RECORD

LAB 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.

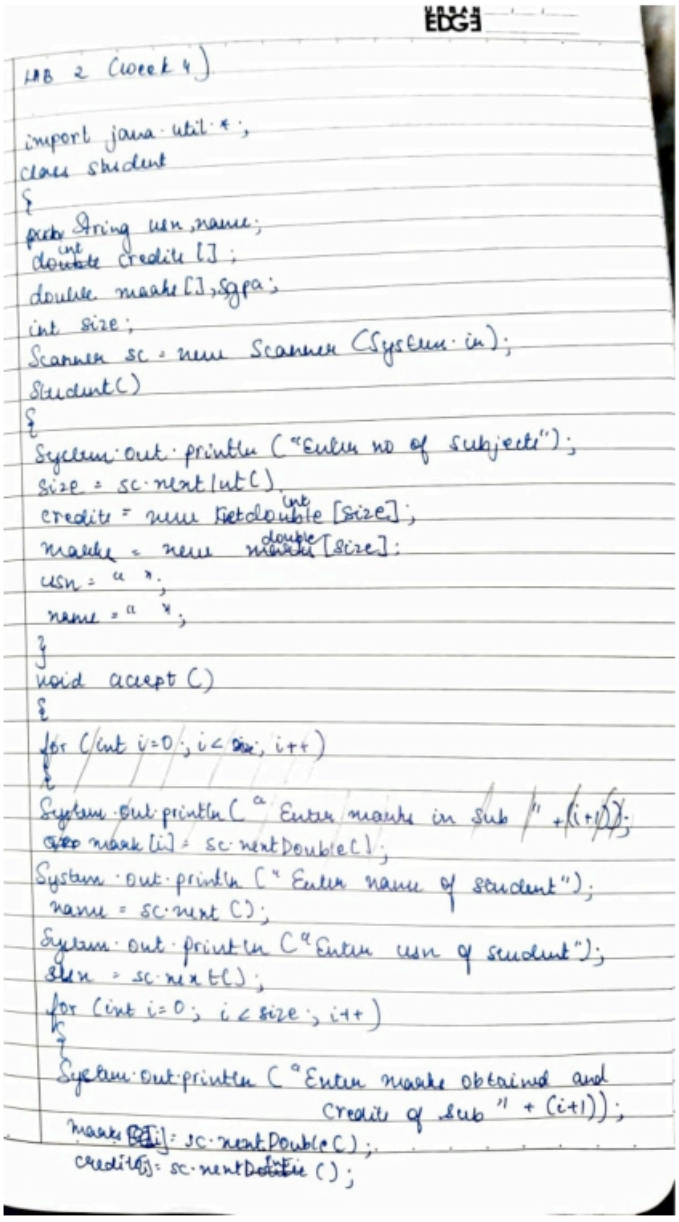
```
import java.util.*;
class quad
{
public static void main (String args[])
{
double a1, b1, c1, ans1, ans2;
Scanner sc= new Scanner (System.in)
System.out.println ("Enter values of a, b, c for quad
                    eqn in the form of ax²+bx+c.
                    where 'a' should be non 0");
```

```
a1 = sc.nextDouble();
b1 = sc.nextDouble();
c1 = sc.nextDouble();
if (a1==0)
System.out.println ("'a' should be non 0");
else
{
a2=(b*b)-(4*a*c);
if (a2>0)
{
System.out.println ("roots are real & unequal");
ans1 = (-b + Math.sqrt(a2)) / (2*a1);
ans2 = (-b - Math.sqrt(a2)) / (2*a1);
System.out.println ("the solutions of quad eqn are"
                    + ans1 + "and" + ans2);
}
else if (a2==0)
{
System.out.println ("roots are real & unequal");
ans1 = (-b + Math.sqrt(a2)) / (2*a1);
ans2 = (-b - Math.sqrt(a2)) / (2*a1);
System.out.println ("Solns of quad eqn are" + ans1
                    + "and" + ans2);
}
else
System.out.println ("no real roots");
}
}
}
```

```
[Arvinds-MacBook-Pro:ooj Arvind$ java quad
Enter the values of a,b,c for quad eqn in the form of ax^2+bx+c
where 'a' should be non zero
1
-3
-10
roots are real and unequal
The solutions of quad eqns are 5,0000 and -2,0000
[Arvinds-MacBook-Pro:ooj Arvind$ java quad
Enter the values of a,b,c for quad eqn in the form of ax^2+bx+c
where 'a' should be non zero
5
4
1
There are no real roots
[Arvinds-MacBook-Pro:ooj Arvind$ java quad
Enter the values of a,b,c for quad eqn in the form of ax^2+bx+c
where 'a' should be non zero
4
-4
1
roots are real and equal
The solutions of quad eqns are 0,5000 and 0,5000
Arvinds-MacBook-Pro:ooj Arvind$
```

LAB 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.



```
LAB 2 (week 4)

import java.util.*;
class student
{
String usn,name;
int credits[];
double marks[],sgpa;
int size;
Scanner sc = new Scanner(System.in);
Student()
{
System.out.println("Enter no of subjects");
size = sc.nextInt();
credits = new int[size];
marks = new double[size];
usn = " ";
name = " ";
}
void accept()
{
for(int i=0; i<size; i++)
{
System.out.println("Enter marks in sub "+(i+1));
marks[i] = sc.nextDouble();
System.out.println("Enter name of student");
name = sc.next();
System.out.println("Enter usn of student");
usn = sc.next();
for(int i=0; i<size; i++)
{
System.out.println("Enter marks obtained and
                    Credits of sub " + (i+1));
marks[i] = sc.nextDouble();
credits[i] = sc.nextInt();
```

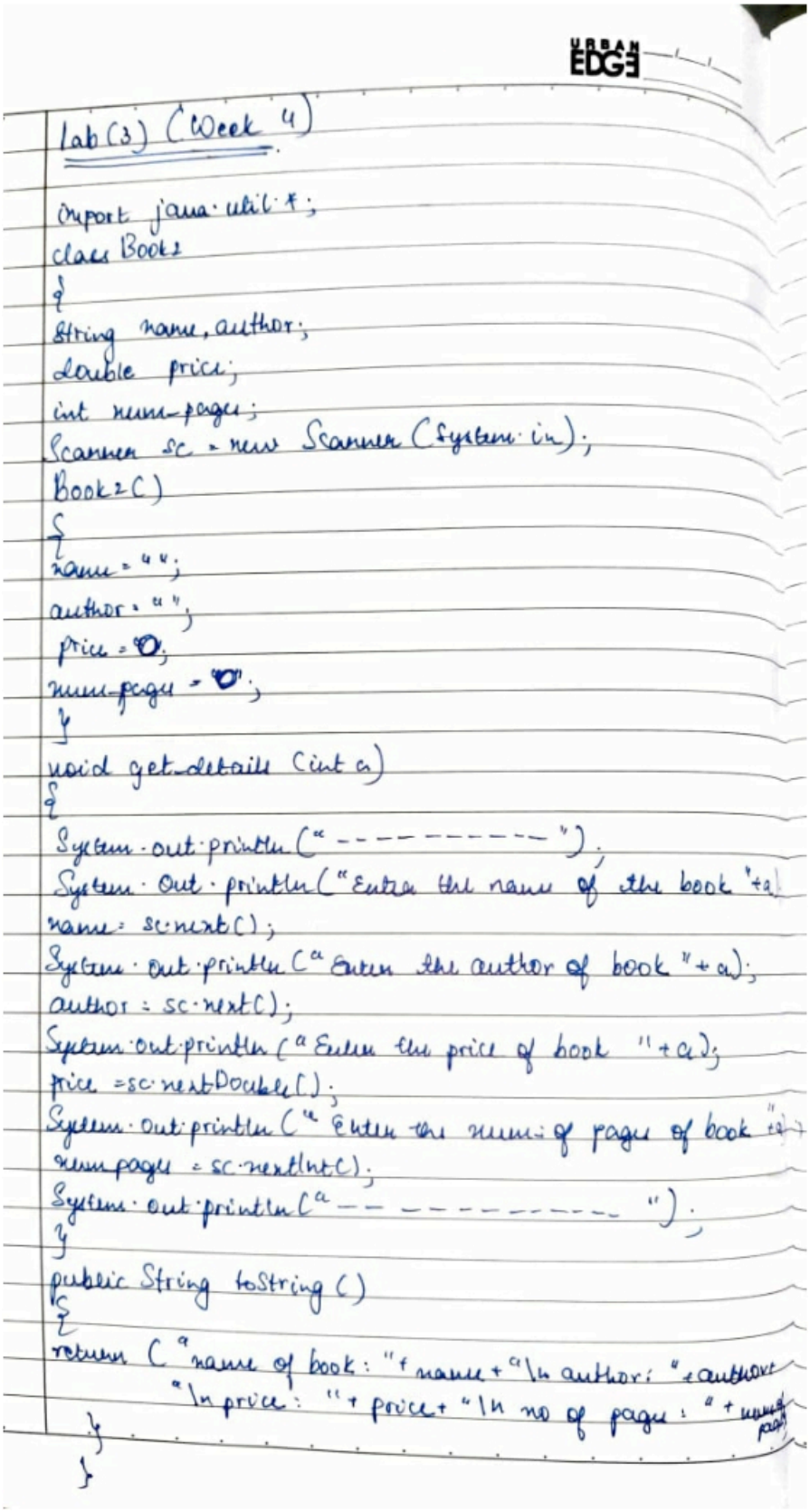
```
}
}
void display()
{
System.out.println("Name: " + name);
System.out.println("usn: " + usn);
System.out.println("Marks obtained");
for(int i=0; i<size; i++)
System.out.println("Marks in sub " + (i+1) + ": " + marks[i]);
System.out.println("sgpa: " + sgpa);
}
void calc()
{
for(int i=0; i<size; i++)
{
double sum=0, total=0;
for(int i=0; i<size; i++)
{
sum = sum + credits[i];
total = total + (marks[i] * credits[i]);
if(marks>=90 && marks<100)
total = total + (marks[i] * credits[i]);
else if(marks>=80 && marks<90)
total = total + (9 * credits[i]);
else if(marks>=70 && marks<80)
total = total + (8 * credits[i]);
else if(marks>=60 && marks<70)
total = total + (7 * credits[i]);
else if(marks>=50 && marks<60)
total = total + (6 * credits[i]);
else if(marks>=40 && marks<50)
total = total + (5 * credits[i]);
else
total = total + (4 * credits[i]);
}
```

```
sgpa = total / sum;
}
}

class student1
{
public static void main (String args[])
{
Student s = new student();
s.accept();
s.calc();
s.display();
}
}
```

```
[Arvinds-MacBook-Pro:ooj Arvind$ javac Student.java
[Arvinds-MacBook-Pro:ooj Arvind$ java student1
Enter number of subjects
4
Enter the name of the student
aaa
enter usn of students
1bm19cs021
enter marks obtained and credits sub1
90
4
enter marks obtained and credits sub2
95
4
enter marks obtained and credits sub3
97
4
enter marks obtained and credits sub4
99
6
name:aaa
usn:1bm19cs021
------marks obtained-----
sub 1: marks=90.0 ; credits=4; grade_points=40.0;
sub 2: marks=95.0 ; credits=4; grade_points=40.0;
sub 3: marks=97.0 ; credits=4; grade_points=40.0;
sub 4: marks=99.0 ; credits=6; grade_points=60.0;
sgpa:10,00
Arvinds-MacBook-Pro:ooj Arvind$
```

LAB 3

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



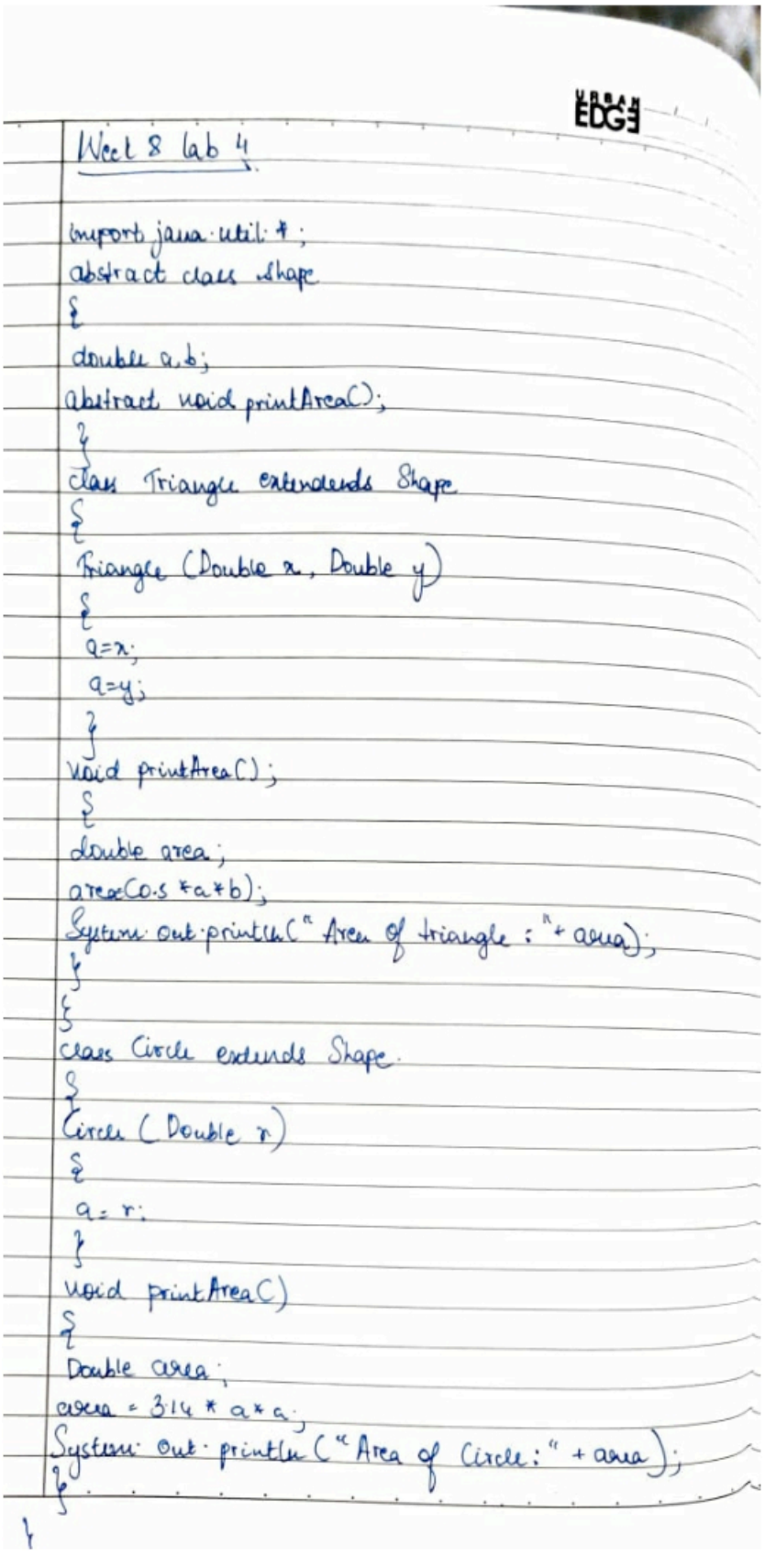
```
lab (3) (Week 4)

import java.util.*;
class Book2
{
String name, author;
double price;
int num_pages;
Scanner sc = new Scanner (System.in);
Book2()
{
name = " ";
author = " ";
price = 0;
num_page = 0;
}
void get_details (int a)
{
System.out.println(" ---------- ");
System.out.println("Enter the name of the book "+a);
name = sc.next();
System.out.println("Enter the author of book "+a);
author = sc.next();
System.out.println("Enter the price of book "+a);
price = sc.nextDouble();
System.out.println("Enter the num of page of book "+a);
num_page = sc.nextInt();
System.out.println(" ---------- ");
}
public String toString()
{
return ("name of book: "+name+"\n author: "+author
"\n price: "+price+"\n no of page: "+num_page);
}
}
```

```
Class Book2main
{
public static void main (String args[])
{
int n;
Book2 o[];
Scanner sc = new Scanner (System.in);
System.out.println ("Enter the no of book details to
                     be entered");
n = sc.nextInt();
o = new Book2[n];
for(int i=0 ; i<n ; i++)
{
o[i] = new Book2();
o[i].get_details (i+1);
}
for (int i=0; i<n; i++)
{
System.out.println ("details of book "+ (i+1));
System.out.println (" ---  ---------- ");
System.out.println (o[i]);
}
}
}
```

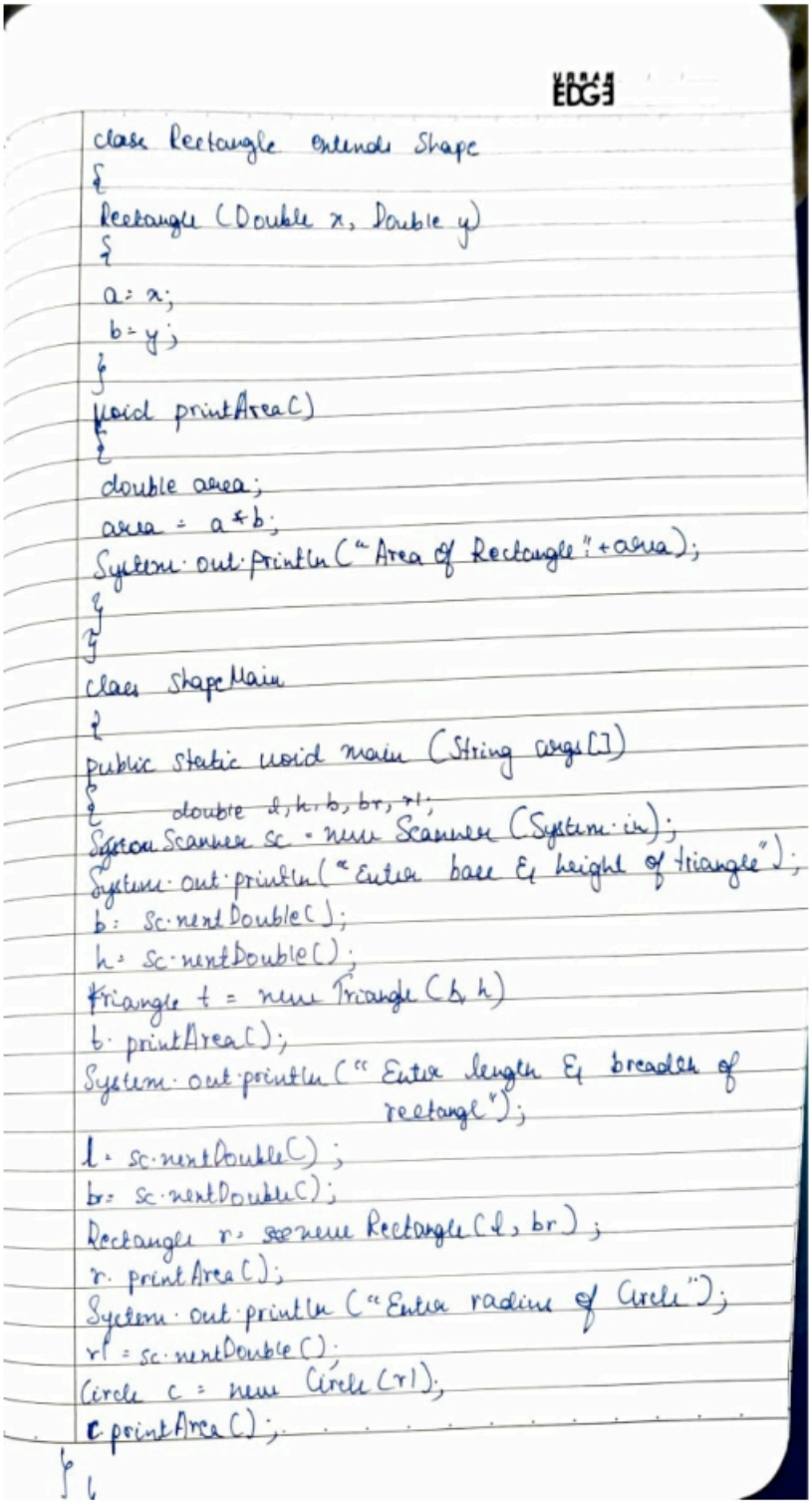
```
Arvinds-MacBook-Pro:ooj Arvind$ java
enter the no of book details to be en
3
--------------------------------
enter the name of the book 1
revolution2020
enter the author of book 1
chethan
enter the price of book 1
300
enter the num_pages of book 1
300
--------------------------------
--------------------------------
enter the name of the book 2
kiterunner
enter the author of book 2
khaled
enter the price of book 2
400
enter the num_pages of book 2
300
--------------------------------
--------------------------------
enter the name of the book 3
divergent
enter the author of book 3
veronica
enter the price of book 3
150
enter the num_pages of book 3
100
--------------------------------
details of book 1
--------------------------------
name of book: revolution2020
author: chethan
price: 300.0
no of pages: 300
--------------------------------
details of book 2
--------------------------------
name of book: kiterunner
author: khaled
price: 400.0
no of pages: 300
--------------------------------
details of book 3
--------------------------------
name of book: divergent
author: veronica
price: 150.0
no of pages: 100
--------------------------------
Arvinds-MacBook-Pro:ooj Arvind$
```

LAB 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.



```
Week 8 lab 4

import java.util.*;
abstract class shape
{
double a,b;
abstract void printArea();
}
class Triangle extendsends Shape
{
Triangle (Double x, Double y)
{
a=x;
a=y;
}
void printArea();
{
double area;
area=0.5 *a*b);
System.out.println("Area of triangle :"+ area);
}
}
class Circle extends Shape.
{
Circle (Double r)
{
a= r;
}
void printArea()
{
Double area;
area = 3.14 * a*a;
System.out.println("Area of Circle:" + area);
}
}
```

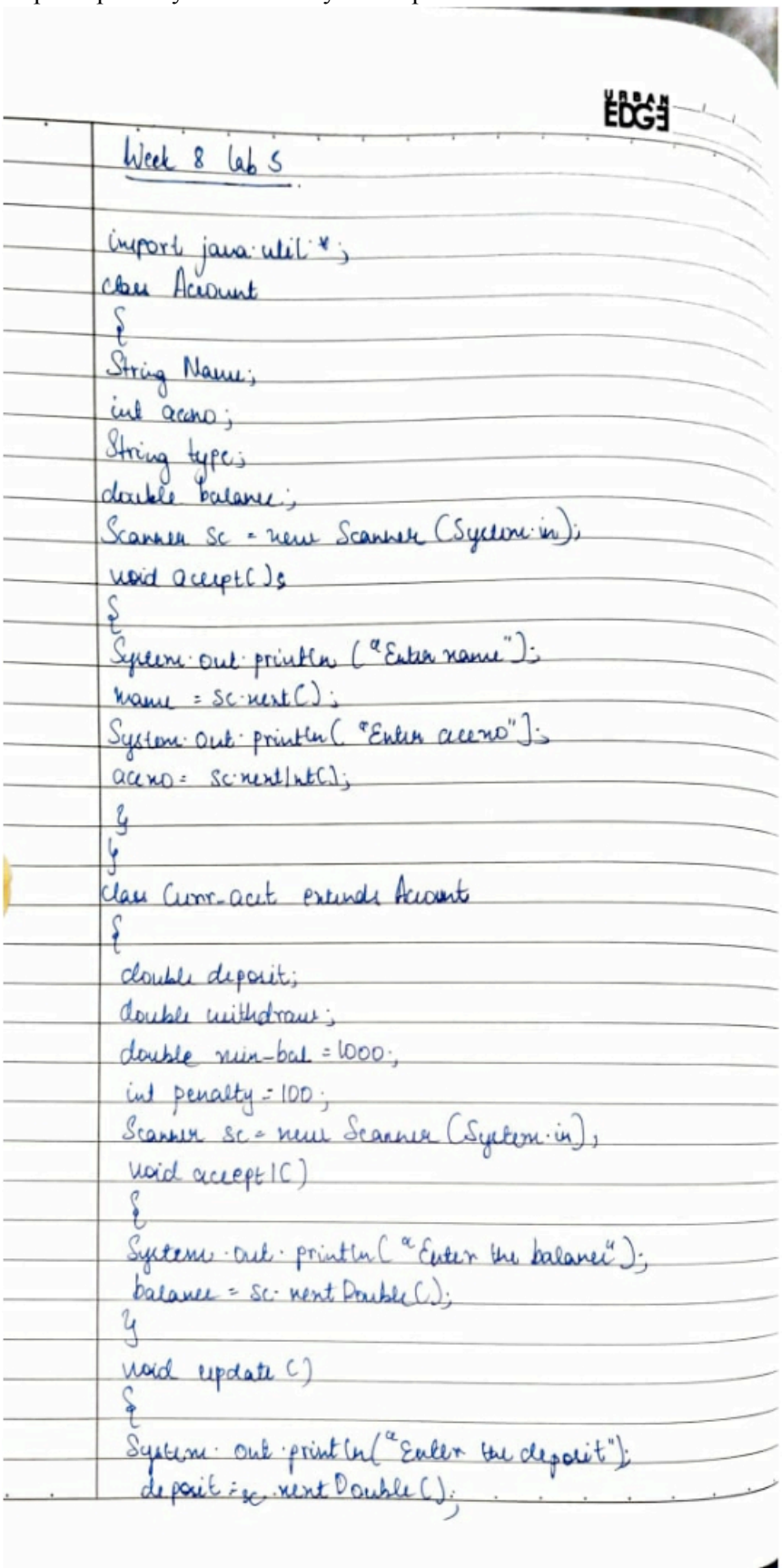


```
class Rectangle extends Shape
{
Rectangle (Double x, Double y)
{
a= x;
b= y;
}
void printArea()
{
double area;
area = a*b;
System.out.println("Area of Rectangle"+area);
}
}
class ShapeMain
{
public static void main (String args[])
{
    double t,h,b,br,rl;
Scanner sc = new Scanner (System.in);
System.out.println("Enter base & height of triangle");
b= sc.nextDouble();
h= sc.nextDouble();
Triangle t = new Triangle (b,h)
b.printArea();
System.out.println(" Enter length & breadth of
                    rectangle");
l= sc.nextDouble();
br= sc.nextDouble();
Rectangle r= new Rectangle (l, br);
r.printArea();
System.out.println ("Enter radius of Circle");
rl = sc.nextDouble();
Circle c = new Circle (rl);
c.printArea();
}
}
```

```
[Arvinds-MacBook-Pro:ooj Arvind$ java ShapeMain
enter the base and height of triangle
2
5
area of triangle:5.0
enter the lenght and breadth of rectangle
3
6
area of Rectangle:18.0
enter the radius of circle
4
area of circle:50.24
Arvinds-MacBook-Pro:ooj Arvind$
```

LAB 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Curr-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: · Accept deposit from customer and update the balance. · Display the balance. · Compute and deposit interest · Permit withdrawal and update the balance · Check for the minimum balance, impose penalty if necessary and update the balance



Week 8 lab 5

```
import java.util.*;
class Account
{
String Name;
int accno;
String type;
double balance;
Scanner sc = new Scanner (System.in);
void accept()
{
System.out.println ("Enter name");
name = sc.next();
System.out.println( "Enter accno");
accno = sc.nextInt();
}
}
class Curr_acct extends Account
{
double deposit;
double withdraw;
double min_bal = 1000;
int penalty = 100;
Scanner sc = new Scanner (System.in);
void accept1()
{
System.out.println( "Enter the balance");
balance = sc.nextDouble();
}
void update()
{
System.out.println("Enter the deposit");
deposit = sc.nextDouble();
```

```
balance += deposit;
System.out.println("balance got update to " + balance);
}
void check()
{
if (balance < min_bal)
{
System.out.println("a penalty of 100 is imposed");
balance -= penalty;
System.out.println("balance got update to: " + balance);
}
}
void withdrawal()
{
System.out.println("Enter amt to be withdrawn");
withdraw = sc.nextDouble();
balance -= withdraw;
System.out.println("balance got updated to:" + balance);
check();
}
void display()
{
System.out.println("------ DETAILS -----");
System.out.println("name: " + name);
System.out.println("accno: " + accno);
System.out.println("type: " + type);
if (balance>=0)
System.out.println("balance: " + balance);
System.out.println("cheque book facility exists");
}
}
```

```
class savacct extends Account
{
double balance;
double deposit;
double withdraw;
double interest, time, rate;
Scanner sc = new Scanner (System.in);
void accept()
{
System.out.println ("Enter the balance");
balance = sc.nextDouble();
}
void update()
{
System.out.println ("Enter the deposit");
deposit = sc.nextDouble();
balance += deposit;
System.out.println ("balance got update to: " + balance);
}
void calcinterest()
{
System.out.println("Enter amt to be withdrawn");
withdraw = sc.nextDouble();
balance -= withdraw;
System.out.println ("balance got update to: " + balance);
}
void display()
{
System.out.println ("------ DETAILS ------");
System.out.println ("name: " + name);
System.out.println ("acno: " + accno);
System.out.println ("type: " + type);
calcinterest();
```

```
if (balance < 0)
{
System.out.printf ("balance : %.3f", balance);
System.out.println ();
System.out.println ("Cheque book facility does not exist");
}
}


class Bank
{
public static void main (String args[])
{
int op, ch;
Scanner sc = new Scanner (System.in);
System.out.println ("1. Curr Acct \n2. Sav.acct");
System.out.println ("Enter your choice");
op = sc.nextInt();
Switch (ch)
{
case 1:
a.update ();
break;
case 2:
if (op == 1)
{
Curr_acct a = new Curr.acct();
a.type = "Curr-acct";
a.accept();
a.accept();
do {
System.out.println ("1. deposit 2. withdrawal 3. exit");
System.out.println ("enter your choice");
ch = sc.nextInt();
Switch (ch)
```

```
{
case 1:
a.update();
break;
case 2:
a.withdrawal();
break;
case 3:
a.display();
break;
default:
System.out.println("Wrong choice");
}
}while(ch!=3);
}
if(op==2)
{
Sav_acct b = new SavAcct();
b.type = "sav_acct";
b.accept();
b.acceptI();
do{
System.out.println("1.deposit 2.withdrawal 3.exit");
System.out.println("Enter your choice");
ch = sc.nextInt();
switch(ch)
{
case 1:
b.update();
break;
case 2:
b.withdrawal();
break;
```

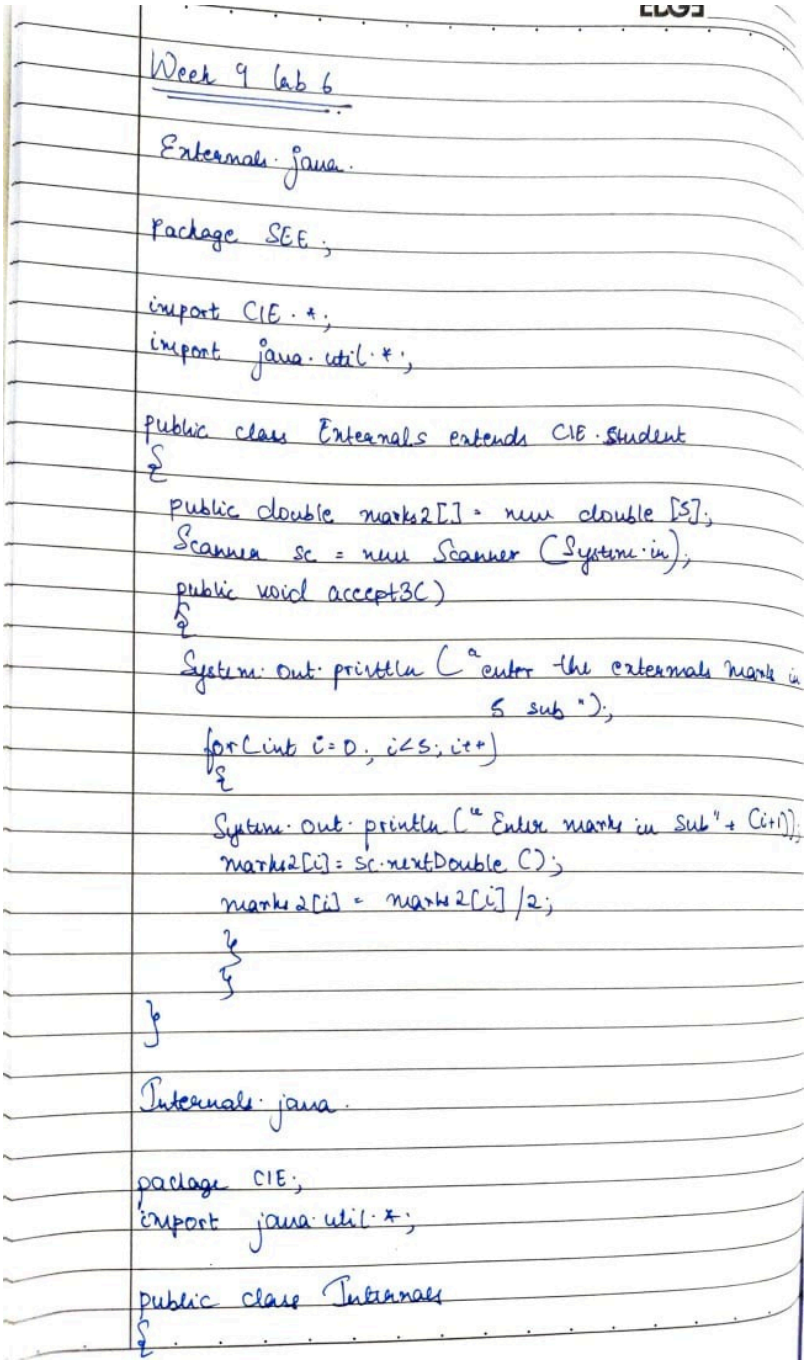
```
case 3:
b.display();
break;
default:
System.out.println("Wrong choice");
}
}while(ch!=3);
}
}
```

```
Arvinds-MacBook-Pro:ooj Arvind$ java Bank
1.curr_Acct
2.sav acct
enter the type
1
enter name
aaa
enter accno
1234
enter the balance
1000
1.deposit 2.withdrawal 3.exit
enter your choice
1
enter the deposit
100
balance got updated to:1100.0
1.deposit 2.withdrawal 3.exit
enter your choice
2
 enter amt to be withdrawn
700
balance got updated to:400.0
a penalty of 100 is imposed
balance got updated to:300.0
1.deposit 2.withdrawal 3.exit
enter your choice
3
----------DETAILS----------
name:aaa
accno:1234
type:curr_Acct
balance:300.0
cheque book facility exist
Arvinds-MacBook-Pro:ooj Arvind$
```

```
Arvinds-MacBook-Pro:ooj Arvind$ java Bank
1.curr_Acct
2.sav acct
enter the type
2
enter name
aaa
enter accno
1234
enter the balance
1000
1.deposit 2.withdrawal 3.exit
enter your choice
1
enter the deposit
300
balance got updated to:1300.0
1.deposit 2.withdrawal 3.exit
enter your choice
2
 enter amt to be withdrawn
100
balance got updated to:1200.0
1.deposit 2.withdrawal 3.exit
enter your choice
3
----------DETAILS----------
name:aaa
accno:1234
type:sav_Acct
a rate of 4.5% is given for deposits in savings bank acc
enter the time duration for which interest to be calc
1
balance:1254,000
cheque book does not facility exist
Arvinds-MacBook-Pro:ooj Arvind$
```

LAB 6

Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.



Week 9 lab 6

Externals.java.

```
package SEE;

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

public class Externals extends CIE.Student
{
  public double marks2[] = new double[5];
  Scanner sc = new Scanner(System.in);
  public void accept3C()
  {
  System.out.println("enter the externals marks in
                                    5 sub ");
    for(int i=0; i<5; i++)
    {
    System.out.println("Enter marks in sub"+(i+1));
    marks2[i] = sc.nextDouble();
    marks2[i] = marks2[i]/2;
    }
  }
}
```

Internals.java.

```
package CIE;
import java.util.*;

public class Internals
{
```

```
public double marks[] = new double[6];
Scanner sc = new Scanner(System.in);
public void accept1()
{
System.out.println("Enter the internals marks in
                    5 sub");
for(int i=0; i<5; i++)
{
System.out.println("Enter marks in sub"+(i+1));
marks[i] = sc.nextDouble();
}
}
}
```

Student.java.

```
package CIE;
import java.util.*;

public class Student
{
  public String usn;
  public String name;
  public int sem;
  public double total[] = new double[5];
  Scanner sc = new Scanner(System.in);
  public void accept2(int a)
  {
    System.out.println("Enter the name of
                        student"+a);
    name = sc.next();
    System.out.println("enter the usn of student"
                        +a);
    usn = sc.next();
```

```
        System.out.println("Enter the sem of Student");
        sem = sc.next();
    }
    public void display(int a)
    {
        System.out.println("---- details of student "+a);
        System.out.println("name: "+name);
        System.out.println("usn: "+usn);
        System.out.println("Sem: "+sem);
        for(int i=0;i<5;i++)
        System.out.println("total marks in sub"
                                +(i+1)+": "+tot[i]);
    }
}
```

Main.java

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

class Main1
{
public static void main(String args[])
{
SEE.Externals s2[];
CIE.Internals s1[];
int n;
System.out.println("Enter the no of Student:");
Scanner sc= new Scanner(System.in);
n=sc.nextInt();
s2= new SEE.Externals[n];
s1= new CIE.Internals[n];
```

```
for(int i=0;i<n;i++)
{
s1[i]= new CIE.Internals();
s2[i]= new SEE.Externals();
s2[i].accept2(i+1);
s1[i].accept1();
s2[i].accept3();
for(int j=0;j<5;j++)
s2[i].total[j]= s1[i].marks1[j]+s2[i].marks2[j];
}
for(int i=0;i<n;i++)
s2[i].display(i+1);
}
}
```

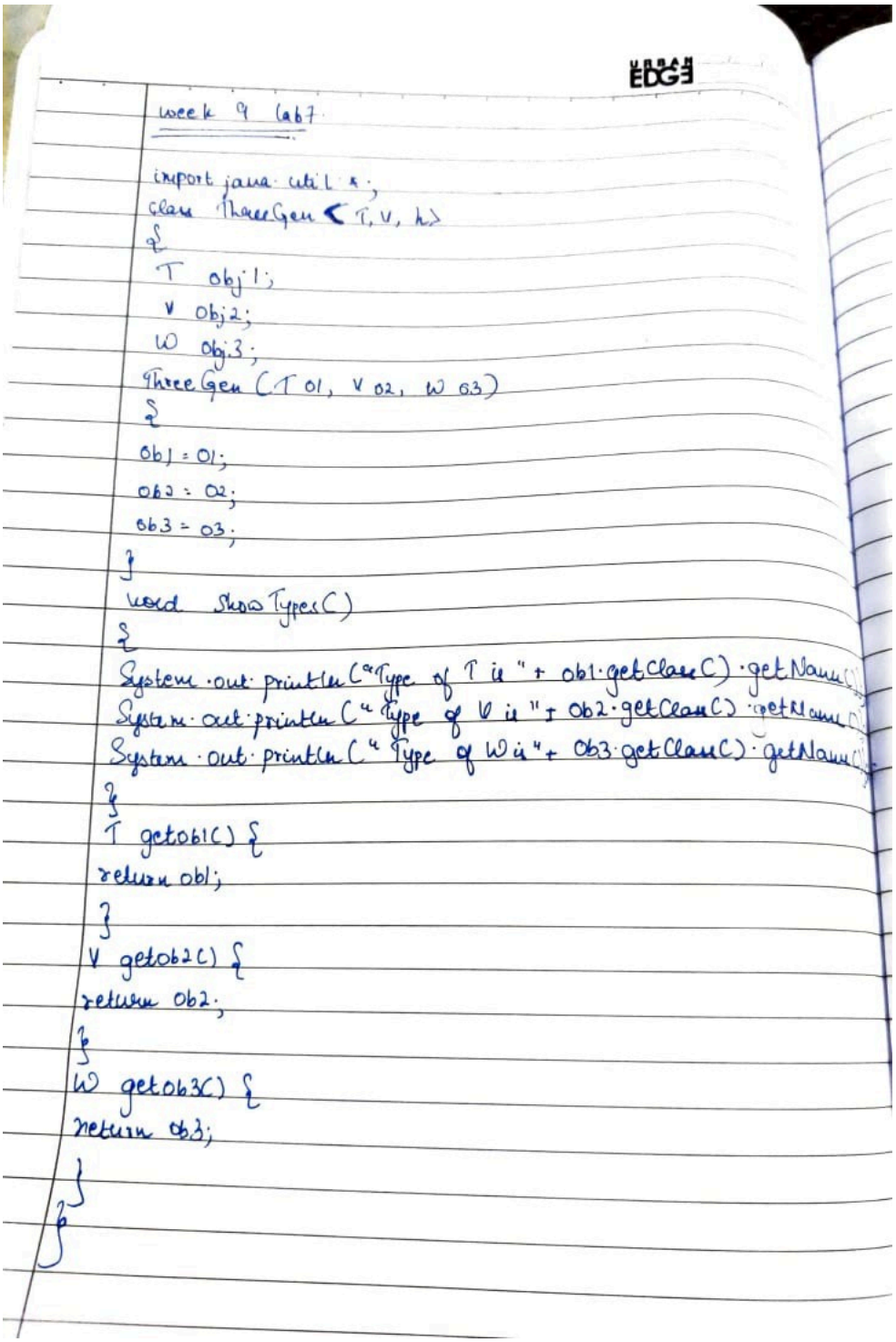
```
1 error
[Arvinds-MacBook-Pro:packages Arvind$ javac CIE/Student.java
[Arvinds-MacBook-Pro:packages Arvind$ javac CIE/Internals.java
[Arvinds-MacBook-Pro:packages Arvind$ javac SEE/Externals.java
[Arvinds-MacBook-Pro:packages Arvind$ javac Main1.java
[Arvinds-MacBook-Pro:packages Arvind$ java Main1
```

```
enter the no of students
2
enter the name of student 1
aaa
enter the usn of student 1
1bm19cs02
enter the sem of student 1
3
enter the internals marks in 5 sub
enter marks in sub 1
45
enter marks in sub 2
50
enter marks in sub 3
45
enter marks in sub 4
50
enter marks in sub 5
45
enter the externals marks in 5 sub
enter marks in sub 1
100
enter marks in sub 2
90
enter marks in sub 3
100
enter marks in sub 4
90
enter marks in sub 5
100
enter the name of student 2
bbb
enter the usn of student 2
1bm19cs03
enter the sem of student 2
3
enter the internals marks in 5 sub
enter marks in sub 1
45
enter marks in sub 2
45
enter marks in sub 3
45
enter marks in sub 4
46
enter marks in sub 5
45
enter the externals marks in 5 sub
enter marks in sub 1
100
enter marks in sub 2
100
enter marks in sub 3
100
enter marks in sub 4
90
```

```
---------details of student 1 ------------
name:aaa
usn:1bm19cs02
sem:3
total marks in sub 1:95.0
total marks in sub 2:95.0
total marks in sub 3:95.0
total marks in sub 4:95.0
total marks in sub 5:95.0
---------details of student 2 ------------
name:bbb
usn:1bm19cs03
sem:3
total marks in sub 1:95.0
total marks in sub 2:95.0
total marks in sub 3:95.0
total marks in sub 4:91.0
total marks in sub 5:90.0
Arvinds-MacBook-Pro:packages Arvind$
```

LAB 7

Write a program to demonstrate generics with multiple object parameters.



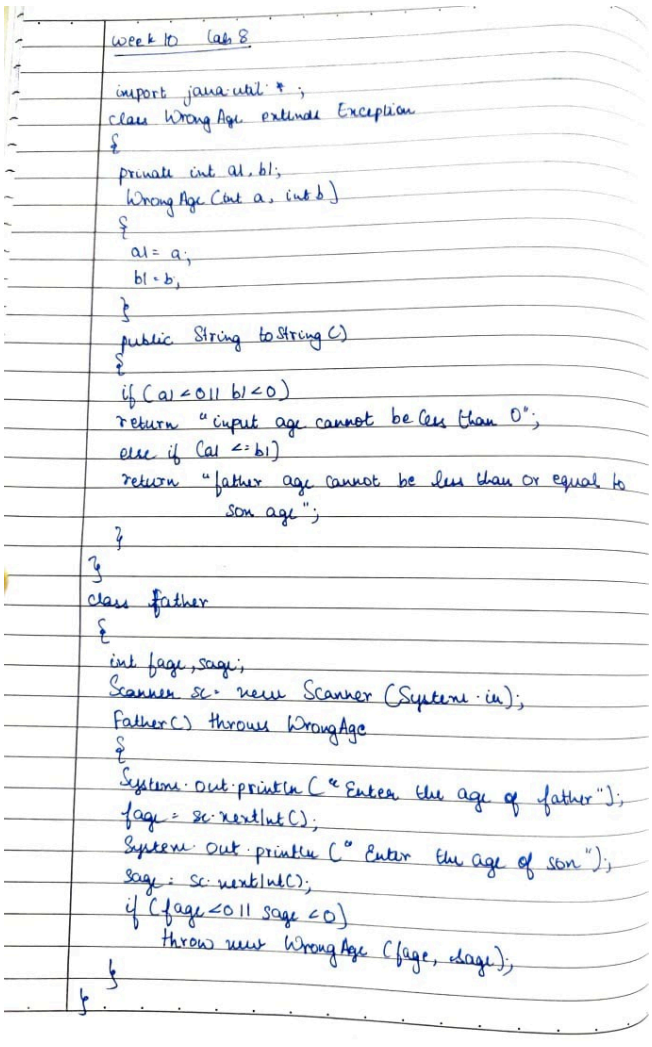
```
week 9 lab7

import java.util.*;
class ThreeGen <T, V, W>
{
 T obj1;
 V obj2;
 W obj3;
ThreeGen (T o1, V o2, W o3)
{
ob1 = o1;
ob2 = o2;
ob3 = o3;
}
void ShowTypes()
{
System.out.println("Type of T is " + ob1.getClass().getName());
System.out.println("Type of V is " + ob2.getClass().getName());
System.out.println("Type of W is " + ob3.getClass().getName());
}
T getob1() {
return ob1;
}
V getob2() {
return ob2;
}
W getob3() {
return ob3;
}
}
```

```
class SimpleGen
{
public static void main (String args[]) {
Scanner sc = new Scanner (System.in);
int a;
String b;
float c, c1;
char a1;
double d1;
System.out.println ("Enter a integer, string, float");
a = sc.nextInt();
b = sc.next();
c = sc.nextFloat();
ThreeGen <Integer, String, Float> tgObj =
new ThreeGen <Integer, String, Float> (a, b, c);
tgObj.showTypes();
int v = tgobj.getob1();
System.out.println ("value1: " + v);
String str = tgObj.getob2();
System.out.println("value1: " + str);
float x = tgObj.getob3();
System.out.println ("value3: " + x);
System.out.println ("Enter a Character, Double, float");
a1 = sc.next().charAt(0);
d1 = sc.nextDouble();
c1 = sc.nextFloat();
ThreeGen <character, Double, Float> tgObl =
new ThreeGen <Character, Double, Float> (a1, d1, c1);
tgObl.showTypes();
char v1 = tgObj1.getob1();
System.out.println ("value1: " + v1);
double str1 = tgObj1.getob2();
System.out.println ("Value2: " + str1);
float x1 = tgObj1.getob3();
System.out.println ("value 3: " + x1);
}}
```

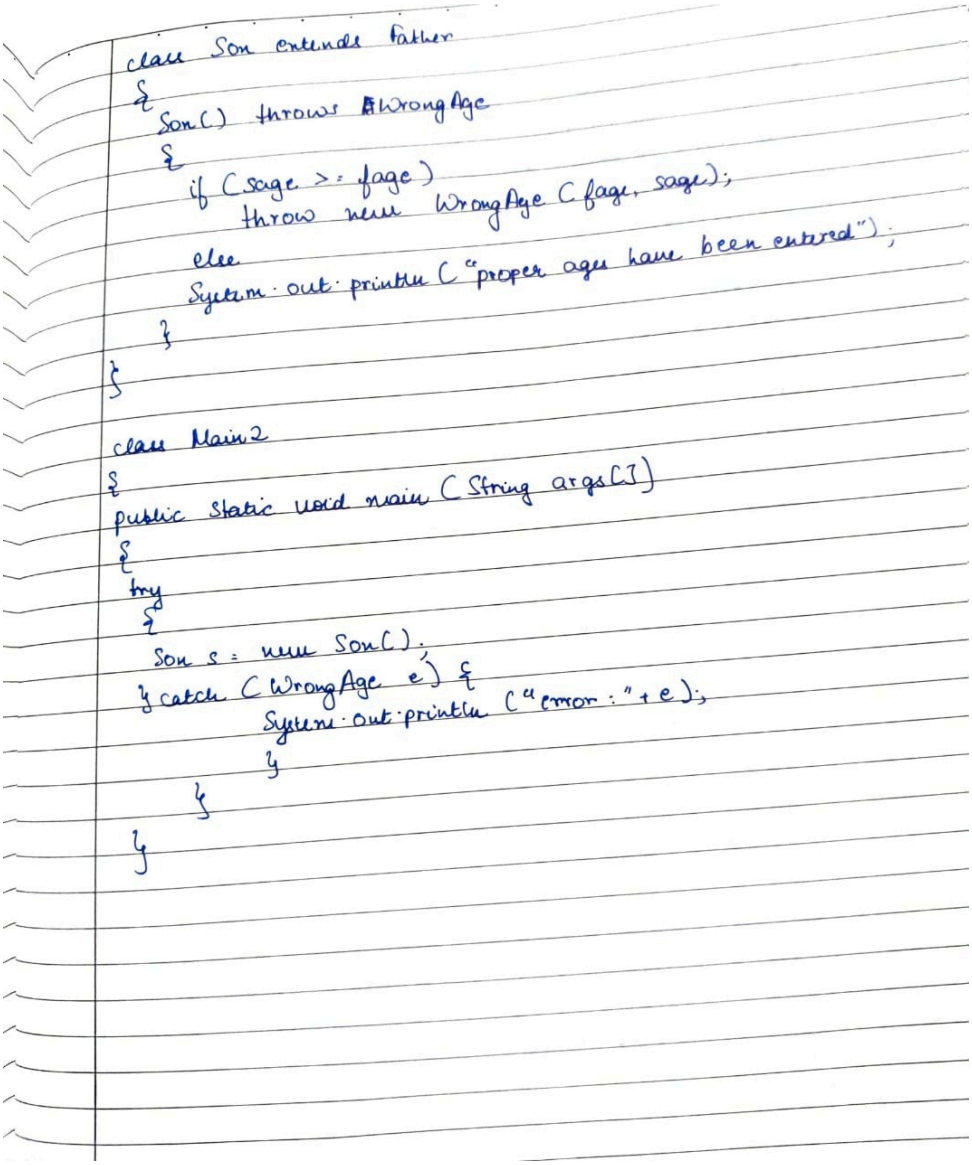
```
[Arvinds-MacBook-Pro:ooj Arvind$ java SimpleGen
enter a integer,string,float
12
anvi
2,5
Type of T is java.lang.Integer
Type of V is java.lang.String
Type of W is java.lang.Float
value1: 12
value2: anvi
value3: 2.5
enter a Character,Double,float
a
4,5555555
3,5
Type of T is java.lang.Character
Type of V is java.lang.Double
Type of W is java.lang.Float
value1: a
value2: 4.5555555
value3: 3.5
Arvinds-MacBook-Pro:ooj Arvind$
```

LAB 8

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception Wrong Age( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.



```
Week 10  Lab 8

import java.util.*;
class WrongAge extends Exception
{
private int a1, b1;
 WrongAge(int a, int b)
 {
  a1 = a;
  b1 = b;
 }
 public String toString()
 {
 if(a1<0 || b1<0)
 return "input age cannot be less than 0";
 else if(a1<=b1)
 return "father age cannot be less than or equal to
          son age";
 }
}
class father
 {
 int fage,sage;
 Scanner sc= new Scanner(System.in);
 Father() throws WrongAge
 {
 System.out.println("Enter the age of father");
 fage = sc.nextInt();
 System.out.println("Enter the age of son");
 sage = sc.nextInt();
 if(fage<0 || sage<0)
    throw new WrongAge(fage, sage);
 }
}
```

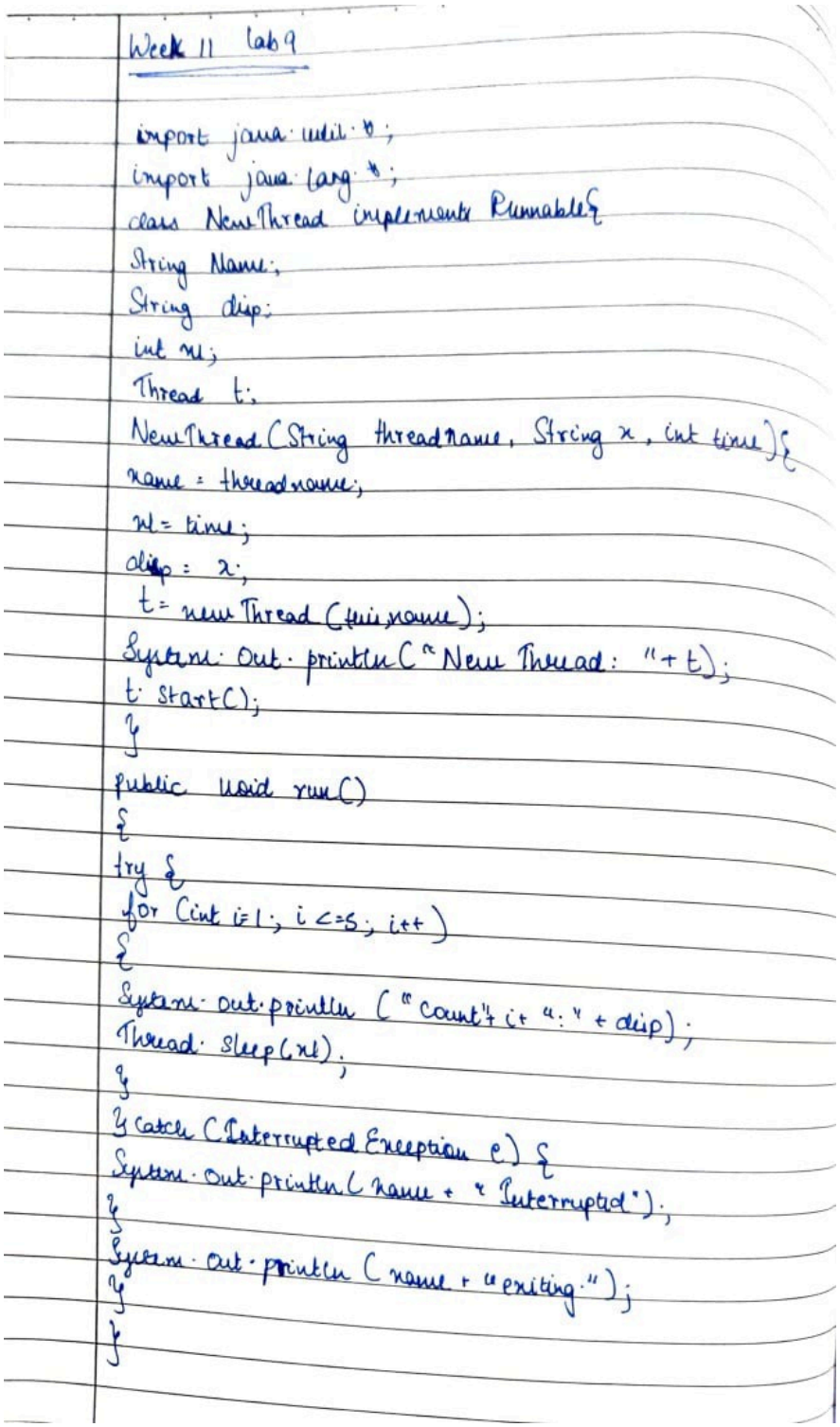


```
class Son extends Father
{
  Son() throws WrongAge
  {
    if (sage >= fage)
      throw new WrongAge (fage, sage);
    else
    System.out.println ("proper ages have been entered");
  }
}

class Main2
{
public static void main (String args[])
{
 try
  {
   Son s = new Son();
 } catch (WrongAge e) {
        System.out.println ("error:"+e);
        }
    }
}
```

```
[Arvinds-MacBook-Pro:ooj Arvind$ java Main2
enter the age of father
20
enter the age of son
40
error:father age cannot be less than or equal to son age
[Arvinds-MacBook-Pro:ooj Arvind$ java Main2
enter the age of father
-3
enter the age of son
-2
error:input age cannot be less than 0
Arvinds-MacBook-Pro:ooj Arvind$
```

LAB 9

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.



```
Week 11 lab 9

import java.util.*;
import java.lang.*;
class NewThread implements Runnable{
String Name;
String disp;
int xl;
Thread t;
NewThread(String threadname, String x, int time){
name = threadname;
xl = time;
disp = x;
t = new Thread(this,name);
System.out.println("New Thread: "+t);
t.start();
}
public void run()
{
try {
for(int i=1; i<=5; i++)
{
System.out.println("Count"+i+": "+disp);
Thread.sleep(xl);
}
}catch(InterruptedException e){
System.out.println(name + "Interrupted");
}
System.out.println(name + "exiting.");
}
}
```

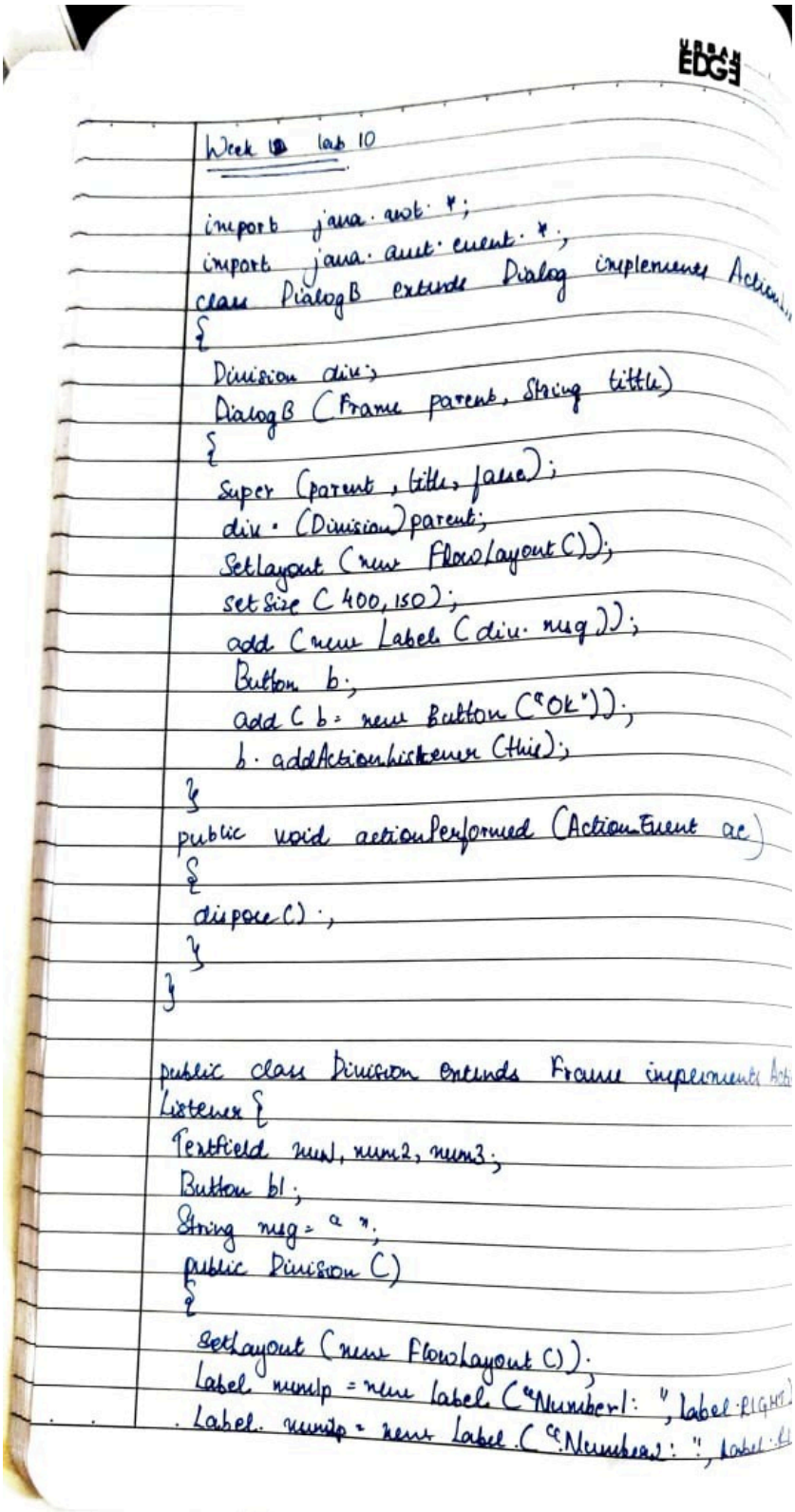
URBAN EDGE

```
class ThreadDemo {
static
public static void main (String args[])
{
new NewThread ("thread1", "BMS College of Engineering",
                10000);
new NewThread ("thread2", "CSE", 2000);
try {
Thread.sleep (51000);
System.out.println("-----------");
System.out.println ("Main Thread ended");
} catch (Interrupted e) {
System.out.println ("Main Thread Interrupted");
}
}
}
```

```
Arvinds-MacBook-Pro:ooj Arvind$ java ThreadDemo
New thread: Thread[thread1,5,main]
New thread: Thread[thread2,5,main]
count 1:CSE
count 1:BMS College of Engineering
count 2:CSE
count 3:CSE
count 4:CSE
count 5:CSE
count 2:BMS College of Engineering
thread2 exiting.
count 3:BMS College of Engineering
count 4:BMS College of Engineering
count 5:BMS College of Engineering
thread1 exiting.
--------------------------------
Main Thread ended
```

LAB 10

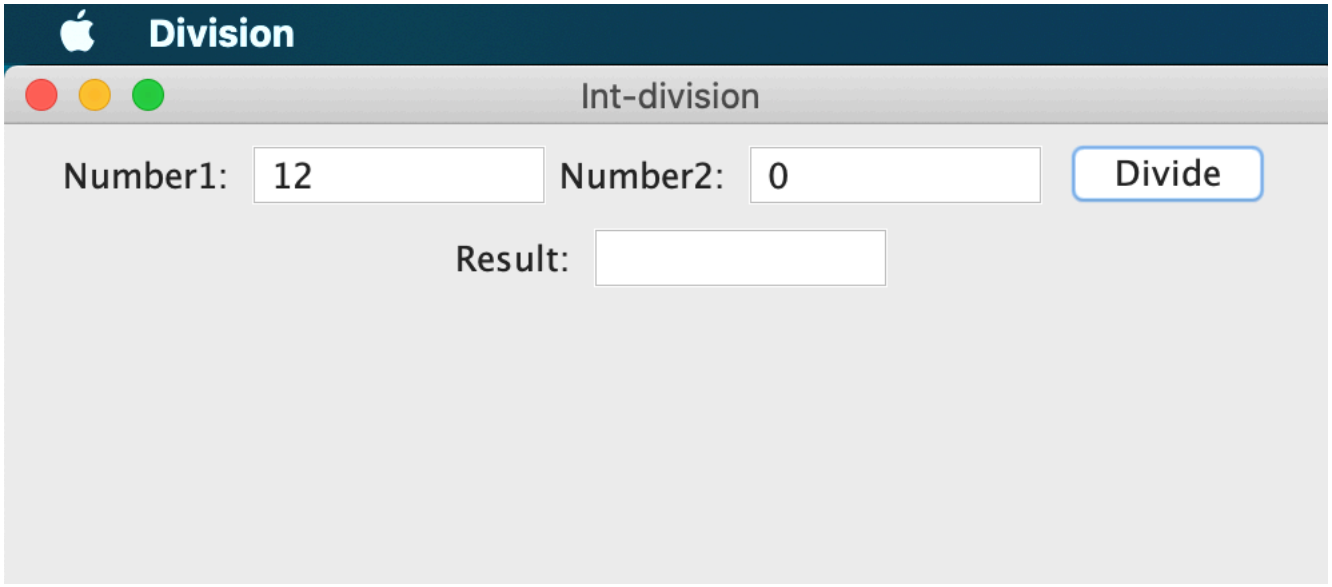
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

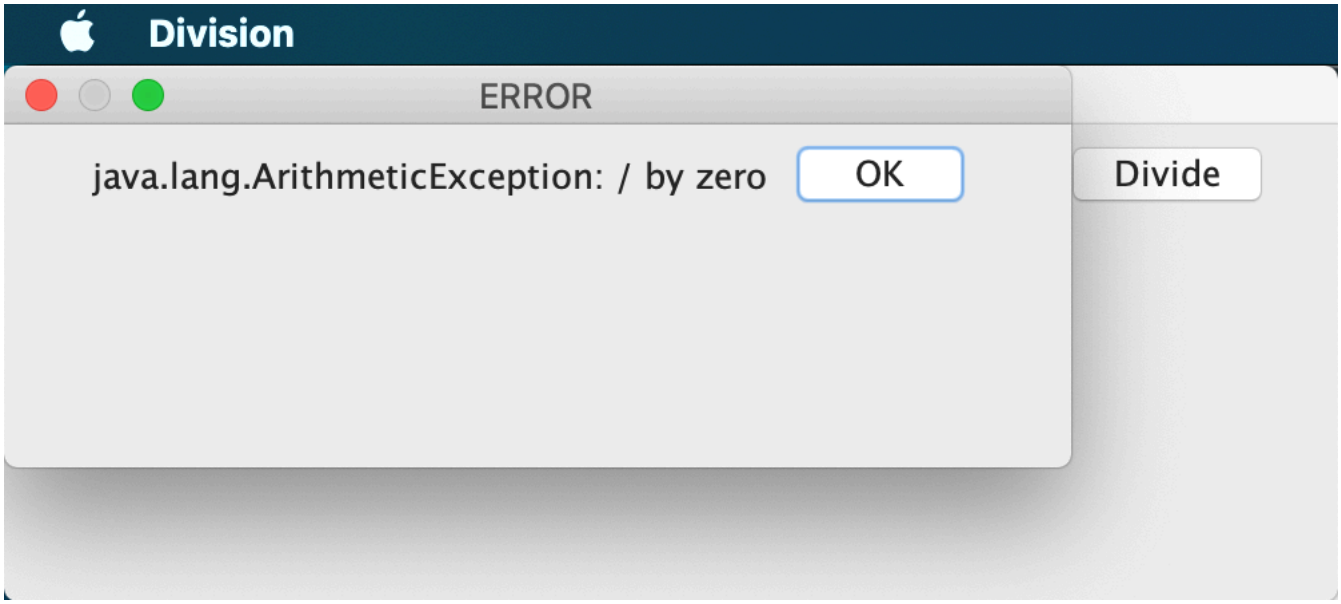


```
Week 10 lab 10

import java.awt.*;
import java.awt.event.*;
class DialogB extends Dialog implements Action...
{
 Division div;
 DialogB (Frame parent, String title)
 {
  super (parent, title, false);
  div = (Division)parent;
  setLayout (new FlowLayout());
  setSize (400,150);
  add (new Label (div.msg));
  Button b;
  add (b = new Button ("OK"));
  b.addActionListener (this);
 }
public void actionPerformed (ActionEvent ae)
 {
  dispose();
 }
}

public class Division extends Frame implements Act...
Listener {
 TextField num1, num2, num3;
 Button b1;
 String msg = " ";
 public Division()
 {
  setLayout (new FlowLayout());
  Label num1p = new Label ("Number1: ", Label.RIGHT...
  Label num2p = new Label ("Number2: ", Label.R...
```

```
b1 = new Button("Divide");
Label resultp = new Label("result:", Label.RIGHT);
num1 = new Textfield(12);
num2 = new Textfield(12);
result = new Textfield(12);
add(num1p);
add(num1);
add(num2p);
add(num2);
add(b1);
add(resultp);
add(result);
num1.addActionListener(this);
num2.addActionListener(this);
b1.addActionListener(this);
addWindowListener(new WindowAdapter()
{
public void windowClosing(WindowEvent we){
    System.exit(0);
}
});
}

public void actionPerformed(ActionEvent ae)
{
  if(!(num.getText().equals("")) && !(num2.getText()
  .equals(
  try
  {
    int x = Integer.parseInt(num1.getText());
    int y = Integer.parseInt(num2.getText());
    int z;
    z = x/y;
    msg = "" + z;
  }
```

```
        catch (NumberFormatException e)
        {
        msg = "" + e;
        result.setText("");
        DialogB d = new DialogB(this, "Error");
        d.setVisible(true);
        }
        catch (ArithmeticException e)
        {
        msg = "" + e;
        result.setText("");
        DialogB d = new DialogB(this, "Error");
        d.setVisible(true);
        }
      }
      else
        {
        msg = "Number fields should NOT be EMPTY!";
        result.setText("");
        d.setVisible(true);
        }
    }
    public void paint(Graphics g)
    {
      result.setText(msg);
    }
    public static void main(String args[])
    {
    Division a = new Division();
      a.setSize(new Dimension(500,200));
      a.setTitle("Int-Division");
      a.setVisible(true);
    }
}
```


Division
Int-division
Number1:
12
Number2:
0
Divide
Result:


Division
ERROR
java.lang.ArithmeticException: / by zero
OK
Divide