Sampreeth.P (1BM19CS142)

Q) 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 6

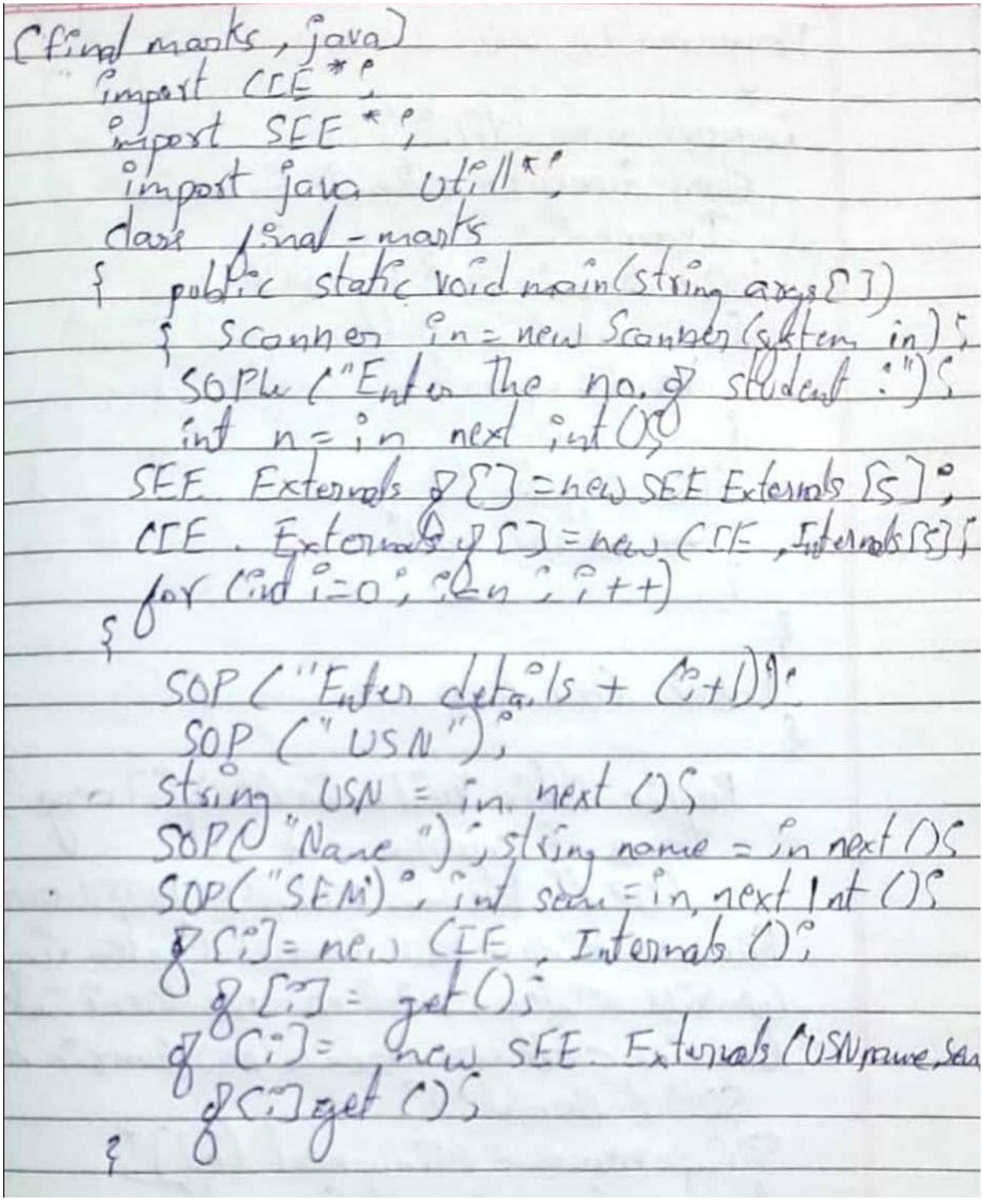
[student.java]

```
Package CIE;
Public class student
{
    Public string Usn;
    Public string name;
    Public int Sem;
    Public student() {}
    Public student (string Usn,
    {
        this.Usn = Usn;
        this name = name;
        this.Sem = Sem;
    }}
```

[internals.java]

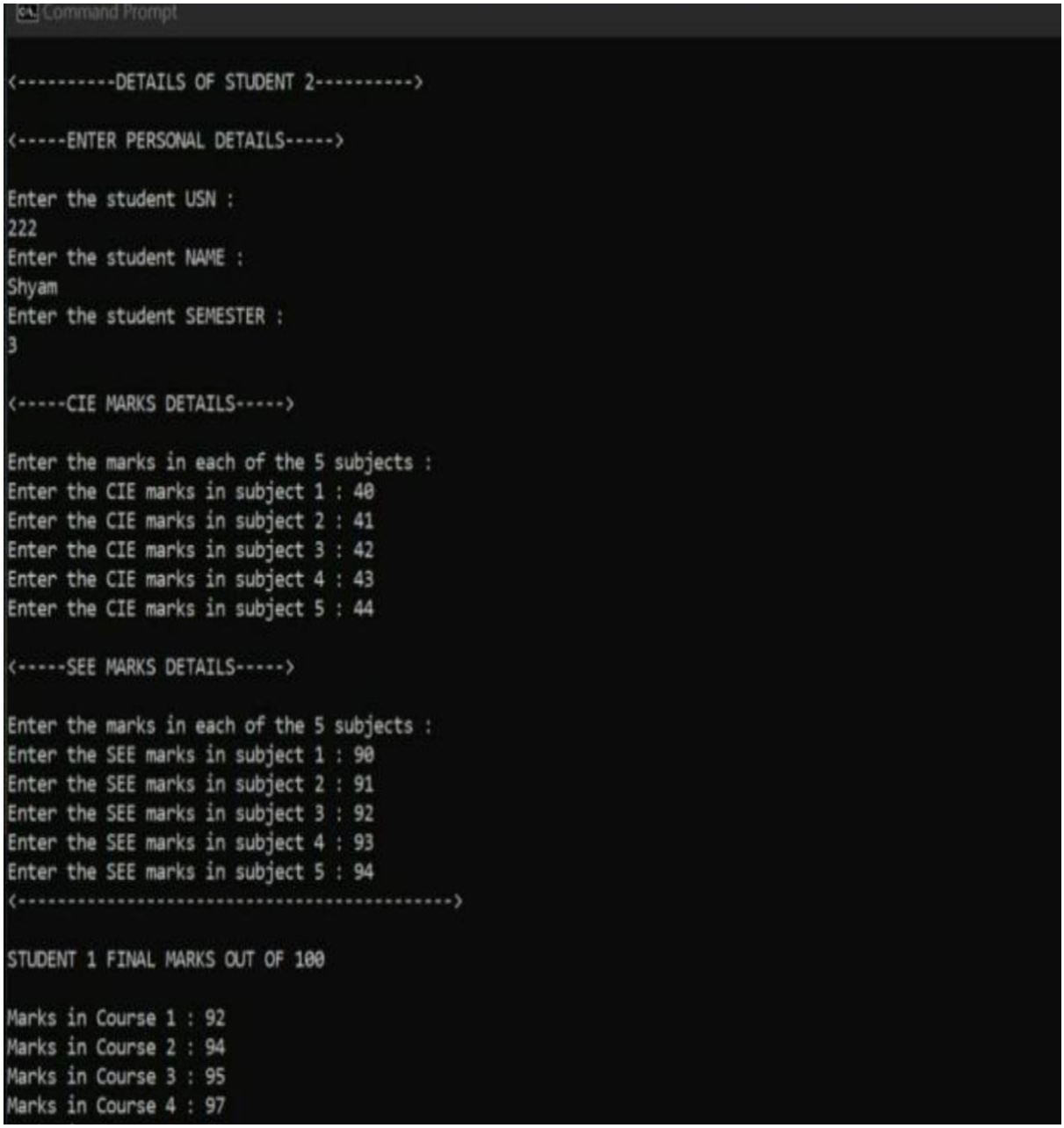
```
Package CIE;
import java.util.Scanner;
Public class internals extends student
{
    Scanner in = new Scanner (System.in);
    Public int [] ie = new int[5];
    Public void get()
    { for (int i=0; i<5; i++)
      { S.O.P ("Enter CIE" + (i+1) + " out of 50) &
                                        correct one);
        ie[i] = in.next int(); }}}
```

```
[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[] sec = new int [5];
  Public void get()
{
  System.out.print ("\n");
  for (int i=0; i<n; i++)
  SOP ("Enter the SEE" + (i+1) + "out of 100");
  See[i] = in.nextInt();
}}}
```



```
(final marks, java)
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 student :");
    int n = in.nextInt();
  SEE.Externals s[] = new SEE.Externals[5];
  CIE.Externals c[] = 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();
    c[i] = new CIE.Internals();
      c[i] = get();
      s[i] = new SEE.Externals(USN,name,sem
        s[i].get();
}
```

Output



Q) Lab Program-7:

Write a program to demonstrate generics with multiple object parameters.

Program - 7

```
import java, util.* ;
 class Generies <T> {
  Tvar1 ;
 Void Generies (T gvar)
  {
    Var1 = gvar;
  }
   TG display () {
    return var1;
  }
}
 Public Class App
{
   Public static Void main (String[] args)
    throws Exception   {
     SOP (" Ex -- PLEASE ENTER STUDENT DETA
 Scanner Minp = New Scanner (System.in);
 Genries <integer> Roll no = new Generic integer()
 Generies <string> Name = new Generic <string>
   SOP ("Name");
 String name = Minp.next (in);
 Name Generies ( Sname )S
```

```
System.out.printer("USN: ");
    int sroll = minp.nextInt();
Roll no, Genories (sroll);
    SOP("--- DISPLAY ---");
SOP("--- STUDENT DETAILS---");
SOP("NAME: " + Name.display());
SOP("USN: " + Roll.display());
    Minp.close();
}
}
```

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

[Program finished]
```

Q)

Lab 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 Wrong Age( )

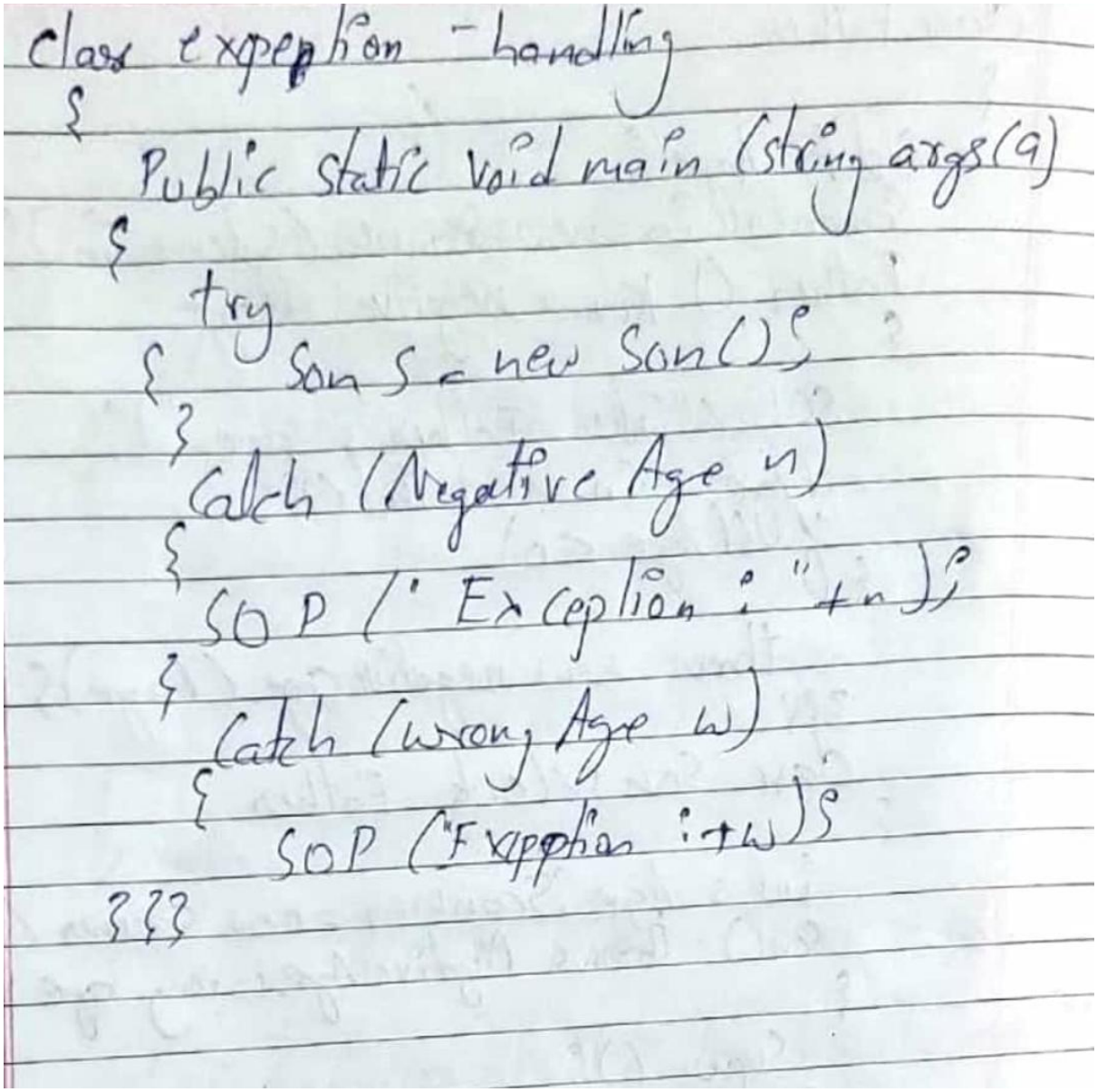
when the input age<0. In Son class, implement a constructor that cases both father and son’s

age and throws an exception if son’s age is >=father’s age

Program 8.

```
import java.util.*;
class WrongAge extends Exception
{
    int f,s;
WrongAge(int fAge, int sAge)
{
    f = fAge;
    s = sAge;
}
Public string to string()
{
  return "Entered correct ages of Father's age
  can't be less than/ eqlat to Son's age.";
}
{
class Negative Age extends Exception
{
    int x;
  Negative Age(int Age)
  {
    x = fage;
  {
    Public string to string()
  {
    return "Age cant be negative";
  }
}
```

```
class Father
{
  int fAge;
  Scanner in = new Scanner(System.in);
  Father() throws negative age
  {
    SOP("Enter Father's age");
    fAge = in.nextInt();
    if(fAge < 0)
    {
      throw new negativeAge(fage);
    }}}
class Son extends Father
{
  int sAge; Scanner = 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);
  }}}
```




```
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.
```

Q) 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 -9

```
class Thread implements Runnable
{
  Thread t;
  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();
  }
```

```
Public Void run()
{
   {for (;;)
      try
      {
         SOP ("CSE");
         Thread.sleep (2000);
      }
   Catch (InterruptedException ie)
      {
         SOP ("interrupted");
   }}}}
   Class Threads
{
   Public Static Void main(String args[])
{
   SOP ("Enter ctrl+C to stop.");
   Thread t1 = new Thread1();
   Thread t2 = new Thread2();
   }
{
```

```
Enter CTRL+C to stop
BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE
```

Q) 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

```
import java.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 ln3 = new label ("RESULT", Label.RIGHT);
  n1 = new TextField (12);
   n2 = new TextField (12);
    b = new Button ("Divide");
   add (ln1);
    add (n1);
    add (ln2);
    add (n2);
    add (b);
    add (lres); add (res);
     b. add action Listner (this);
 add window Listner (new Window Adapter ());
  }
  Public void action performed (Action Event ae)
{
```

```
if (age.getSource() == b)
{
    try {
int num1 = Integer.parseInt(n1.getText());
int num2 = Integer.parseInt(n2.getText());
 int num3 = num1 / num2;
nes.setText(String.valueOf(num3));
}
catch (NumberFoundException ne)
{
  JOptionPane.showMessageDialog(this,"Error",
  JOptionPane.ERROR_MESSAGE);
{
catch (ArithmeticException exception 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 WindowAdapter extends WindowAdapter
{ public void windowClosing(WindowEvent we)
{
    System.exit(0);
}}}
```

Output


NUMBER 1
th
NUMBER 2
0
DIVIDE
RESULT


ERROR
java.lang.NumberFormatException: For input string: "th"
OK