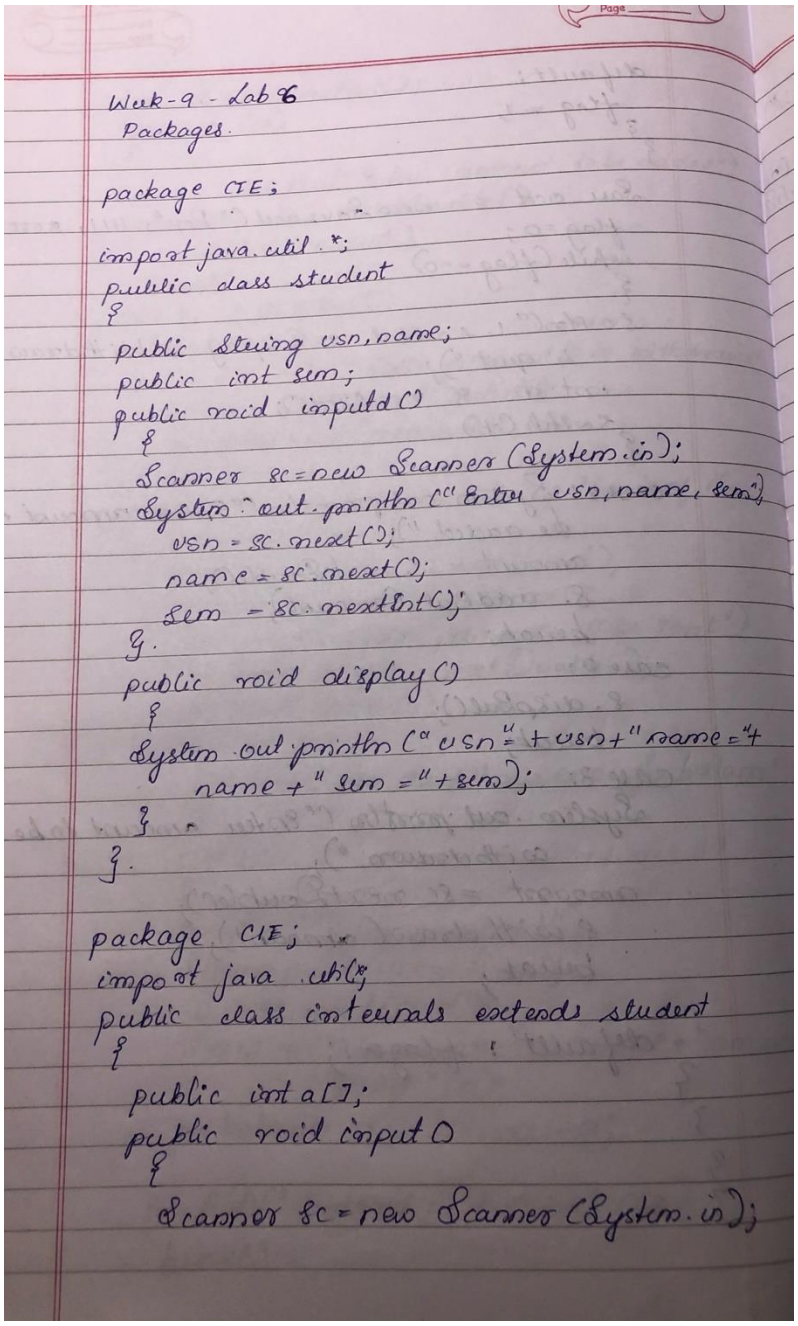
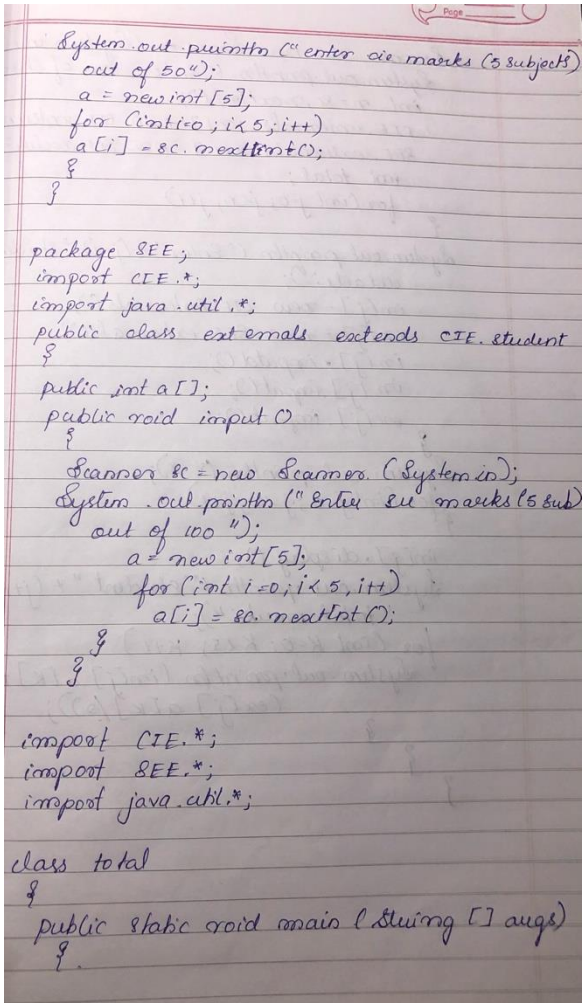
Lab program 6:

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

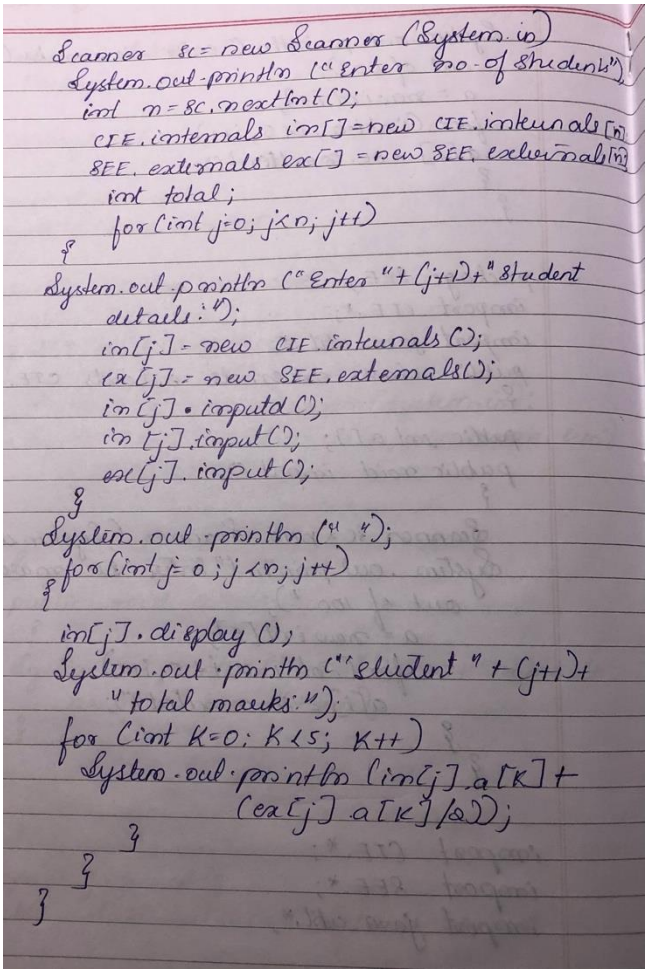
Program:



```
Week-9 - Lab 6
Packages

package CIE;

import java.util.*;
public class student
{
public String usn, name;
public int sem;
public void inputd()
 {
 Scanner sc=new Scanner(System.in);
 System.out.println("Enter usn, name, sem");
   usn = sc.next();
   name = sc.next();
   sem = sc.nextInt();
 }
public void display()
 {
 System.out.println("usn="+usn+"name="+
    name+"sem="+sem);
 }
}

package CIE;
import java.util;
public class internals extends student
{
  public int a[];
  public void input()
   {
    Scanner sc=new Scanner(System.in);
```



```
System.out.println("enter cie marks (5 subjects)
  out of 50");
  a = new int[5];
  for(int i=0; i<5; i++)
  a[i] = sc.nextInt();
  }
}

package SEE;
import CIE.*;
import java.util.*;
public class externals extends CIE.student
{
public int a[];
public void input()
  {
  Scanner sc = new Scanner(System.in);
  System.out.println("Enter see marks (5 sub)
    out of 100");
      a = new int[5];
      for(int i=0; i<5; i++)
        a[i] = sc.nextInt();
  }
}

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

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



