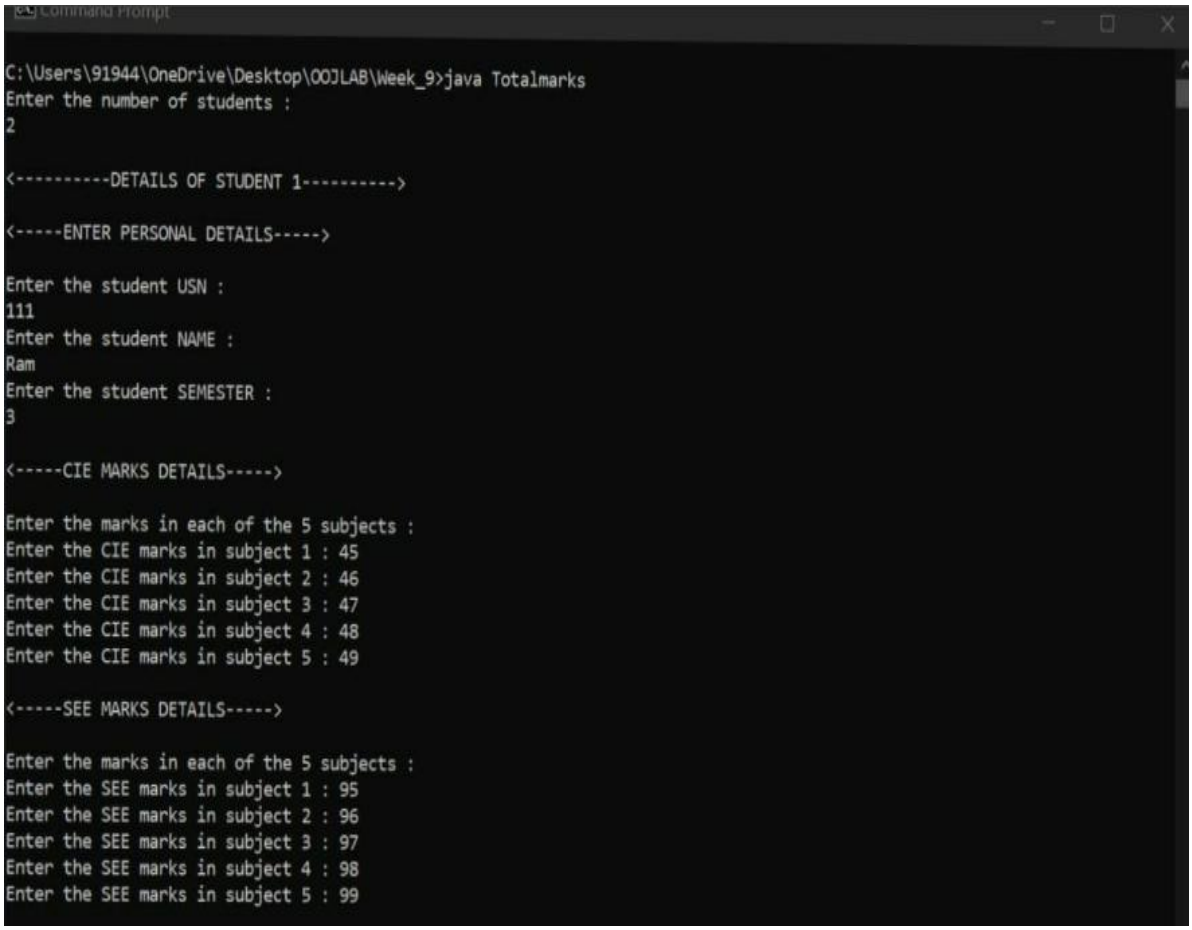
