

```
package SEE;

import java.util.Scanner;

import CIE.Student;

public class Externals extends CIE.Student

{

    public int n = 5;

    public int see[] = new int[n];

    public int i;

    Scanner ss = new Scanner(System.in);

    public void acceptsee()

    {

        System.out.println("\n<----SEE MARKS DETAILS---->\n");

        System.out.println("Enter the marks in each of the "+n+" subjects :");

        for(i = 0; i < n; i++)

        {

            System.out.print("Enter the SEE marks in subject "+(i+1)+" : ");

            see[i] = ss.nextInt();

        }

    }

}
```

```
package CIE;

import java.util.Scanner;

public class Internals extends Student
{
    public int n = 5;
    public int cie[] = new int[n];
    public int i;
    Scanner ss = new Scanner(System.in);

    public void acceptcie()
    {
        System.out.println("\n<----CIE MARKS DETAILS---->\n");
        System.out.println("Enter the marks in each of the "+n+" subjects :");
        for(i = 0; i < n; i++)
        {
            System.out.print("Enter the CIE marks in subject "+(i+1)+": ");
            cie[i] = ss.nextInt();
        }
    }
}
```

```
package CIE;

import java.util.Scanner;

public class Student
{
    public String usn;
    public String name;
    public int sem;
    Scanner ss = new Scanner(System.in);

    public Student()
    {
        usn = "";
        name = "";
        sem = 0;
    }

    public void accept()
    {
        System.out.println("\n<-----ENTER PERSONAL DETAILS----->\n");
        System.out.println("Enter the student USN :");
        usn = ss.next();
        System.out.println("Enter the student NAME :");
        name = ss.next();
        System.out.println("Enter the student SEMESTER :");
        sem = ss.nextInt();
    }
}
```



```

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

class Totalmarks
{
    public static void main(String[] args)
    {
        int n;
        int tot[][];
        Scanner ss = new Scanner(System.in);
        System.out.println("Enter the number of students : ");
        n = ss.nextInt();
        tot = new int[n][5];
        CIEInternals ci[] = new CIEInternals[n];
        SEEExternals se[] = new SEEExternals[n];
        for(int i = 0; i < n; i++)
        {
            System.out.println("\n<-----DETAILS OF STUDENT "+(i+1)+"----->");
            ci[i] = new CIEInternals();
            se[i] = new SEEExternals();
            ci[i].accept();
            ci[i].acceptcie();
            se[i].acceptsee();
            for(int j = 0; j < 5; j++)
            {
                tot[i][j] = ci[i].cie[j] + (se[i].see[j] / 2);
            }
        }
    }
}

```

```
        }

    }

System.out.println("<----->");

for(int i = 0; i < n; i++)

{

    System.out.println("\nSTUDENT "+(i+1)+" FINAL MARKS OUT OF 100\n");

    for(int j = 0; j < 5; j++)

    {

        System.out.println("Marks in Course "+(j+1)+" : "+tot[i][j]);

    }

}

System.out.println("<----->\n");

}

}
```

```
package C15;  
import java.util.Scanner;  
public class Student  
{  
    public String usn;  
    public String name;  
    public int sem;  
    Scanner ss = new Scanner(System.in);
```

```
public Student()  
{
```

```
    usn = " ";  
    name = " ";  
    sem = 0;
```

```
    public void accept()  
{
```

```
        System.out.println("|\n--- ENTER PERSONAL  
DETAILS ---|\n");
```

```
        System.out.print("Enter the student USN");
```

```
        usn = ss.next();
```

```
        System.out.print("Enter the student  
NAME:");
```

```
        name = ss.next();
```

```
        System.out.print("Enter the student  
Semester |\n");
```

```
        sem = ss.nextInt();
```

```
}  
}
```

Date: / /

```
package CIE;
import java.util.Scanner;
public class Internals extends Student
{
    public int n = 5;
    public int cie[] = new int [n];
    public int i;
    Scanner ss = new Scanner (System.in);
    public void acceptie()
    {
        System.out.println ("In --- CIE MARK DETAILS ---> \n");
        System.out.println ("Enter the marks in each of the "+n+" subjects : ");
        for (i=0; i<n; i++)
        {
            System.out.println ("Enter the CIE mark in subject " +(i+1) + ":" );
            cie[i] = ss.nextInt();
        }
    }
}
```

```
System.out.println ("Enter the CIE mark in subject " +(i+1) + ":" );
cie[i] = ss.nextInt();
```

y
y
y

package SEE;

import java.util.Scanner;

import CIE.Student;

public class Enternals extends CIE.Student

{

public int n = 5;

public int arr[] = new int [n];

public int i;

Scanner ss = new Scanner(system.in);

public void accept()

System.out.println ("\\n<---SEE Marks
Detail-->\\n");

System.out.println ("Enter the marks
in each of the "+n+"
subjects : ");

for (i=0 ; i<n ; i++)

{

System.out.print ("enter the SEE mark
in subject "+(i+1)+" : ");

arr[i] = ss.nextInt();

y
y
y

y

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

class Totalmarks

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

```
int n;
```

```
int tot[][];
```

```
Scanner ss = new Scanner (System.in);
```

```
System.out.println ("Enter the no of students");
```

```
n = ss.nextInt();
```

```
tot = new int [n][5];
```

```
CIE.Externals ci[] = new CIE.Externals[n];
```

```
SEE.Externals se[] = new SEE.Externals[n];
```

```
for (int i=0; i<n; i++)
```

```
System.out.println ("\\n<-- - Detail of  
Student "+(i+1)+"-----");
```

```
ci[i] = new CIE.Externals();
```

```
se[i] = new SEE.Externals();
```

```
ci[i].accept();
```

```
ci[i].acceptcie();
```

```
se[i].acceptsee();
```

```
for (int j=0; j<5; j++)
```

```
tot [i][j] = ci[i].cie[j] + (se[i].sec[j]/2);
```

```
System.out.println("<----->");  
for (int i=0 ; i<n ; i++)  
{  
    System.out.println("IN STUDENT"+(i+1)+  
        "FINAL MARKS OUT OF 100\n");  
    for (int j=0 ; j<5 ; j++)  
    {  
        System.out.println("Marks in Course "+(j+1)  
            + " : "+tot[i][j]);  
    }  
    System.out.println("<----->");  
}
```

C:\Users\91944\OneDrive\Desktop\OOJLAB\Week_9>java Totalmarks

Enter the number of students :

2

<-----DETAILS OF STUDENT 1----->

<----ENTER PERSONAL DETAILS---->

Enter the student USN :

111

Enter the student NAME :

Ram

Enter the student SEMESTER :

3

<----CIE MARKS DETAILS---->

Enter the marks in each of the 5 subjects :

Enter the CIE marks in subject 1 : 45

Enter the CIE marks in subject 2 : 46

Enter the CIE marks in subject 3 : 47

Enter the CIE marks in subject 4 : 48

Enter the CIE marks in subject 5 : 49

<----SEE MARKS DETAILS---->

Enter the marks in each of the 5 subjects :

Enter the SEE marks in subject 1 : 95

Enter the SEE marks in subject 2 : 96

Enter the SEE marks in subject 3 : 97

Enter the SEE marks in subject 4 : 98

Enter the SEE marks in subject 5 : 99

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

<----->

C:\Users\91944\OneDrive\Desktop\OOJLAB\Week_9>

Lab 8:

```
1 import java.util.*;
2 class Wrongage extends Exception{
3     private String detail;
4     Wrongage(String s)
5     {
6         detail=s;
7     }
8     public String toString(){
9         return ("invalid age exception"+detail);
10    }
11 }
12 class Father{
13     int age;
14     Father(int a){
15         age=a;
16     }
17 }
18 class Son extends Father{
19     int age1;
20     Son(int a,int b)
21     {
22         super(a);
23         age1=b;
24     }
25 }
26 public class abc{
27     public static void main(String[] args){
28         Scanner sc=new Scanner(System.in);
29         System.out.println("enter the age of father and son respectively");
```

```
29         System.out.println("enter the age of father and son respectively");
30         int m=sc.nextInt();
31         int n=sc.nextInt();
32         try{
33             if(m<0)
34                 throw new Wrongage("negative");
35             else if(n>m)
36                 throw new Wrongage("illogical");
37             else
38             {
39                 Son ob=new Son(m,n);
40                 System.out.println("father's age:"+ob.age+"son's age"+ob.age1);
41             }
42         }
43         catch (Wrongage e){
44             System.out.println(e);
45         }
46     }
47 }
```

The screenshot shows a Java code editor with the following interface elements:

- A toolbar at the top with a "Execute" button and other icons.
- A "Result" section below the toolbar.
- The "Result" section displays the output of the program execution.

The output in the "Result" section is as follows:

```
Result
compiled and executed in 54.49 sec(s)

enter the age of father and son respectively
10
25
invalid age exceptionillogical
```

```
import java.util.*;  
class Wrongage extends Exception{  
    private String detail;  
    Wrongage (String s){  
        detail = s;  
    }  
    public String toString() {  
        return ("invalid age exception "+detail);  
    }  
}  
class Father {  
    int age;  
    Father (int a) {  
        age = a;  
    }  
}  
class Son extends Father {  
    int age1;  
    Son (int a, int b) {  
        super (a);  
        age1 = b;  
    }  
}
```

Notes

```
public class abc {  
    public static void main (String [] args)  
    {  
        Scanner sc = new Scanner (System .in);  
        System.out.println ("enter the age of  
        father & son respectively.");  
        int m = sc.nextInt();  
        int n = sc.nextInt();  
        try {  
            if (m < 0)  
                throw new Wrongage ("negative");  
            else if (n >= m)  
                throw new Wrongage ("illegal");  
            else {  
                Son ob = new Son (m, n);  
                System.out.println ("father's age : " + ob.  
                age + " son's age : " + ob.age);  
            }  
        } catch (Wrongage e) {  
            System.out.println (e);  
        }  
    }  
}
```

LAB 7

```
Project Name: generic
 1+ class Gen<T> {
 2
 3   T ob; // declare an object of type T
 4
 5   // Pass the constructor a reference to
 6   // an object of type T.
 7
 8   Gen(T o) {
 9
10     ob = o;
11
12   }
13
14   // Return ob.
15   T getob() {
16
17     return ob;
18
19   }
20
21 }
22
23 // Show type of T.
24
25 void showType() {
26
27   System.out.println("Type of T is " +
28
29   ob.getClass().getName());
30
31 }
32
33 }
34
35 public class GenDemo {
36
37   public static void main(String args[]) {
38
39     Gen<Integer>iob;
40
41     iob = new Gen<Integer>(88);
42
43     iob.showType();
```

```
43   iob.showType();
44
45   // Get the value in iob. Notice that
46
47   // no cast is needed.
48
49   int v = iob.getob();
50
51   System.out.println("value: " + v);
52
53   System.out.println();
54
55   // Create a Gen object for Strings.
56
57   Gen<String>strob = new
58   Gen<String>("Generics Test");
59
60   // Show the type of data used by strob.
61
62   strob.showType();
63
64   // Get the value of strob. Again, notice
65
66   // that no cast is needed.
67
68   String str = strob.getob();
69
70   System.out.println("value: " + str);
71
72 }
73
74 }
75
76 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4 Interactive CommandLine Arguments

Result

compiled and executed in 0.984 sec(s)

```
Type of T is java.lang.Integer  
value: 88  
  
Type of T is java.lang.String  
value: Generics Test
```

Lab 1

generic program :

```
class Gen<T> {
    T ob;
    Gen'(T o) {
        ob = o;
    }
    T getOb() {
        return ob;
    }
    void showType() {
        System.out.println("Type of T is " +
            ob.getClass().getName());
    }
}
public class GenDemo {
    public static void main (String [] args)
    {
        Gen<Integer> ob;
```

```
iob = new Gen<Integer>(88);
```

```
iob = iob.showType();
```

```
int v = iob.getob();
```

```
System.out.println("Value :" + v);
```

```
System.out.println();
```

```
Gen<String> stob = new
```

```
Gen<String> ("Generic Test");
```

```
stob.showType();
```

```
String str = stob.getob();
```

```
System.out.println("Value :" + str);
```

```
}
```

```

1  class thread1 implements Runnable
2  {
3      Thread t;
4      thread1()
5      {
6          t = new Thread(this,"thread1");
7          t.start();
8      }
9
10     public void run()
11     {
12         for(;;)
13         {
14             try
15             {
16                 System.out.println("BMS College Of Engineering");
17                 Thread.sleep(10000);
18             }
19             catch(InterruptedException ie)
20             {
21                 System.out.println("Interrupted");
22             }
23         }
24     }
25 }
26
27 class thread2 implements Runnable
28 {
29     Thread t2;
30     thread2()
31     {
32         t2 = new Thread(this,"thread2");
33         t2.start();
34     }
35 }
36

```

```

34
35 }
36
37     public void run()
38     {   for(;;)
39     {
40         try
41         {
42             System.out.println("CSE");
43             Thread.sleep(2000);
44         }
45         catch(InterruptedException ie)
46         {
47             System.out.println("Interrupted");
48         }
49     }
50 }
51 }
52
53 public class threadmain
54 {
55     public static void main(String args[])
56     {
57         System.out.println("Enter CONTROL+C to stop");
58         thread1 t1 = new thread1();
59         thread2 t2 = new thread2();
60     }
61 }
62 }
```

Execute ... EJ

Result
compiled and executed in 120.593 sec(s)

```

Enter CONTROL+C to stop
BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College Of Engineering
CSE

```

Tuesday

DECEMBER 2019

class Thread implements Runnable

{

Thread t;

thread() {

t = new Thread(this, "the
t.start();

}

public void run()

{

for(;;) {

try {

System.out.println("BMS coll
Thread.sleep(10000);

}

catch (InterruptedException ie)

System.out.println("interrupted");

} }

}

Notes

class Thread2 implements Runnable

{ Thread t2;

Thread2() {

t2 = new Thread(this, "Thread2");

t2.start();

}

public void run() {

for(;;) {

try {

System.out.println("CSE");

Thread.sleep(2000);

} catch (InterruptedException e) {

System.out.println("Interrupted");

}}

public class Threadmain {

public static void main(String[] args) {

System.out.println("Enter Ctrl+C to stop");

Thread1 t1 = new Thread1();

Thread2 t2 = new Thread2();

}}}

```
class CarService {  
  
    public static void main(String args[]){  
  
        Car_Queue q= new Car_Queue();  
  
        new Car_Owner(q);  
        new Car_Mechanic(q);  
  
        System.out.println("Press Ctrl+C to stop.");  
  
    }  
}  
  
class Car_Queue{  
  
    int n;  
  
    boolean valueSet = false;  
  
    synchronized int get()  
    {  
  
        while(!valueSet)  
  
        try{  
  
            wait();  
        }  
    }  
}
```

```
}

catch (InterruptedException e)
{
    System.out.println("InterruptedException caught");

}

System.out.println("Service Provided:"+n);

valueSet = false;

notify();

return n;

}

synchronized void put(int n)

{

while(valueSet)

try{

wait();

}

catch(InterruptedException e)
```

```
{  
  
    System.out.println("InterruptedException caught");  
  
}  
  
this.n=n;  
  
valueSet = true;  
  
System.out.println("Order placed:"+n);  
  
notify();
```

```
}  
}  
  
class Car_Mechanic implements Runnable  
{
```

```
Car_Queue q;  
  
Car_Mechanic(Car_Queue q)  
{  
  
    this.q=q;  
  
    new Thread(this,"Car_Mechanic").start();  
}
```

```
public void run(){

    int i=0;

    while(true){

        q.put(i++);

    }

}

}

}

class Car_Owner implements Runnable {

    Car_Queue q;

    Car_Owner(Car_Queue q)

    {

        this.q=q;

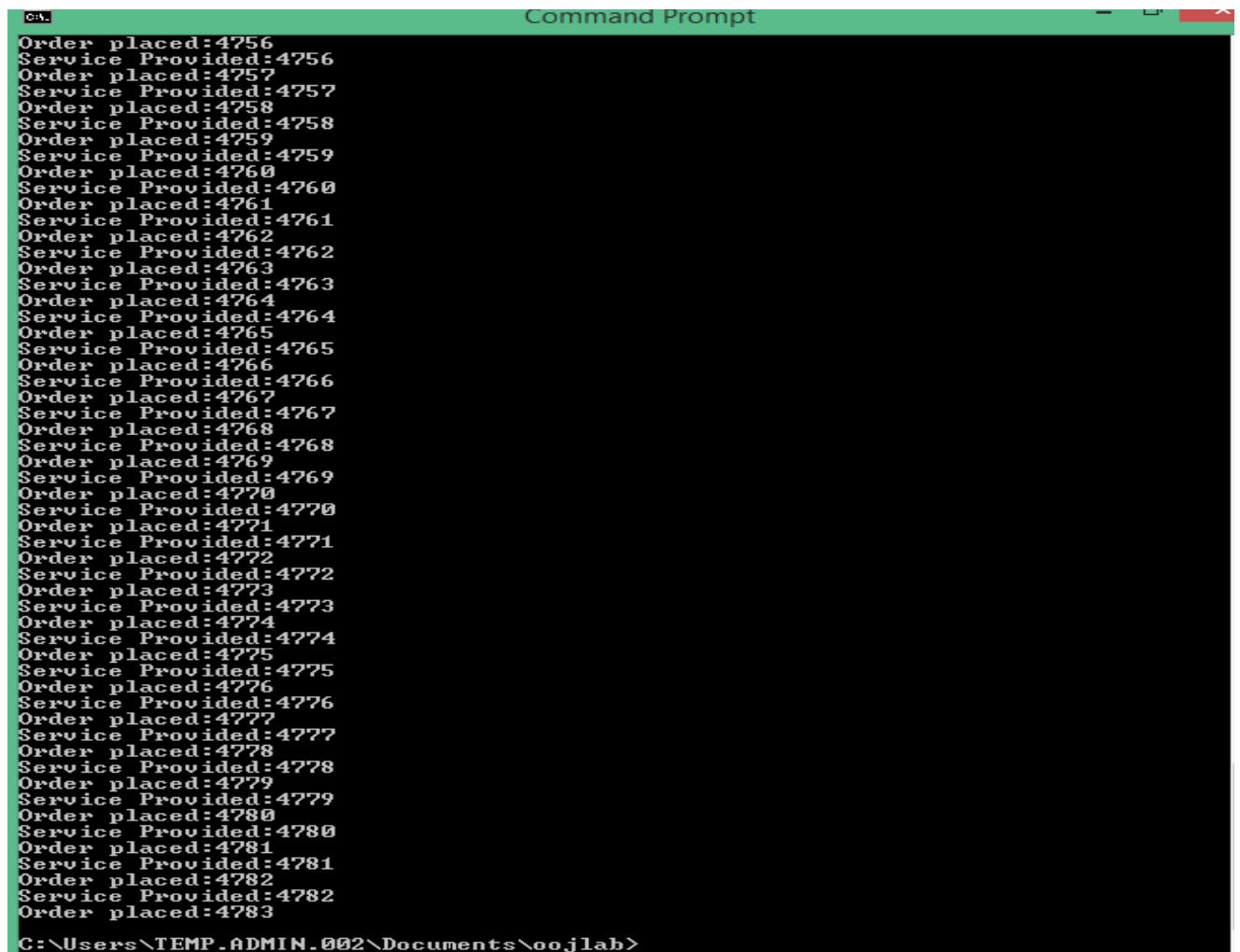
        new Thread(this,"Car_Owner").start();

    }

    public void run()

}
```

```
{  
  
while(true)  
{  
  
    q.get();  
}  
  
})
```



The screenshot shows a Command Prompt window with a green title bar labeled "Command Prompt". The window contains a large block of text representing log entries. The entries are in pairs, where each pair consists of "Order placed:" followed by a number and "Service Provided:" followed by the same number. The numbers range from 4756 to 4783. The text is white on a black background. At the bottom of the window, the path "C:\Users\TEMP.ADMIN.002\Documents\oopjlab>" is visible.