```
Scanner sc= new Scanner(System.in)
System.out.println("enter no of studnts");
 int n=sc.nextInt();
 CIE.internals in[]=new CIE.internals[n]
 SEE.externals ex[] = new SEE.externals[n]
  int total;
  for(int j=0; j<n; j++)
{
System.out.println("Enter "+(j+1)+" student
    details:");
    in[j] = new CIE.internals();
    ex[j] = new SEE.externals();
    in[j].inputd();
    in[j].input();
    ex[j].input();
}
System.out.println("");
for(int j= 0; j<n; j++)
{
 in[j].display();
 System.out.println("student "+(j+1)+
    "total marks:");
 for(int K=0; K<5; K++)
  System.out.println(in[j].a[K]+
          (ex[j].a[K]/2));
    }
  }
}
```

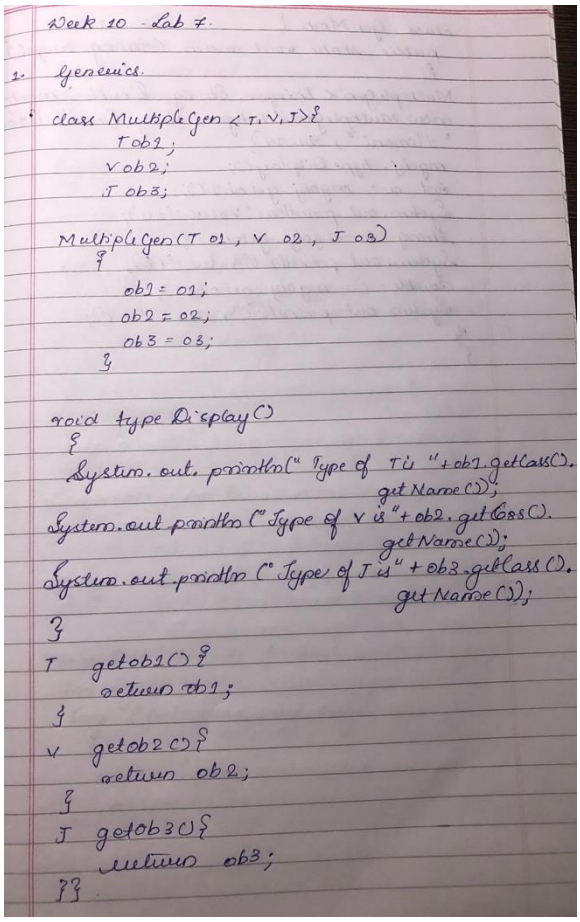
Output:

```
C:\Users\Hima\Desktop\java\pack>javac internals.java

C:\Users\Hima\Desktop\java\pack>javac student.java

C:\Users\Hima\Desktop\java\pack>javac externals.java

C:\Users\Hima\Desktop\java\pack>javac total.java

C:\Users\Hima\Desktop\java\pack>java total
enter number of students
3
enter1student details:
enter usn,name,sem
12jnvjks himani 3
enter cie marks(5 subjects) out of 50
12 34 56 78 90
enter see marks(5 subjects) out of 100
90 90 90 90 90
enter2student details:
enter usn,name,sem
sdwbyer64 bora 3
enter cie marks(5 subjects) out of 50
12 34 56 78 90
enter see marks(5 subjects) out of 100
80 80 80 80 80
enter3student details:
enter usn,name,sem
6gy7 tera 4
enter cie marks(5 subjects) out of 50
09 87 65 43 21
enter see marks(5 subjects) out of 100
89 89 89 89 89

usn=12jnvjks  name=himani  sem=3

student 1 total marks:
```

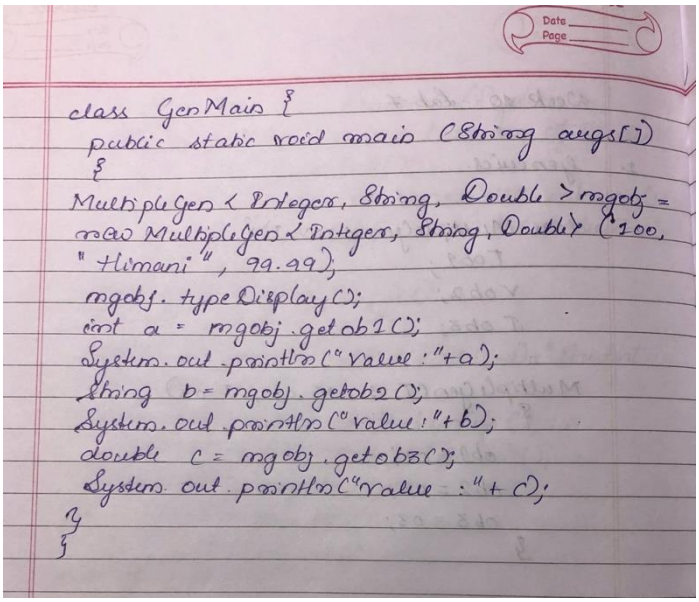
```
enter3student details:
enter usn,name,sem
6gy7 tera 4
enter cie marks(5 subjects) out of 50
09 87 65 43 21
enter see marks(5 subjects) out of 100
89 89 89 89 89

usn=12jnvjks  name=himani  sem=3

student 1 total marks:
57
79
101
123
135
usn=sdwbyer64  name=bora  sem=3

student 2 total marks:
52
74
96
118
130
usn=6gy7  name=tera  sem=4

student 3 total marks:
53
131
109
87
65
```

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

Program:



```
Week 10 - Lab 7.

1. Generics.

class MultipleGen <T, V, J>{
    T ob1;
    V ob2;
    J ob3;

MultipleGen(T o1, V o2, J o3)
  {
    ob1 = o1;
    ob2 = o2;
    ob3 = o3;
  }

void typeDisplay()
 {
 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 J is "+ob3.getClass().
                                    getName());
}
T getob1(){
   return ob1;
}
V getob2(){
   return ob2;
}
J getob3(){
   return ob3;
}}
```