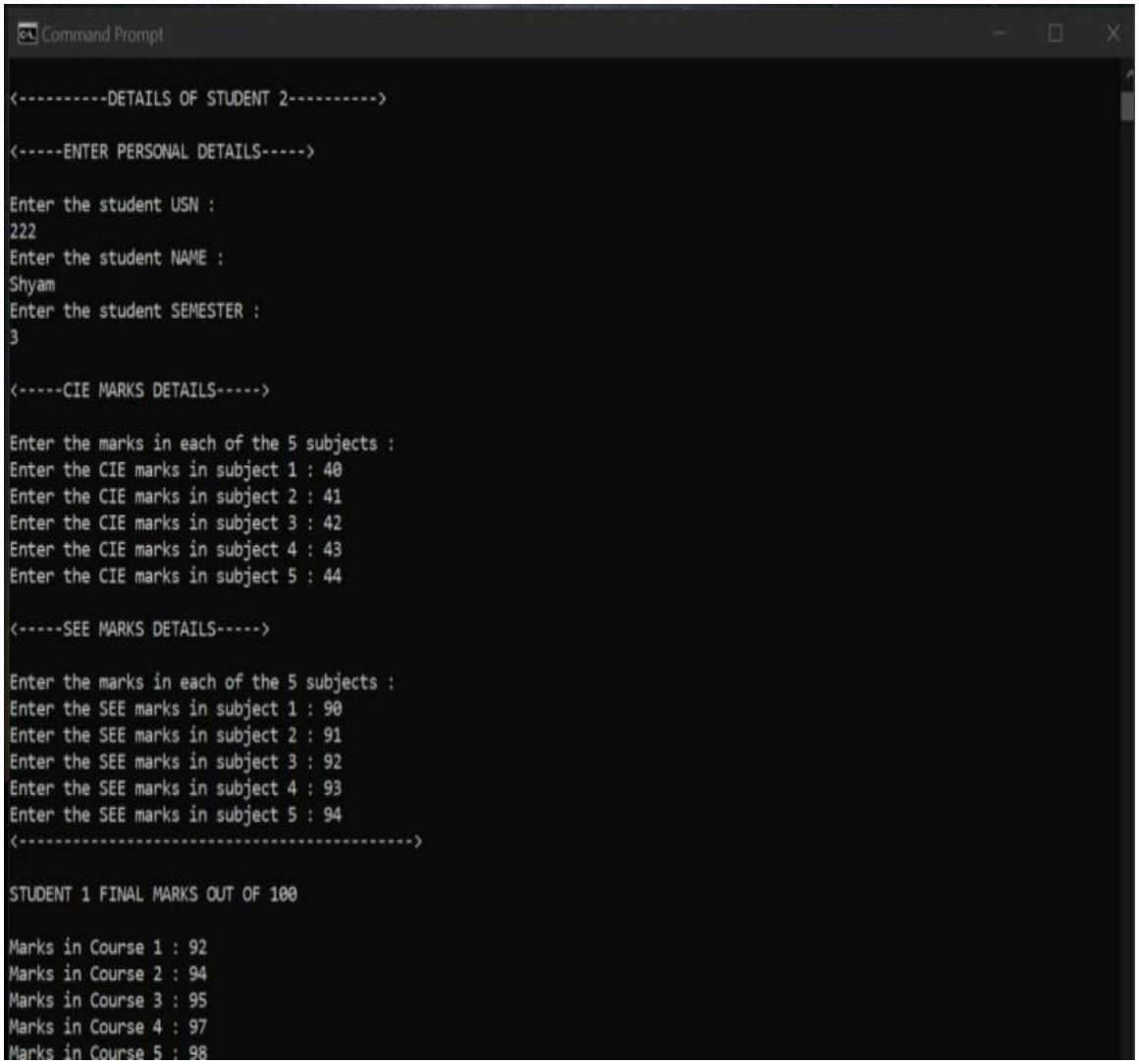
Lab record 2 -1BM19CS136