```
Order placed:4756  
Service Provided:4756  
Order placed:4757  
Service Provided:4757  
Order placed:4758  
Service Provided:4758  
Order placed:4759  
Service Provided:4759  
Order placed:4760  
Service Provided:4760  
Order placed:4761  
Service Provided:4761  
Order placed:4762  
Service Provided:4762  
Order placed:4763  
Service Provided:4763  
Order placed:4764  
Service Provided:4764  
Order placed:4765  
Service Provided:4765  
Order placed:4766  
Service Provided:4766  
Order placed:4767  
Service Provided:4767  
Order placed:4768  
Service Provided:4768  
Order placed:4769  
Service Provided:4769  
Order placed:4770  
Service Provided:4770  
Order placed:4771  
Service Provided:4771  
Order placed:4772  
Service Provided:4772  
Order placed:4773  
Service Provided:4773  
Order placed:4774  
Service Provided:4774  
Order placed:4775  
Service Provided:4775  
Order placed:4776  
Service Provided:4776  
Order placed:4777  
Service Provided:4777  
Order placed:4778  
Service Provided:4778  
Order placed:4779  
Service Provided:4779  
Order placed:4780  
Service Provided:4780  
Order placed:4781  
Service Provided:4781  
Order placed:4782  
Service Provided:4782  
Order placed:4783  
Service Provided:4783  
C:\Users\TEMP.ADMIN.002\Documents\oopjlab>
```

```
class CarService {
    public static void main (String args[])
    {
        CarQueue q = new CarQueue();
        new Car_Owner(q);
        new Car_Mechanic(q);
        System.out.println ("press Ctrl+C to stop");
    }
}

class CarQueue {
    int n;
    boolean valueSet = false;

    synchronized int get ()
    {
        while (!valueSet)
            try {
                wait();
            } catch (InterruptedException e)
        {
            System.out.println ("Interrupted Exception caught");
            System.out.println ("Service provided : "+n);
            valueSet = false;
            notify();
            return n;
        }
    }
}
```

```
synchronized void put (int n)
{
    while (ValueSet)
        try {
            wait();
        } catch (InterruptException e)
    {
        System.out.println ("InterruptedException
                            caught");
    }
    this.n = n;
    ValueSet = true;
    System.out.println ("ordered placed:" + n);
    notify();
}
```

class Car_Mechanic implements Runnable

```
{
    CarQueue q;
    Car_Mechanic (CarQueue q)
    {
        this.q = q;
        new Thread (this, "Car_Mechanic").start();
    }
}
```

public void run()

```
    int i = 0;
    while (true) {
        q.put (i++);
    }
}
```

```
    }  
    public void run()  
{
```

```
        while (true)
```

```
{
```

```
    q.get();
```

```
}
```

```
class Table
{
    void printTable(int n)
    {
        synchronized(this)
        {
            for(int i=1;i<=5;i++)
            {
                System.out.println(+n+"*"+i+"="++(n*i));
            }
            Thread.sleep(400);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

}

}

}

}

}

class Mythread1 extends Thread

{

Table t;

Mythread1(Table t)

{

this.t=t;

}

public void run()

{

t.printTable(5);

```
    }  
}  
  
}
```

```
class Mythread2 extends Thread
```

```
{
```

```
Table t;
```

```
Mythread2(Table t)
```

```
{
```

```
this.t=t;
```

```
}
```

```
public void run()
```

```
{
```

```
t.printTable(10);
```

```
}
```

```
}
```

```
class Use

{

public static void main(String args[])

{

Table obj = new Table();

Mythread1 th1 = new Mythread1(obj);

Mythread2 th2 = new Mythread2(obj);

th1.start();

th2.start();

}

}
```



Command Prompt

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\TEMP.ADMIN.002>cd documents

C:\Users\TEMP.ADMIN.002\Documents>cd oojlab

C:\Users\TEMP.ADMIN.002\Documents\oojlab>javac Use.java

C:\Users\TEMP.ADMIN.002\Documents\oojlab>java Use
5*1=5
5*2=10
5*3=15
5*4=20
5*5=25
10*1=10
10*2=20
10*3=30
10*4=40
10*5=50