```
class GenMain {
 public static void main (String args[])
 {
MultipleGen < Integer, String, Double > mgobj =
new MultipleGen < Integer, String, Double > (100,
" Himani ", 99.99);
 mgobj. typeDisplay();
 int a = mgobj.getob1();
 System.out.println(" value :"+a);
 String b = mgobj.getob2();
 System.out.println(" value :"+b);
 double c = mgobj.getob3();
 System.out.println(" value :"+c);
}
}
```

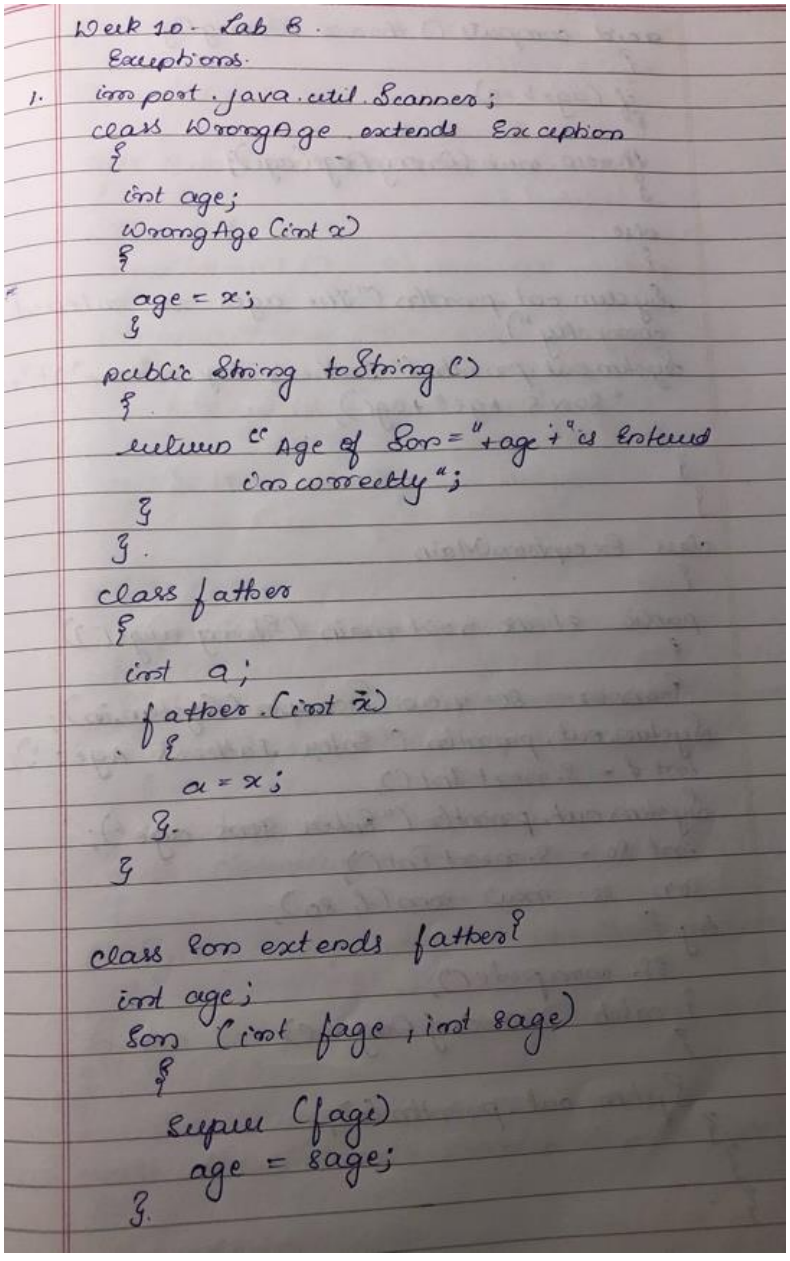
Output:

```
C:\Users\Hima\Desktop\java>javac GenMain.java

C:\Users\Hima\Desktop\java>java GenMain
Type of T is java.lang.Integer
Type of V is java.lang.String
Type of J is java.lang.Double
Value: 100
Value: himani
Value: 99.99
```

Program8:

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.

Program:



```
Week 10 - Lab 8.
Exceptions.
1. import java.util.Scanner;
class WrongAge extends Exception
{
 int age;
 WrongAge(int x)
 {
  age = x;
 }
 public String toString()
 {
  return "Age of Son = "+age+" is entered
          incorrectly";
 }
}.
class father
{
 int a;
 father.(int x)
  {
    a = x;
  }.
}

class Son extends father{
 int age;
 Son (int fage, int sage)
  {
   super(fage)
    age = sage;
}.
```

