account {
    public double rate;
    public Savacct(String cus_name, String acc_no,
    double bal, double rate){
        super(cus_name, "Savings", acc_no, bal);
        this.rate = rate; }
    public void interest(){
    double I = bal*(rate/100);
        bal += I;
        System.out.println("Interest amount:"+I+
        "Current Balance:"+bal); }
```



```
public void withdraw(double amnt){
    bal -= amnt;
    System.out.println("Withdrawn amount"
    + amnt + " Balance :" + bal);}}
class Curracct extends Amount{
public double minbal;
public double servcharge;
public Curracct(String cus_name, String acc_no,
double bal, double minbal, double servcharge){
    super(cus_name, "Current", acc_no, bal);
    this.minbal = minbal;
    this.servcharge = servcharge;}
public void minBalance(){
    if (bal < minbal){
        bal -= servcharge;
        System.out.println(" Charge applied
        due to low minimum balance \n
        Current balance : " + bal);}}
public void checkBook(double amnt){
    bal -= amnt;
    System.out.println(" Check confirmed");
    System.out.println("withdrawn amount :"+
    amnt+"only \n Current Balance :"+bal);
    System.out.println(" Name :"+cus_name+
    "Account Number :" + acc_no);}}
public class Bank{
    public static void main(String[] args){
    Scanner o= new Scanner(System.in);
    System.out.println("Enter name, account no &
    type of account(s: Savings/Current");
```



```
String name = o.nextLine();
String accno = o.nextLine();
String acctype = o.nextLine();
double b = o.nextDouble();
if (acctype.equals("Savings")){
    Savacct s= new Savacct(name,accno,bal
while(ch!=4){ 10); int ch;
    System.out.println("1 Deposit \n2 Display
    balance\n3 Interest\n4 exit \n Enter
    you choice:"); ch= o.nextInt();
    Switch (ch){
    case 1: System.out.println("Enter amount to
    deposit:");
    double a = o.nextDouble();
    s.deposit(a); break;
    Case 2 : s.display(); break;
    Case 3 : s.Interest(); break;
    default : System.out.println("Invalid");}}}
else{ int ch;
/while(ch!= ){
    System.out.println("
else{
    Curracct c= new Curracct(name,accno,bal
    1000, 0.50); c.minBalance();
    int ch;
    while(ch!=){
    System.out.
    System.out.println(" Do you want to
    withdraw with check?(1) ");
    ch = o.nextInt()
    if(ch==1){
```



```
System.out.println("Enter amount:")
double w = o.nextDouble()
C.checkBook(w);}}
System.out.println(" Thank you"); {}
```

Seen

CODE

```
import java.util.Scanner;

class Account {
    public String cus_name;
    public String acc_type;
    public String acc_no;
    public double bal;
```

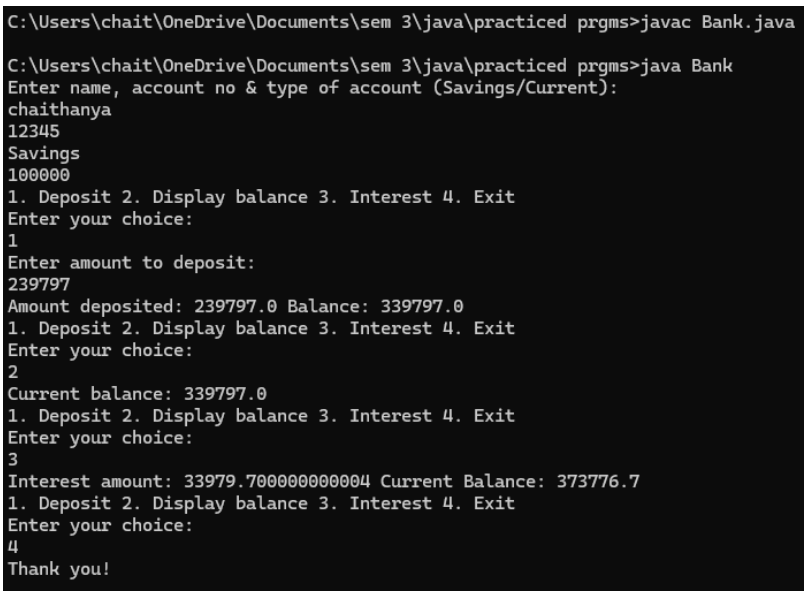
```
    public Account(String
cus_name, String acc_type, String
acc_no, double bal) {
        this.cus_name = cus_name;
        this.acc_type = acc_type;
        this.acc_no = acc_no;
        this.bal = bal;
    }

    public void deposit(double
amnt) {
        bal += amnt;
        System.out.println("Amount
deposited: " + amnt + " Balance: "
+ bal);
    }

    public void display() {
        System.out.println("Current
balance: " + bal);
    }
}

class Savings extends Account {
    public double rate;

    public Savings(String
cus_name, String acc_type, String
acc_no, double bal, double rate) {
        super(cus_name, acc_type,
acc_no, bal);
        this.rate = rate;
    }

    public void interest() {
        double i = bal * (rate / 100);
        bal += i;
        System.out.println("Interest
amount: " + i + " Current Balance:
" + bal);
    }

    public void withdraw(double
amnt) {
        bal -= amnt;

System.out.println("Withdrawn
amount: " + amnt + " Balance: " +
bal);
    }
}
```

```
class Current extends Account {
   public double minbal;
   public double surcharge;

   public Current(String
cus_name, String acc_no, double
bal, double minbal, double
surcharge) {
      super(cus_name, "Current",
acc_no, bal);
      this.minbal = minbal;
      this.surcharge = surcharge;
   }

   public void minBalance() {
      if (bal < minbal) {
         bal -= surcharge;

System.out.println("Charge
applied due to low balance.
Current balance: " + bal);
      }
   }

   public void checkBook(double
amnt) {
      bal -= amnt;
      System.out.println("Check
confirmed");

System.out.println("Withdrawn
amount: " + amnt + " Balance: " +
bal);
      System.out.println("Name: "
+ cus_name + " Account Number:
" + acc_no);
   }
}

public class Bank {
   public static void main(String[]
args) {
      Scanner o = new
Scanner(System.in);

      System.out.println("Enter
name, account no & type of
account (Savings/Current):");
      String name = o.nextLine();
      String accno = o.nextLine();
      String acctype =
o.nextLine();
```

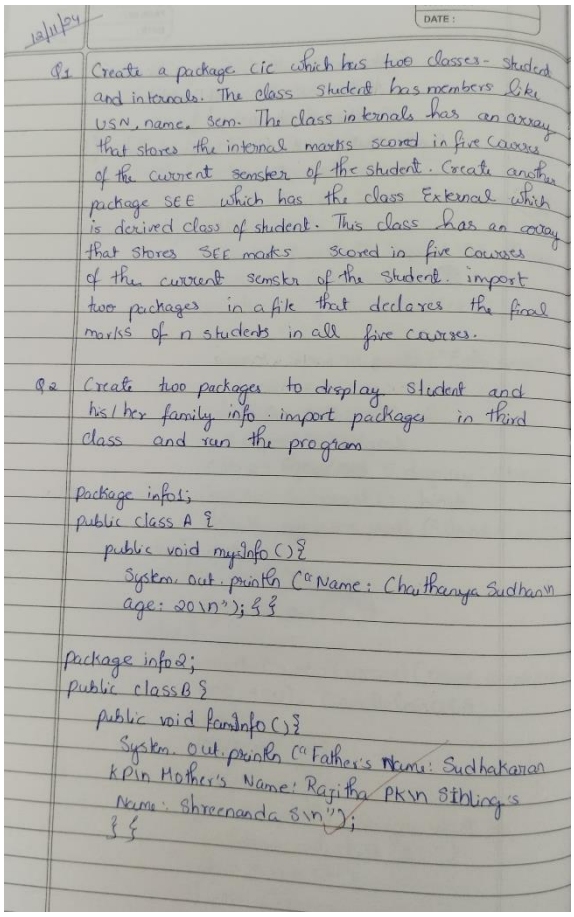
```
        double b = o.nextDouble();

        if
(acctype.equals("Savings")) {
            Savings s = new
Savings(name, acctype, accno, b,
10);

            while (true) {
                System.out.println("1.
Deposit 2. Display balance 3.
Interest 4. Exit");

System.out.println("Enter your
choice: ");
                int ch = o.nextInt();

                switch (ch) {
                    case 1:

System.out.println("Enter amount
to deposit:");
                        double a =
o.nextDouble();
                        s.deposit(a);
                        break;
                    case 2:
                        s.display();
                        break;
                    case 3:
                        s.interest();
                        break;
                    case 4:

System.out.println("Thank you!");
                        return;
                    default:

System.out.println("Invalid
choice");
                }
            }
        } else {
            Current c = new
Current(name, accno, b, 1000,
50);
            c.minBalance();

            System.out.println("Do
you want to withdraw with check
(1)?");
            int ch = o.nextInt();
```

```
            if (ch == 1) {

System.out.println("Enter
amount:");
                double w =
o.nextDouble();
                c.checkBook(w);
            }

            System.out.println("Thank
you");
        }
    }
}
```

OUTPUT



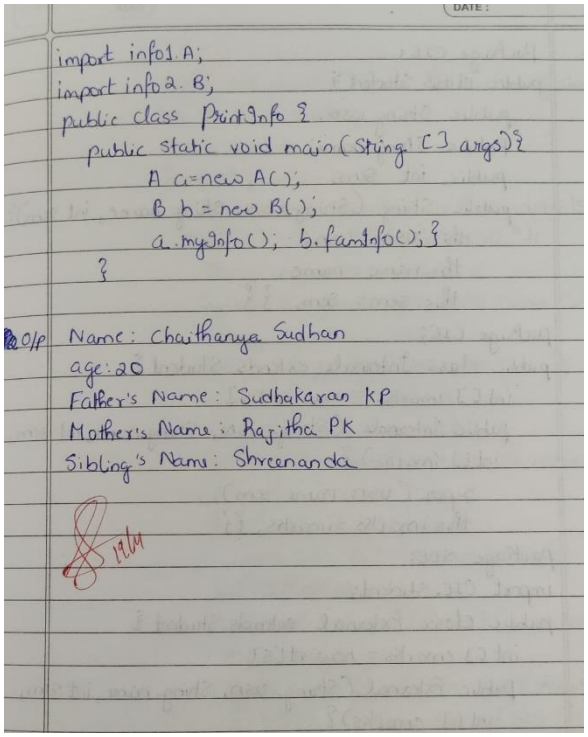
Program 6
Packages



Q1 Create a package cie which has two classes - student and internals. The class student has members like USN, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semster of the student. Create another package SEE which has the class External which is derived class of student. This class has an array that stores SEE marks scored in five courses of the current semster of the student. import two packages in a file that declares the final marks of n students in all five courses.

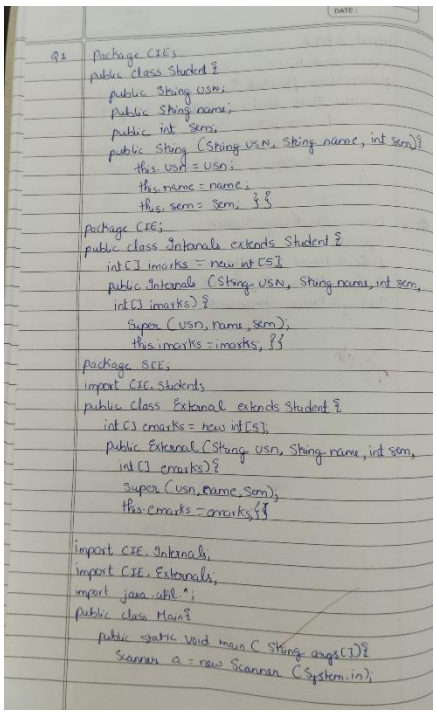
Q2 Create two packages to display student and his/her family info. import packages in third class and run the program

```
package info1;
public class A {
  public void myInfo(){
    System.out.println("Name: Chaithanya Sudhan
    age: 20\n");}}

package info2;
public class B {
  public void famInfo(){
    System.out.println("Father's Name: Sudhakaran
    KP\n Mother's Name: Rajitha PK\n Sibling's
    Name: Shreenanda S\n");
  }}
```

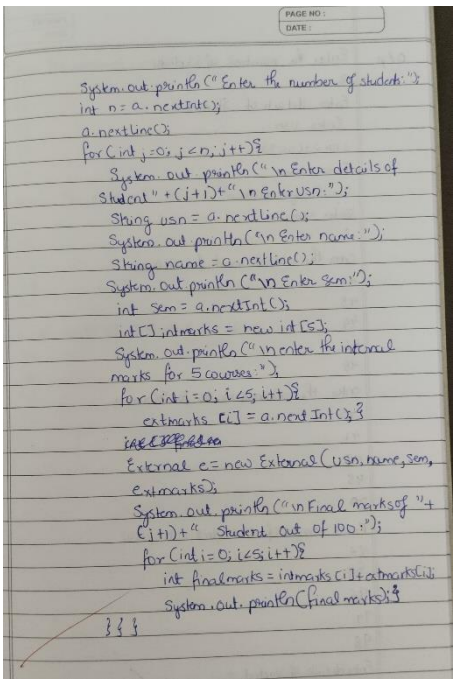


```
import info1.A;
import info2.B;
public class PrintInfo {
  public static void main(String[] args){
      A a=new A();
      B b = new B();
      a.myInfo(); b.famInfo();}
  }
```

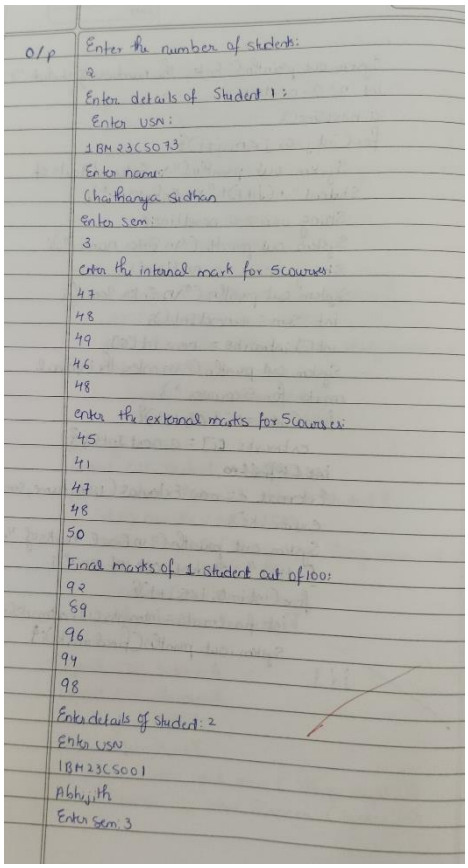
O/p
Name: Chaithanya Sudhan
age: 20
Father's Name: Sudhakaran KP
Mother's Name: Rajitha PK
Sibling's Name: Shreenanda

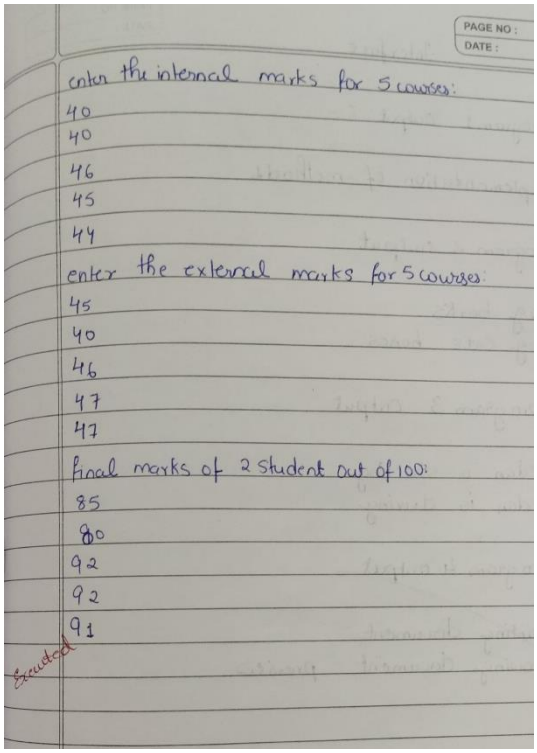


```
Q3 Package CIE;
public class Student {
    public String USN;
    public String name;
    public int Sem;
    public String (String usN, String name, int sem){
      this.usn = usn;
      this.name = name;
      this.sem = sem; }}
package CIE;
public class Internals extends Student {
  int[] imarks = new int[5];
  public Internals (String usN, String name, int sem,
  int[] imarks) {
    super(usn, name, sem);
    this.imarks = imarks; }}
package SEE;
import CIE.Students;
public class External extends Student {
  int[] emarks = new int[5];
  public External(String usn, String name, int sem,
  int[] emarks){
    super(usn, name, sem);
    this.emarks = emarks;}}

import CIE.Internals;
import CIE.Externals;
import java.util.*;
public class Main{
  public static void main(String args[]){
    Scanner a = new Scanner(System.in);
```



```
System.out.println("Enter the number of students:");
int n = a.nextInt();
a.nextLine();
for(int j=0; j<n; j++){
  System.out.println("\n Enter details of
  Student" +(j+1)+"\n Enter USN:");
  String usn = a.nextLine();
  System.out.println("\n Enter name:");
  String name = a.nextLine();
  System.out.println("\n Enter sem:");
  int sem = a.nextInt();
  int[] intmarks = new int[5];
  System.out.println("\n enter the internal
  marks for 5 courses:");
  for(int i=0; i<5; i++){
    extmarks[i] = a.nextInt();}
  External e= new External(usn,name,sem,
  extmarks);
  System.out.println("\n Final marks of "+
  (j+1)+" Student out of 100:");
  for(int i=0; i<5; i++){
    int finalmarks = intmarks[i]+extmarks[i];
    System.out.println(finalmarks);}
}}}
```





CODE

```
package CIE;
public class Student{
   public String usn;
   public String name;
   public int sem;
   public Student(String
usn,String name,int sem){
      this.usn=usn;
      this.name=name;
      this.sem=sem;
  }

}
package CIE;
public class Internals extends
Student{
   int[] imarks=new int[5];
   public Internals(String
usn,String name,int sem,int[]
imarks){
     super(usn,name,sem);
```

```
    this.imarks=imarks;
  }

}
package SEE;
public class Externals extends
Student{
  int[] emarks=new int[5];
  public Internals(String
usn,String name,int sem,int[]
emarks){
    super(usn,name,sem);
    this.emarks=emarks;
  }

}
import CIE.Internals;
import SEE.External;
import java.util.*;
public class Main{
  public static void main(String
args[]){
     Scanner a=new
Scanner(System.in);
     System.out.println("Enter
the number of students:");
     int n=a.nextInt();
     a.nextLine();
     for(int j=0;j<n;j++){

System.out.println("\nEnter
details of Student:"+(j+1)+"\n
Enter usn:");
      String usn=a.nextLine();
a.nextLine();

System.out.println("\nEnter
name:");
      String name=a.nextLine();

System.out.println("\nEnter
sem:");
      int sem=a.nextInt();
      int[] intmarks=new int[5];

System.out.println("\nenter the
internal marks for 5 courses:");
      for(int i=0;i<5;i++){
```

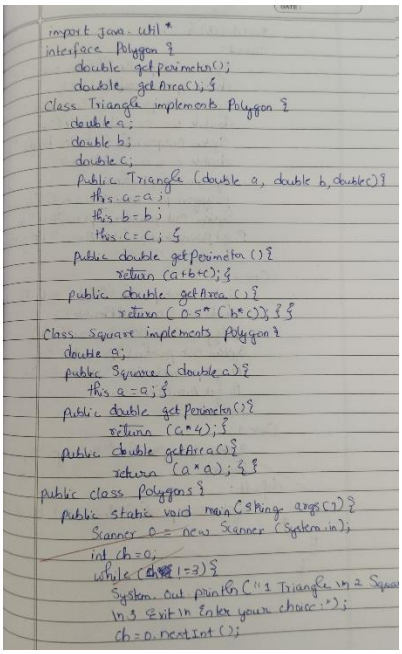
```
        intmarks[i]=a.nextInt();
  }
      int[] extmarks=new int[5];

System.out.println("\nenter the
external marks for 5 courses:");
      for(int i=0;i<5;i++){
        extmarks[i]=a.nextInt();
  }
      External e=new
External(usn,name,sem,extmar
ks);

System.out.println("\nFinal
marksof "+(j+1)+" student out
of 100:");
      for(int i=0;i<5;i++){
        int
finalmarks=intmarks[i]+extmar
ks[i];

System.out.println(finalmarks);
   }

  }

 }
}
```

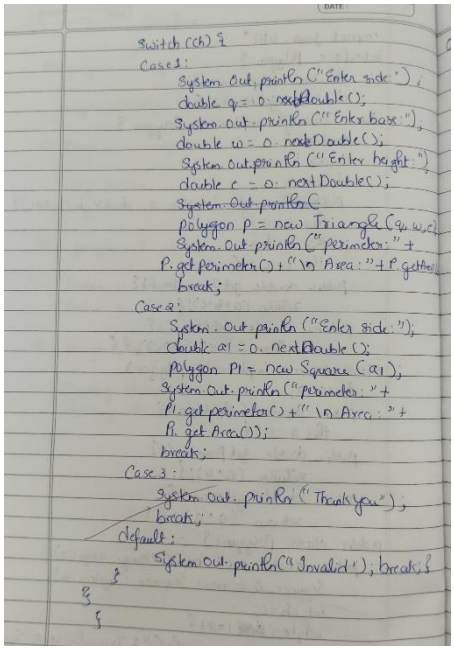
OUTPUT

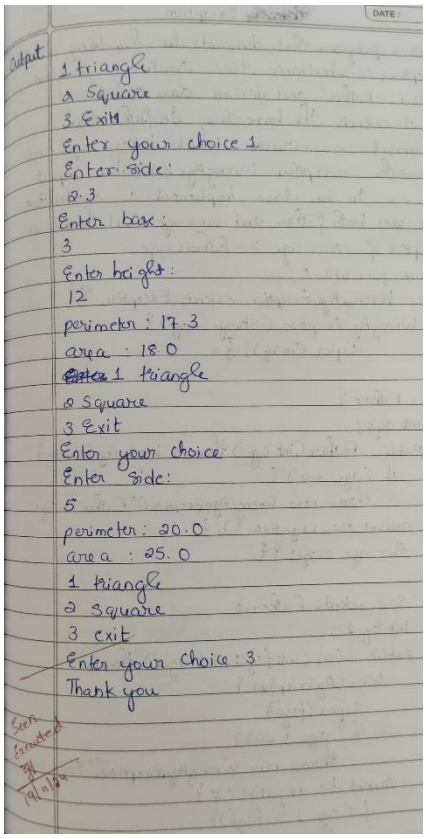
```
C:\Users\chait\OneDrive\Documents\sem 3\java\finalmarks>javac CIE/Student.java CIE/Internals.java SEE/External.java Main.java

C:\Users\chait\OneDrive\Documents\sem 3\java\finalmarks>java Main
Enter the number of students:
2

Enter details of Student:1
 Enter usn:
1BM23CS073


Enter name:
Chaithanya Sudhan

Enter sem:
3

enter the internal marks for 5 courses:
47
48
49
46
48

enter the external marks for 5 courses:
45
41
47
48
50

Final marksof 1 student out of 100:
92
89
```

```
96
94
98

Enter details of Student:2
 Enter usn:
1BM23CS001

Enter name:
Abhijit

Enter sem:
3

enter the internal marks for 5 courses:
40
40
46
45
44

enter the external marks for 5 courses:
45
40
46
47
47

Final marksof 2 student out of 100:
85
80
92
92
91
```

Program 7
Interface







CODE

```
import java.util.Scanner;

interface Polygon {
   double getPerimeter();
   double getArea();
}

class Triangle implements
Polygon {
   double a, b, c;

   public Triangle(double a,
double b, double c) {
      this.a = a;
      this.b = b;
      this.c = c;
   }

   public double getPerimeter() {
      return (a + b + c);
   }

   public double getArea() {
      return (0.5 * (b * c));
   }
}
```

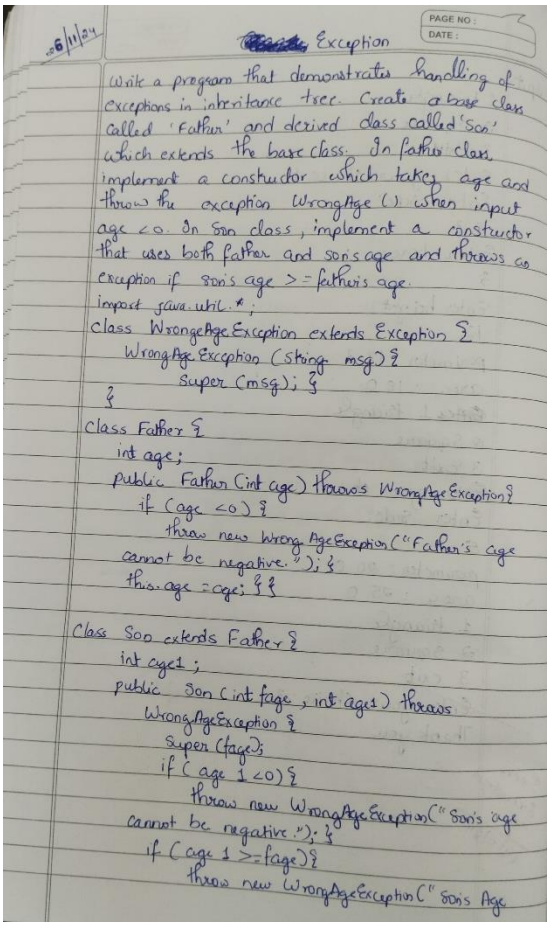
```
class Square implements Polygon
{
   double a;

   public Square(double a) {
      this.a = a;
   }

   public double getPerimeter() {
      return (4 * a);
   }

   public double getArea() {
      return (a * a);
   }
}

public class Polygons {
   public static void main(String[]
args) {
      Scanner D = new
Scanner(System.in);
      int ch = 0;

      while (ch != 3) {
         System.out.println("1
Triangle\n2 Square\n3 Exit\nEnter
your choice: ");
         ch = D.nextInt();

         switch (ch) {
            case 1:

System.out.println("Enter side
1:");
                  double q =
D.nextDouble();

System.out.println("Enter side
2:");
                  double w =
D.nextDouble();

System.out.println("Enter side
3:");
                  double e =
D.nextDouble();

                  Polygon p = new
Triangle(q, w, e);

System.out.println("Perimeter: " +
```

```
p.getPerimeter() + "\nArea: " +
p.getArea());
                break;

            case 2:

System.out.println("Enter side:");
                double a1 =
D.nextDouble();

                Polygon p1 = new
Square(a1);

System.out.println("Perimeter: " +
p1.getPerimeter() + "\nArea: " +
p1.getArea());
                break;

            case 3:

System.out.println("Thank you");
                break;

            default:

System.out.println("Invalid");
                break;
        }
    }
  }
}
```

OUTPUT

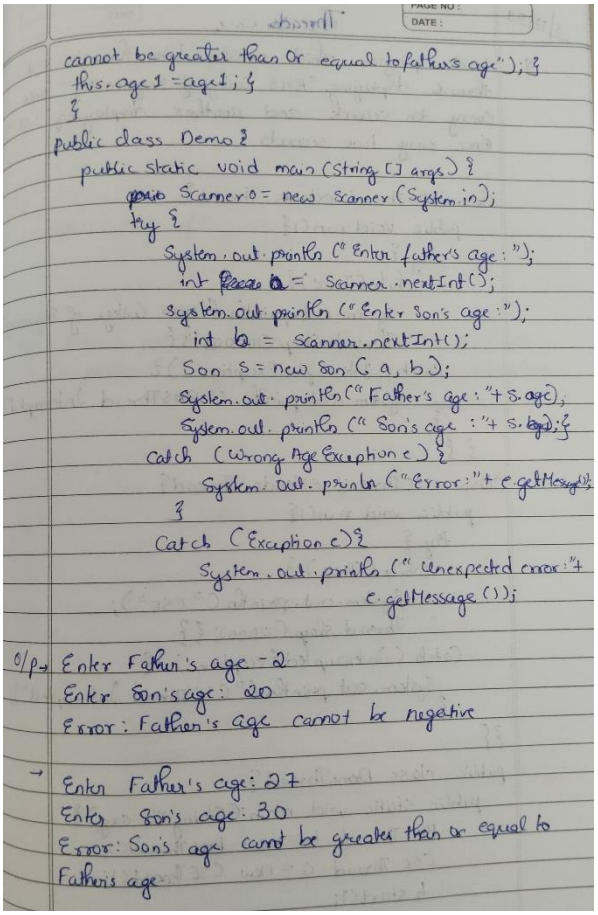
```
C:\Users\chait\OneDrive\Documents\sem 3\java\practiced prgms>javac Polygons.java

C:\Users\chait\OneDrive\Documents\sem 3\java\practiced prgms>java Polygons
1 Triangle
2 Square
3 Exit
Enter your choice:
1
Enter side 1:
2.3
Enter side 2:
3
Enter side 3:
12
Perimeter: 17.3
Area: 18.0
1 Triangle
2 Square
3 Exit
Enter your choice:
2
Enter side:
5
Perimeter: 20.0
Area: 25.0
1 Triangle
2 Square
3 Exit
Enter your choice:
3
Thank you
```

Program 8
Exception Handling


Exception

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

```
import java.util.*;
class WrongeAgeException extends Exception {
    WrongAgeException (String msg) {
        super (msg); }
}
class Father {
    int age;
    public Father (int age) throws WrongAgeException {
        if (age <0) {
            throw new WrongAgeException ("Father's age
    cannot be negative."); }
        this.age =age; } }

class Son extends Father {
    int age1 ;
    public Son (int fage , int age1) throws
        WrongAgeException {
        super (fage);
        if ( age 1 <0) {
            throw new WrongAgeException (" Son's age
    cannot be negative."); }
        if (age 1 >=fage) {
            throw new WrongAgeException (" Sons Age
```



```
cannot be greater than or equal to father's age"); }
    this.age 1 =age1; }
}
public class Demo {
    public static void main (String [] args) {
        Scanner o = new Scanner (System.in);
        try {
            System.out.println (" Enter father's age: ");
            int a = scanner.nextInt();
            System.out.println (" Enter Son's age :");
            int b = scanner.nextInt();
            Son s = new Son (a, b);
            System.out.println (" Father's age : "+ s.age);
            System.out.println (" Son's age : "+ s.b);}
        catch (WrongAgeException e) {
            System.out.println (" Error :"+ e.getMessage());
        }
        Catch (Exception e) {
            System.out.println (" Unexpected error :"+
                e.getMessage ());
```

O/p→ Enter Father's age - 2
Enter Son's age: 20
Error: Father's age cannot be negative

→ Enter Father's age: 27
Enter Son's age: 30
Error: Son's age cannot be greater than or equal to Father's age

CODE

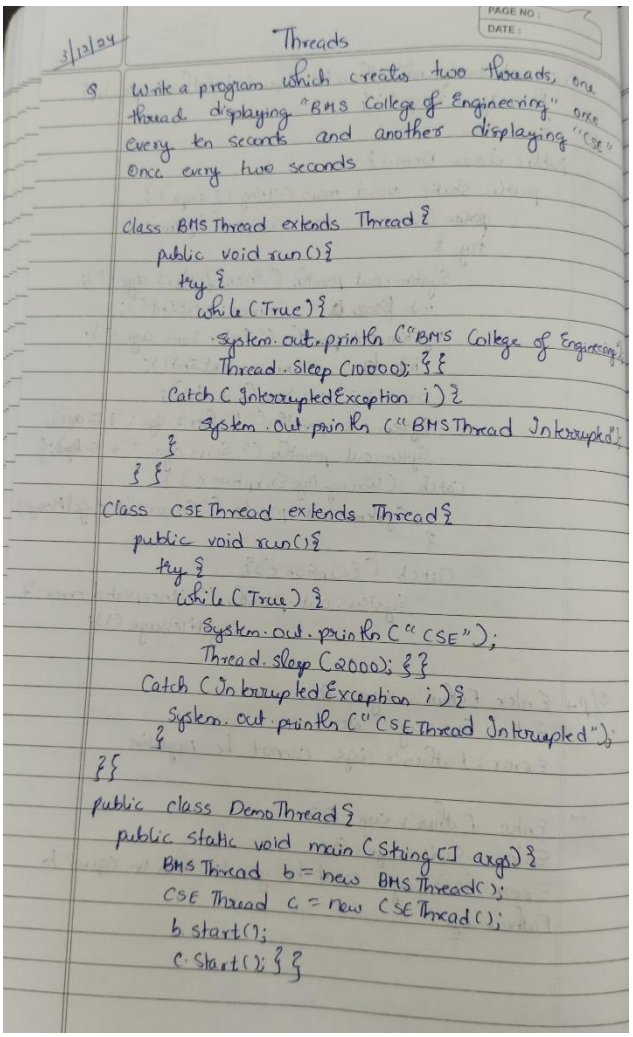
```
import java.util.*;
class MyException extends
Exception{
  MyException(){
    super();
  }
  MyException(String msg){
    super(msg);
  }

}
class Father{
  int age;
  public Father(int age) throws
MyException{
     if(age<0){
      throw new
MyException("fathers age cannot
be negative");}
    this.age=age;
  } }
```

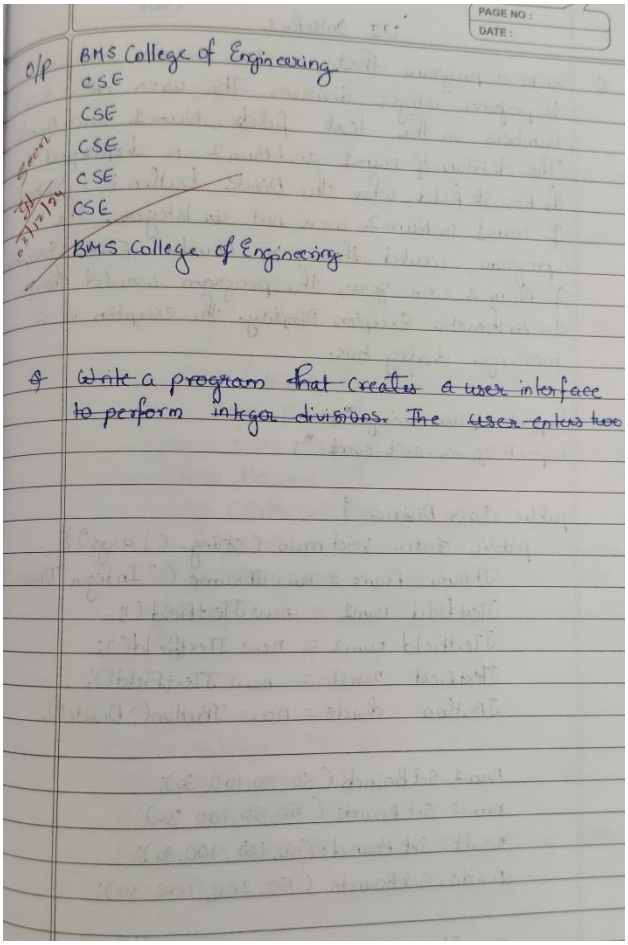
```
class Son extends Father{
   int sage;
   public Son(int fage, int sage)
throws MyException{
    super(fage);
    if (age < 0) {
        throw new
MyException("son's age cannot be
negative");
    }
    if (fage<=sage) {
        throw new
MyException("son's age cannot be
greater than or equal to father's");
    }
    this.sage = sage;
}
}
public class ExceptionDemo{
  public static void main(String[]
args){
   Scanner s=new
Scanner(System.in);
   try{
    System.out.println("enter
father's age:");
    int f1=s.nextInt();
    System.out.println("enter son's
age:");
    int s1=s.nextInt();
    Son son= new Son(f1,s1);
    System.out.println("fathers
age:"+son.age+"sons
age:"+son.sage);}
  catch(MyException e){

System.out.println(e.getMessage()
);}
 }}
```

OUTPUT

```
C:\Users\chait\OneDrive\Documents\sem 3\java>javac ExceptionDemo.java

C:\Users\chait\OneDrive\Documents\sem 3\java>java ExceptionDemo
enter father's age:
-2
enter son's age:
45
fathers age cannot be negative

C:\Users\chait\OneDrive\Documents\sem 3\java>java ExceptionDemo
enter father's age:
24
enter son's age:
45
son's age cannot be greater than or equal to father's

C:\Users\chait\OneDrive\Documents\sem 3\java>java ExceptionDemo
enter father's age:
42
enter son's age:
24
fathers age:42sons age:24

C:\Users\chait\OneDrive\Documents\sem 3\java>
```

Program 9:
Threads





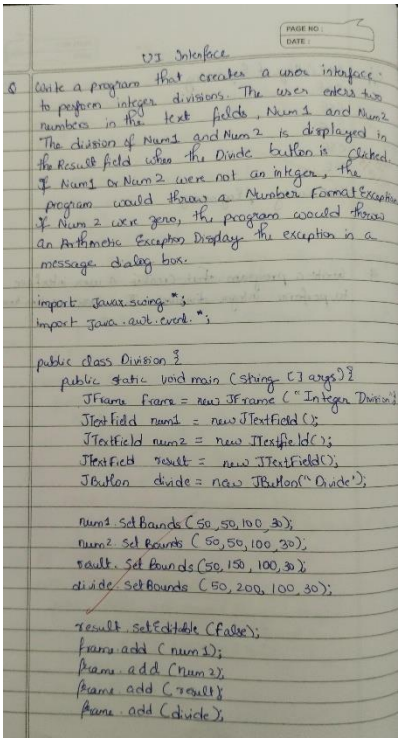
CODE
class BMSThread extends
Thread{
public void run(){