C:\Users\TEMP.ADMIN.002\Documents\oojlab>_
```

class Table

{
void printTable (int n)
{

synchronized (this)

{ for (int i = 1; i <= 5; i++)

System.out.println (" " + n + " * " + i + " = " + (n * i));

try

{

Thread.sleep (400);

} catch (Exception e)

System.out.println(e);

}
}
}
}

class Mythread1 extends Thread

{ Table t;

Mythread1 (Table t)

{

this.t = t;

}

public void run()

{ t.printTable (5);

}

class Mythread2 extends Thread

{

Table t;

{ Mythread2 (Table t)

 this.t = t;

}
y

class Use

{

{ public static void main (String args)

 Table Obj = new Table();

 Mythread1 th1 = new Mythread1 (Obj);

 Mythread2 th2 = new Mythread2 (Obj);

 th1.start();

 th2.start();

}
y
y

Lab Program 10

Q) Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 & Num2. The division of Num1 & 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 Number Format Exception if Num2 is zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.

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

public class integerdivision extends Frame implements ActionListener {
    JTextField n1, n2, res;
    Label l1, l2, lres;
    Button b;

    public integerdivision() {
        setLayout(new FlowLayout());
        l1 = new Label("Number 1", Label.RIGHT);
        l2 = new Label("Number 2", Label.RIGHT);
        lres = new Label("RESULT", Label.RIGHT);
        n1 = new TextField(12);
        n2 = new TextField(8);
        res = new TextField(10);
        b = new Button("DIVIDE");
        add(l1);
        add(n1);
        add(l2);
        add(n2);
        add(b);
        add(lres);
        add(res);
    }

    public void actionPerformed(ActionEvent e) {
        try {
            int a = Integer.parseInt(n1.getText());
            int b = Integer.parseInt(n2.getText());
            res.setText(Integer.toString(a / b));
        } catch (Exception e1) {
            JOptionPane.showMessageDialog(this, "Error");
        }
    }
}
```

~~to add~~

b. addActionListener(this);

addWindowListener(new WindowAdapter() {

g.

public void actionPerformed(ActionEvent ae) {

if (ae.getSource() == b) {

try {

int num1 = Integer.parseInt(n1.getText());

int num2 = Integer.parseInt(n2.getText());

int num3 = num1 / num2;

res.setText(String.valueOf(num3));

} catch (NumberFormatException ne) {

JOptionPane.showMessageDialog(this, ne, "ERROR", JOptionPane.ERROR_MESSAGE);

g

catch (ArithmaticException a) {

JOptionPane.showMessageDialog(this, a, "ERROR", JOptionPane.ERROR_MESSAGE);

g

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);

g

class WindowAdapter1 extends WindowAdapter {

public void windowClosing(WindowEvent we) {

System.exit(0);

g

```
integerdivision - Notepad
File Edit Format View Help
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(8);
        res=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 WindowAdapter1());
    }
    public void actionPerformed(ActionEvent ae)
    {
        if(ae.getSource()==b)
        < Ln 17, Col 11 100% Windows (CRLF) UTF-8
        >
```

integerdivision - Notepad

File Edit Format View Help

```
import java.awt.*;
import java.awt.event.*;
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(8);
        res=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 WindowAdapter1());
    }
    public void actionPerformed(ActionEvent ee)
    {
        String s1=n1.getText();
        String s2=n2.getText();
        int i1=Integer.parseInt(s1);
        int i2=Integer.parseInt(s2);
        res.setText(i1/i2+"");
    }
}
```

Ln 17, Col 11 100% Windows (CRLF) UTF-8

Type here to search

21:56
16-01-2021

```
integerdivision - Notepad
File Edit Format View Help
b.addActionListener(this);
addWindowListener(new WindowAdapter1());
}
public void actionPerformed(ActionEvent ae)
{
if(ae.getSource()==b)
{
try{
int num1=Integer.parseInt(n1.getText());
int num2=Integer.parseInt(n2.getText());
int num3=num1/num2;
res.setText(String.valueOf(num3));
}catch(NumberFormatException ne ){
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");
}

```

Ln 17, Col 11 100% Windows (CRLF) UTF-8

Type here to search

21:56 16-01-2021

 INTEGER DIVISION OF TWO NUMBERS

- □ ×

NUMBER 1 NUMBER 2 RESULT 

Type here to search

21:53
16-01-2021 ENG

 INTEGER DIVISION OF TWO NUMBERS

- □ ×

NUMBER 1 NUMBER 2 RESULT 

Type here to search

21:54
16-01-2021 ENG

INTEGER DIVISION OF TWO NUMBERS

NUMBER 1 NUMBER 2 DIVIDE RESULT