```
void compute() throws WrongAge
 {
  if (age>=a)
  {
  throw new WrongAge(age);
  }
  else
  {
  System.out.println("The ages are entered
  correctly");
  System.out.println("Father's Age = "+a+"\t"+
     "Son's Age"+age);
  }
 }
}
class ExceptionsMain
 {
 public static void main (String args[])
  {
  Scanner sc= new Scanner(System.in);
 System.out.println("Enter Father's Age:");
 int f = s.nextInt();
 System.out.println("Enter son's age");
 int so = s.nextInt();
 Son ss = new Son(f, so);
 try {
   ss.compute();
  } catch (WrongAge e)
  {
  System.out.println(e);
 }
}
}
```

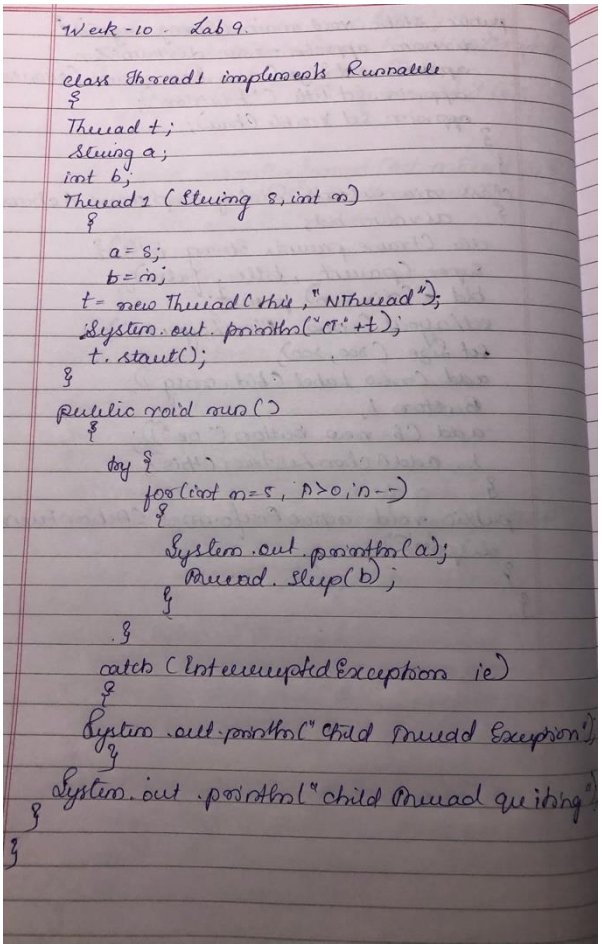
Output :

```
C:\Users\Hima\Desktop\java>javac ExceptionsMain.java

C:\Users\Hima\Desktop\java>java ExceptionsMain
ENTER FATHER'S AGE:
55
ENTER SON'S AGE:
6
THE AGES ARE ENTERED CORECTLY
FATHER'S AGE=55 SON'S AGE=6

C:\Users\Hima\Desktop\java>javac ExceptionsMain.java

C:\Users\Hima\Desktop\java>java ExceptionsMain
ENTER FATHER'S AGE:
12
ENTER SON'S AGE:
56
AGE OF SON=56 IS ENTERED INCORRECTLY

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

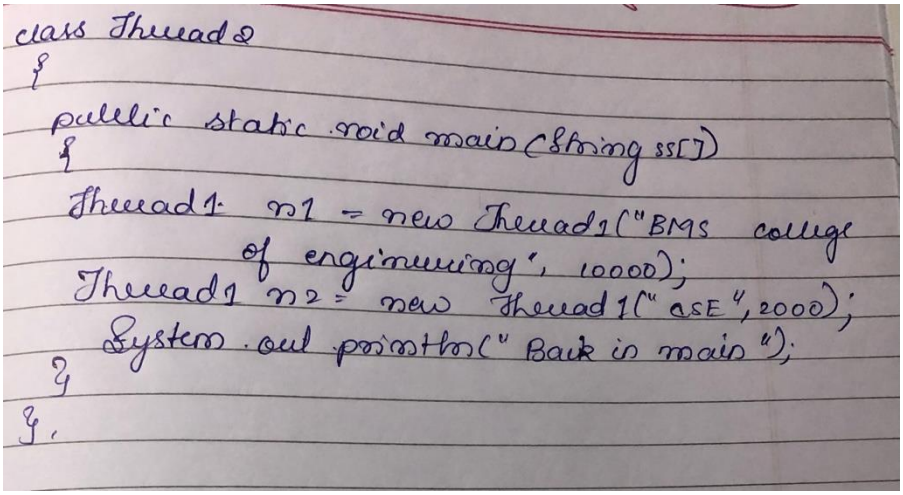
Lab program 9:
Write a program which creates two threads, one thread displaying “BMS College of

Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

Program :



```
Week-10 Lab 9

class Thread1 implements Runnable
{
Thread t;
String a;
int b;
Thread1 (String s, int m)
  {
    a = s;
    b = m;
  t = new Thread(this, "NThread");
  System.out.println("o:" +t);
  t.start();
}
public void run()
  {
    try {
      for(int n=5; n>0; n--)
        {
          System.out.println(a);
          Thread.sleep(b);
        }
    }
    catch (InterruptedException ie)
    {
    System.out.println("Child Thread Exception");
    }
System.out.println("child Thread quiting");
}
}
```



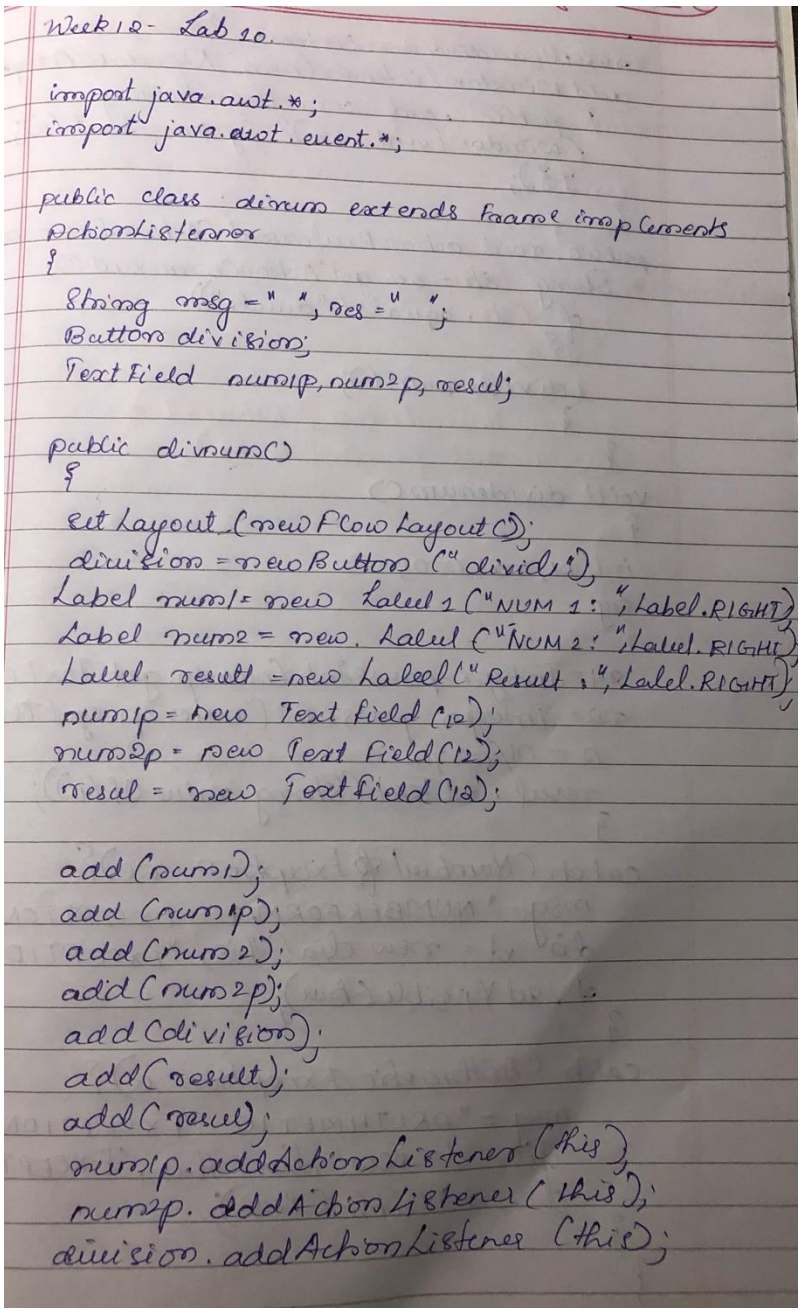
```
class Thread2
{
public static void main(String ss[])
{
Thread1 n1 = new Thread1("BMS college
        of engineering", 10000);
Thread1 n2 = new Thread1("CSE", 2000);
System.out.println("Back in main");
}
}.
```