```
    try{
      while(true){
       System.out.println("BMS
College Of Engineering");
       Thread.sleep(10000);
  }}
    catch(InterruptedException i){

System.out.println("BMSThread
Interrupted");
  }
 }
}
class CSEThread extends Thread{
  public void run(){
    try{
      while(true){
       System.out.println("CSE");
       Thread.sleep(2000);
  }}
    catch(InterruptedException i){

System.out.println("CSEThread
Interrupted");
  }
 }
}
public class DemoThread{
  public static void main(String[]
args){
    BMSThread b=new
BMSThread();
    CSEThread c=new
CSEThread();
    b.start();
    c.start();
  }
}
```

OUTPUT

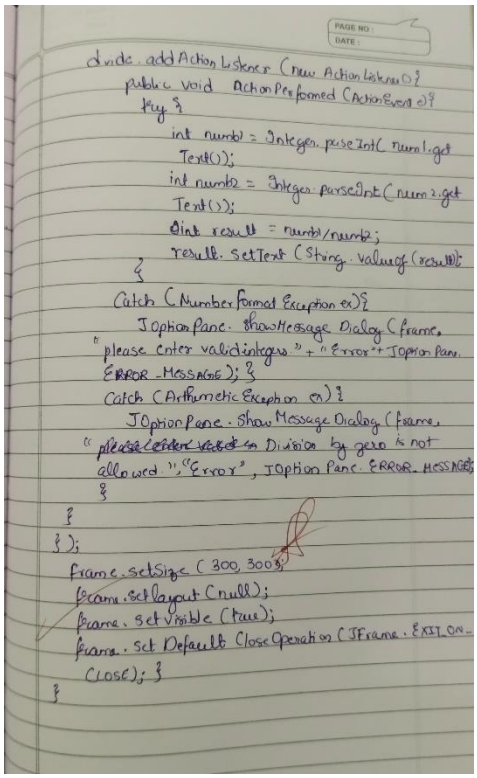
```
C:\Users\Admin\Desktop\073>java DemoThread
BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College Of Engineering
CSE
^C
C:\Users\Admin\Desktop\073>java DemoThread
BMS College Of Engineering
CSE
CSE
CSE
CSE
BMS College Of Engineering
CSE
CSE
^C
C:\Users\Admin\Desktop\073>
```

Program 10
UI Interface


UI Interface

Q 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 ArithmeticException. Display the exception in a message dialog box.

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

public class Division {
   public static void main (String [] args){
      JFrame frame = new JFrame ("Integer Division");
      JTextField num1 = new JTextField ();
      JTextField num2 = new JTextField ();
      JTextField result = new JTextField ();
      JButton divide = new JButton ("Divide");

      num1.setBounds (50, 50, 100, 30);
      num2.setBounds (50, 50, 100, 30);
      result.setBounds (50, 150, 100, 30);
      divide.setBounds (50, 200, 100, 30);

      result.setEditable (false);
      frame.add (num1);
      frame.add (num2);
      frame.add (result);
      frame.add (divide);
```



```
divide.addActionListener (new ActionListener() {
   public void actionPerformed (ActionEvent e){
      try {
         int num1 = Integer.parseInt (num1.get
         Text());
         int num2 = Integer.parseInt (num2.get
         Text());
         int result = num1/num2;
         result.setText (String.valueOf (result));
      }
      Catch (NumberFormatException ex){
         JOptionPane.showMessageDialog (frame,
"please enter valid integers" + "Error" + JOptionPane.
         ERROR_MESSAGE); }
      Catch (ArithmeticException ex) {
         JOptionPane.showMessageDialog (frame,
"Division by zero is not
         allowed.", "Error", JOptionPane.ERROR_MESSAGE);
      }
   }
});
frame.setSize (300, 300);
frame.setLayout (null);
frame.setVisible (true);
frame.setDefaultCloseOperation (JFrame.EXIT_ON_
   CLOSE); }
}
```

CODE

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

public class DivisionApp {
   public static void main(String[]
args) {
      JFrame frame = new
JFrame("Integer Division");
      JTextField num1Field = new
JTextField();
      JTextField num2Field = new
JTextField();
      JTextField resultField = new
JTextField();
      JButton divideButton = new
JButton("Divide");

      num1Field.setBounds(50, 50,
100, 30);
      num2Field.setBounds(50,
100, 100, 30);
```

```
        resultField.setBounds(50,
150, 100, 30);
        divideButton.setBounds(50,
200, 100, 30);

        resultField.setEditable(false);
        frame.add(num1Field);
        frame.add(num2Field);
        frame.add(resultField);
        frame.add(divideButton);


divideButton.addActionListener(n
ew ActionListener() {
            public void
actionPerformed(ActionEvent e) {
                try {
                    int num1 =
Integer.parseInt(num1Field.getTex
t());
                    int num2 =
Integer.parseInt(num2Field.getTex
t());
                    int result = num1 /
num2;

resultField.setText(String.valueOf
(result));
                } catch
(NumberFormatException ex) {

JOptionPane.showMessageDialog
(frame, "Please enter valid
integers.", "Error",
JOptionPane.ERROR_MESSAGE
);
                } catch
(ArithmeticException ex) {

JOptionPane.showMessageDialog
(frame, "Division by zero is not
allowed.", "Error",
JOptionPane.ERROR_MESSAGE
);
                }
            }
        });

        frame.setSize(300, 300);
        frame.setLayout(null);
        frame.setVisible(true);
```

```
frame.setDefaultCloseOperation(J
Frame.EXIT_ON_CLOSE);
    }
}
```

OUTPUT