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.




Command Prompt
<----------DETAILS OF STUDENT 2---------->
<-----ENTER PERSONAL DETAILS----->
Enter the student USN :
222
Enter the student NAME :
Shyam
Enter the student SEMESTER :
3
<-----CIE MARKS DETAILS----->
Enter the marks in each of the 5 subjects :
Enter the CIE marks in subject 1 : 40
Enter the CIE marks in subject 2 : 41
Enter the CIE marks in subject 3 : 42
Enter the CIE marks in subject 4 : 43
Enter the CIE marks in subject 5 : 44
<-----SEE MARKS DETAILS----->
Enter the marks in each of the 5 subjects :
Enter the SEE marks in subject 1 : 90
Enter the SEE marks in subject 2 : 91
Enter the SEE marks in subject 3 : 92
Enter the SEE marks in subject 4 : 93
Enter the SEE marks in subject 5 : 94
<---------->
STUDENT 1 FINAL MARKS OUT OF 100
Marks in Course 1 : 92
Marks in Course 2 : 94
Marks in Course 3 : 95
Marks in Course 4 : 97
Marks in Course 5 : 98


Command Prompt
STUDENT 2 FINAL MARKS OUT OF 100
Marks in Course 1 : 85
Marks in Course 2 : 86
Marks in Course 3 : 88
Marks in Course 4 : 89
Marks in Course 5 : 91
<---------->

PROGRAM-6. 1BM19CS136

[student.java]

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

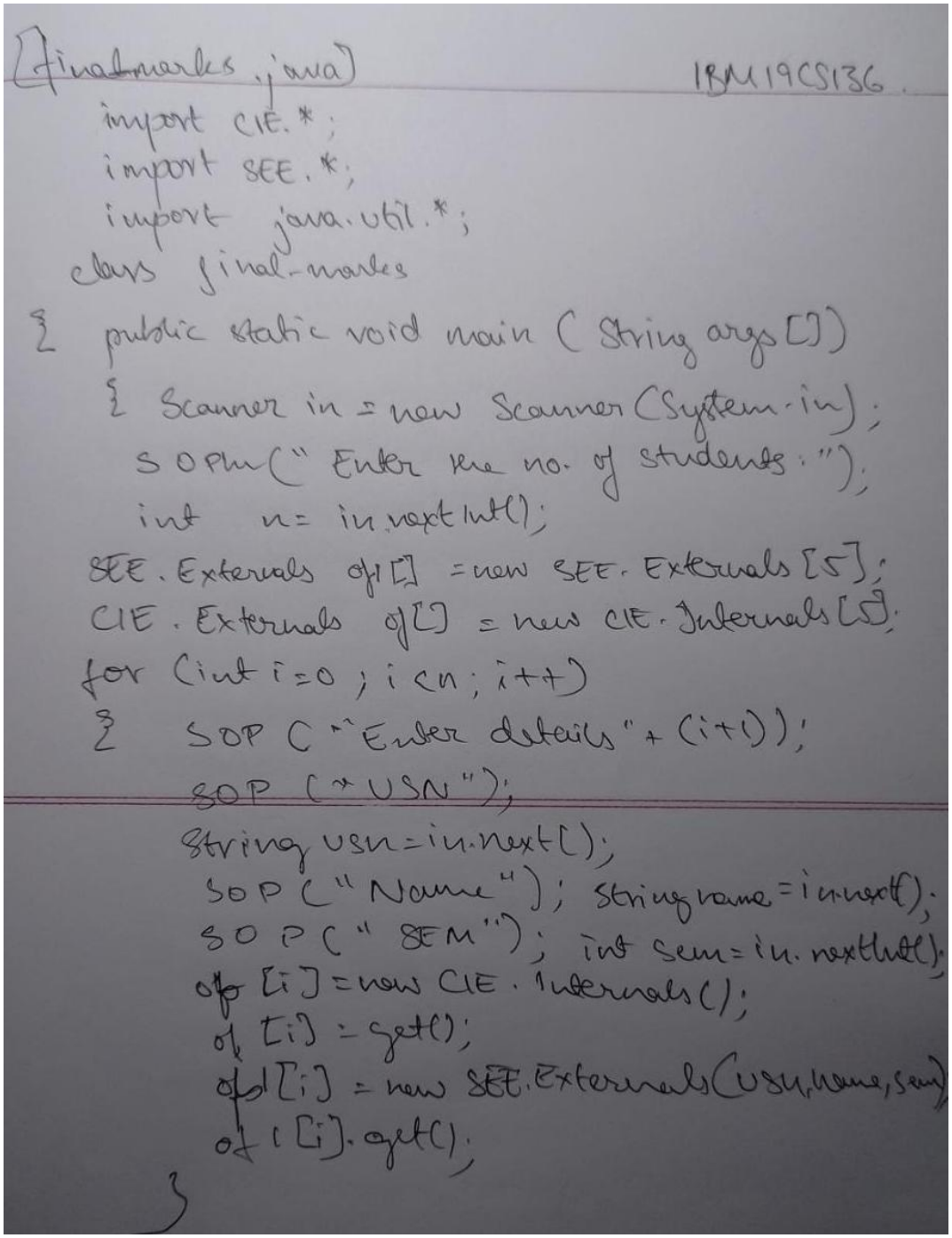
[internals.java]

```
package CIE;
import java.util.Scanner;
public class Internals extends Student
{ Scanner in=new Scanner(System.in);
  public int[]
        cie=new int[5];
  public void get()
{ for(int i=0;i<5;i++)
  { SOP("Enter CIE"+(i+1)+" out of 50) of current
    sem"); cie[i]=in.nextInt();} }
```

1BM19CS136

[External.java]

```
package SEE;
import java.util.Scanner;
public class Externals extends CIE Student
{ public Externals (String usn, String name, int sem)
  { super (usn, name, sem); }
 Scanner in = new Scanner (System.in);
 public int[] see = new int[5];
  public void get()
  { System.out.print("\n");
    for (int i=0; i<5; i++)
    SOP (" Enter the SEE " + (i+1) + " out of 100");
    see[i] = in.nextInt();
333}
```



```
finalmarks.java                                    1BM19CS136.
import CIE.*;
import SEE.*;
import java.util.*;
class final_marks
{ public static void main (String args[])
  { Scanner in = new Scanner(System.in);
    SOPln("Enter the no. of students:");
    int n= in.nextInt();
SEE.Externals ob[] =new SEE.Externals[5];
CIE.Externals ob[] = new CIE.Internals[5];
for (int i=0; i<n; i++)
{  SOP ("Enter details" + (i+1));
   SOP ("USN");
   String usn=in.next();
    SOP("Name"); String name=in.next();
    SOP("SEM"); int sem=in.nextInt();
    ob[i]=new CIE.Internals();
    ob[i] = get();
    ob[i] = new SEE.Externals(usn,name,sem);
    ob[i].get();
}
```

Program 7

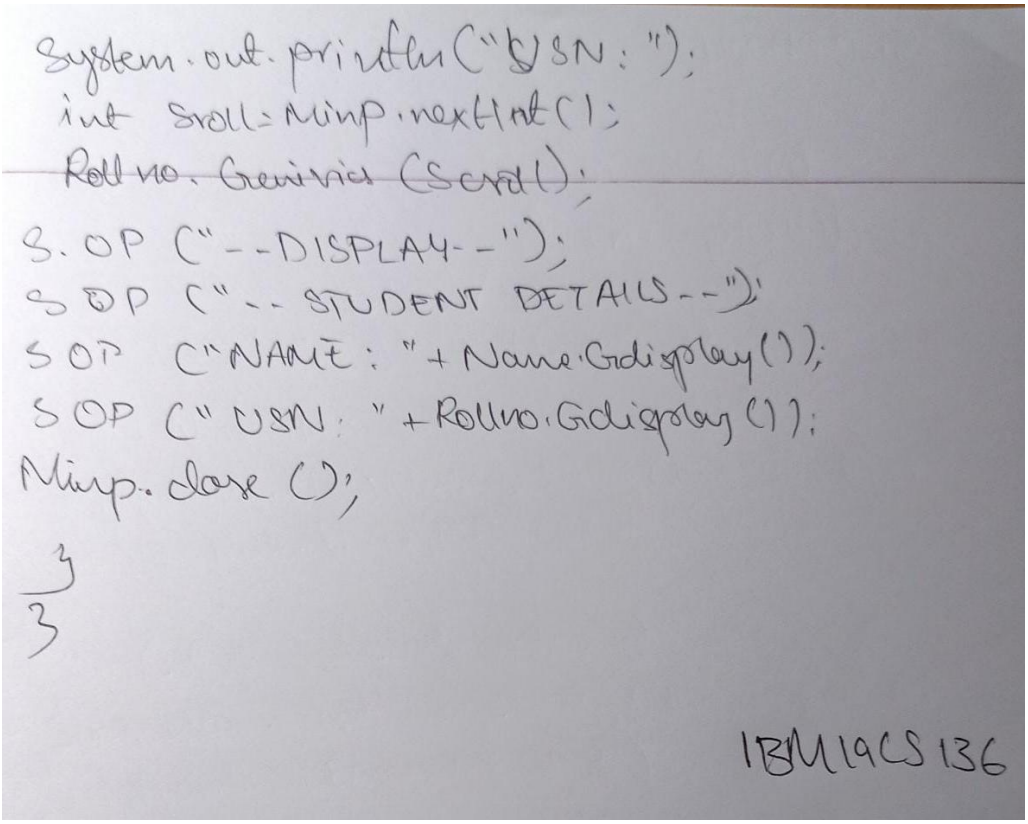
Write a program to demonstrate generics with multiple object parameters

```
--PLEASE ENTER STUDENT DETAILS--
NAME:
xyz
USN:
1234
-------DISPLAY-------
--STUDENT DETAILS--
NAME: xyz
USN: 1234

[Program finished]
```

PROGRAM-7

RM19CS136

```
import java.util.*;
class Generics<T> {
  T var1;
void Genirics(T gvar)
 { var1 = gvar;
  }
  TGdisplay() {
   return var1;
  }
}
public class App
{  public static void main(String[] args)
   throws Exception {
    SOP("--PLEASE ENTER STUDENT DETAILS--");
Scanner Minp = new Scanner(System.in);
Generics<Integer> Roll no = new Generic Integer();
Generics<String> Name = new Genrc<Stiy>();
SOP("NAME");
String Sname = Minp.nextLine();
Name.Genirics(Sname);
```



Program 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 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

```
C:\Users\Shree Lakshmi\Desktop>java exception_handling
Enter Father's age:
40
Enter Son's age:
56
Exception: Enter correct ages as Father's age can't be less than or equal to Son's age.

C:\Users\Shree Lakshmi\Desktop>
```

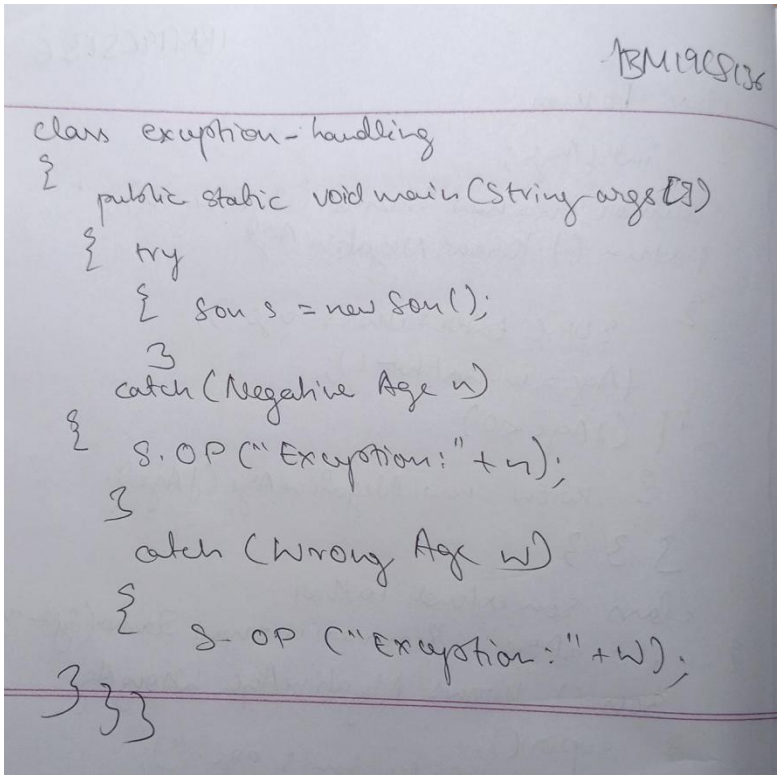
PROGRAM-8

1BM19CS136.

```
import java.util.*;
class WrongAge extends Exception
{
    int f,s;
  WrongAge (int fAge, int sAge)
   { f=fAge;
     s=sAge;
   }
   public String toString()
  { return "Enter correct ages as Father's age
    can't be less than/equal to Son's age.";
  }
}
class NegativeAge extends Exception
{ int x;
   NegativeAge (int fAge)
   {    x=fAge;
    }
     public String toString()
    {   return "Age can't be negative";
    }
}
```

1BM19CS136

```
class Father
{   int fAge;
Scanner in = new Scanner(System.in);
Father() throws NegativeAge
{
    SOP("Enter Father's age");
   fAge = in.nextInt();
 if (fAge<0)
   { throw new NegativeAge(fAge);
} } }
class Son extends Father
{  int sAge; Scanner in = new Scanner(System.in);
   Son() throws NegativeAge, WrongAge
{  super();
   SOP("Enter son's age:");
        sAge = in.nextInt();
         if (sAge<0)
      {  throw new NegativeAge(sAge);
      }
      if (sAge>=fAge)
      { throw new WrongAge(fAge, sAge);
} } }
```



Program 9

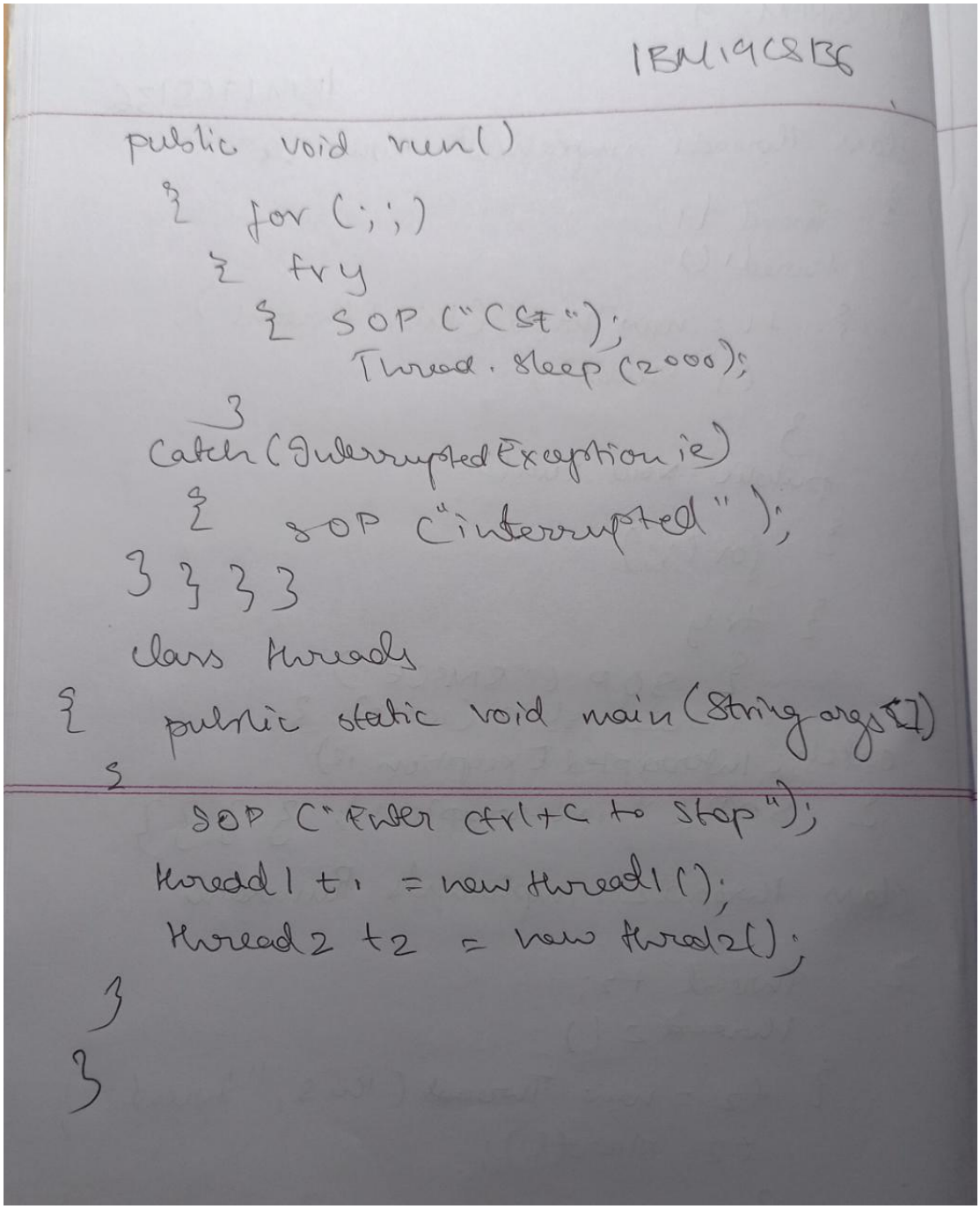
Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

```
C:\Users\Shree Lakshmi\Desktop>javac threads.java

C:\Users\Shree Lakshmi\Desktop>java threads
Enter CTRL+C to stop
BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE

C:\Users\Shree Lakshmi\Desktop>
```

PROGRAM- 9

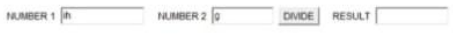
1BM19CS136.

```
class thread1 implements Runnable
{ Thread t1;
  thread1()
  { t1 = new Thread (this, "thread1");
    t1.start();
  }
  public void run()
  { for(;;)
  { try
    { SOP ("BMSCE")
    Thread.sleep (10000); }
catch (InterruptedException ie)
  { SOP ("interrupted"); } } }

class thread2 implements Runnable
{   Thread t2;
    thread2()
  { t2 = new Thread (this, "thread2");
    t2.start();
  }
```



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



Program - 10. 1BM19CS136.

```
import javax.swing.*;
public class integerdivision extends Frame implements ActionListener
{
    TextField n1,n2,res;
    Label ln1,ln2,lres;
    Button b;
    public integerdivision()
    {
      setLayout(new FlowLayout());
Label ln1 = new Label("NUMBER 1", Label.RIGHT);
Label ln2 = new Label("NUMBER 2", Label.RIGHT);
Label lres = new Label("RESULT", Label.RIGHT);
n1 = new Textfield(12);
n2 = new Textfield(10);
b = new Button("DIVIDE");
add(ln1);
add(n1);
add(ln2);
add(n2);
add(b);
add(lres); add(res);
b.addActionListener(this);
addWindowListener(new WindowAdapter());
}
public void actionPerformed(ActionEvent ae)
{
```

IBM19CS136

```
if (ae.getSource() == b)
{ try {
    int num1 = Integer.parseInt (n1.getText());
    int num2 =      "       "       "  (n2.getText());
     "  num3 = num1 / num2;
   res.setText (String.valueOf (num3));
  }
  catch (NumberFormatException nfe)
  { JoptionPane.showMessageDialog (this, ne, "Error",
    JoptionPane.ERROR_MESSAGE);
  }
  catch (ArithmeticException a)
  { JoptionPane.showMessageDialog (this, a, "ERROR",
    JoptionPane.ERROR_MESSAGE);
  }}}
  public static void Main (String args[])
{   integerdivision i = new integerdivision();
    i.setSize (new Dimension (400, 400));
    i.setTitle ("Integer DIVISION OF TWO NUMBERS");
    i.setVisible (true);
  }
class WindowAdapter1 extends WindowAdapter
{ public void windowClosing (WindowEvent we)
{ System.exit(0);
  }}}
```