Output :

```
C:\Users\Hima\Desktop\java>javac Threads.java

C:\Users\Hima\Desktop\java>java Thread2
CT:Thread[NThread,5,main]
CT:Thread[NThread,5,main]
BMS college of engineering
CSE
Back in main
CSE
CSE
CSE
CSE
BMS college of engineering
Child Thread quitting
BMS college of engineering
BMS college of engineering
BMS college of engineering
Child Thread quitting
```

Lab program 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 12- Lab 10.

import java.awt.*;
import java.awt.event.*;

public class divnum extends Frame implements
ActionListener
{
 String msg = " ", res = " ";
 Button division;
 TextField num1p, num2p, resul;

 public divnum()
 {
  setLayout (new FlowLayout());
  division = new Button ("divide");
  Label num1 = new Label ("NUM 1: ", Label.RIGHT);
  Label num2 = new Label ("NUM 2: ", Label.RIGHT);
  Label result = new Label ("Result :", Label.RIGHT);
  num1p = new TextField (10);
  num2p = new TextField (12);
  resul = new TextField (12);

  add (num1);
  add (num1p);
  add (num2);
  add (num2p);
  add (division);
  add (result);
  add (resul);
  num1p.addActionListener (this);
  num2p.addActionListener (this);
  division.addActionListener (this);
```

```
resul.addActionListener(this);
addWindowListener(new WindowAdapter()
{ public void windowClosing
(WindowEvent we){ System.exit(0);
}});
}
public void actionPerformed(ActionEvent ae){
String str=ae.getActionCommand();
if(str.equals("divide"))
{
dividenum();
}
}
void dividenum()
{
int n1,n2,n=0;
try
{
n1=Integer.parseInt(num1p.getText());
n2=Integer.parseInt(num2p.getText());
n=n1/n2;
resul.setText(String.valueOf(n));
}
catch(NumberFormatException ne){
msg="NUMBERFORMAT EXCEPTION";
dia d=new dia(this,"EXCEPTION");
d.setVisible(true);
}
catch(ArithmeticException a){
msg="ARITHMETIC EXCEPTION";
dia d=new dia(this,"EXCEPTION");
d.setVisible(true);
}
}
```

```
public static void main(String args[]){
divnum appwin=new divnum();
appwin.setSize(new Dimension(250,150));
appwin.setTitle("Division");
appwin.setVisible(true);
}
}
class dia extends Dialog implements ActionListener
{ divnum bld;
dia(Frame parent,String title){
super(parent,title,false);
bld=(divnum)parent;
setLayout(new FlowLayout());
setSize(300,200);
add(new Label(bld.msg));
Button b;
add(b=new Button("OK"));
b.addActionListener(this);
}
public void actionPerformed(ActionEvent ae){
dispose();
}
}
```

Output: