

---

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.

## Lab program 6 (week 9):-

② Student.java :- (..\\packs\\CIE\\Student.java)

package CIE;

import java.util.Scanner;

public class Student

    public ~~int~~ String usn, name;

    public int sem;

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

    public void getData()

}

        System.out.println("Name: ");

        name=ss.next();

        System.out.println("USN: ");

        usn=ss.next();

        System.out.println("Semester: ");

        sem=ss.nextInt();

}

}

Internals.java :- (..\\packs\\CIE\\Internals.java)

package CIE;

import java.util.Scanner;

public class Internals extends Student

{

    public int ciemarks[] = new int[5];

    public void getmarks()

{

        getData();

        System.out.print("Enter the CIE marks of the student: ");

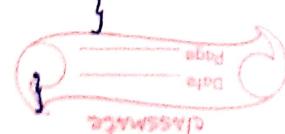
        for (int i=0; i<5; i++)

{

            System.out.print("Subject "+(i+1)+" : ");

            ciemarks[i] = ss.nextInt();

}



```
public void display()
{
    System.out.println("Name: " + this.name);
    System.out.println("USN: " + this.usn);
    System.out.println("Semester: " + this.sem);
}
```

External.java:-(..\\pack<sup>3</sup>\\SEE\\External.java)

```
package SEE;
import java.util.Scanner;
public class External extends CIE.Student
{
    public int semarks[] = new int[5];
    Scanner ss = new Scanner(System.in);
    public void getmarks()
    {
        System.out.println("Enter the SEE marks of the student:");
        for(int int i=0; i<5; i++)
        {
            System.out.print("Subject " + (i+1) + ": ");
            semarks[i] = ss.nextInt();
        }
    }
}
```

Main.java:- (..\\packs\\ Main.java)

import java.util.Scanner;

import CIE.\*;

import SEE.\*;

class Main

{

public static void main (String args[])

Scanner ss=new Scanner (System.in);

int n;

int finalmarks []= new int [5];

System.out.println("Enter the number of Students : ");

n=ss.nextInt();

Internals studcie []= new Internals [n];

Externals studsee []= new Externals [n];

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

{

System.out.println ("\n x x x x x x x x x x x x \n");

studcie [i]= new Internals();

studsee [i]= new Externals();

studcie [i].getmarks();

studsee [i].getmarks();

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

finalmarks [j]= studcie [i].ciemarks [j] +  
(studsee [i].seemarks [j]/2);

}

System.out.println ("\n x x x x x x x x x x x x \n");

System.out.println ("The Student and their final marks are: \n");

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

{

studcie [i].display();

~~NOTE~~



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

```
    System.out.println("Subject " + (j+1) + ":" + finalmarks[j]);
```

```
    System.out.println("\nxx x x x x x x x x x\n");
```

```
}
```

**Enter the number of Students:**

2

\*\*\*\*\*

**Name:**

Khushil

**USN:**

1BM19CS072

**Semester:**

3

**Enter the CIE marks of the Student:**

Subject 1 :34

Subject 2 :50

Subject 3 :48

Subject 4 :46

Subject 5 :40

**Enter the SEE marks of the Student:**

Subject 1 :89

Subject 2 :

90

Subject 3 :92

Subject 4 :76

Subject 5 :78

\*\*\*\*\*

**Name:**

Sharat

**USN:**

1BM19CS072

**Semester:**

3

**Enter the CIE marks of the Student:**

Subject 1 :50

Subject 2 :48

Subject 3 :45

Subject 4 :43

Subject 5 :35

**Enter the SEE marks of the Student:**

Subject 1 :76

Subject 2 :78

Subject 3 :98

Subject 4 :80

Subject 5 :90

\*\*\*\*\*

The Students and their finals marks are:

Name: Khushil  
USN: 1BM19CS072  
Semester: 3  
Subject 1 :78  
Subject 2 :95  
Subject 3 :94  
Subject 4 :84  
Subject 5 :79

\*\*\*\*\*

Name: Sharat  
USN: 1BM19CS072  
Semester: 3  
Subject 1 :88  
Subject 2 :87  
Subject 3 :94  
Subject 4 :83  
Subject 5 :80

\*\*\*\*\*

**Write a program to demonstrate generics with multiple object parameters.**

## Lab program 7 :-

```
class famous<T, S>
{
    T mem1;
    S mem2;
    famous(T ob1, S ob2)
    {

```

```
        this.mem1 = ob1;
```

```
        this.mem2 = ob2;
    }
```

```
    public void displayItems()
    {
```

```
        System.out.println("The value of " + this.mem1 + " is : " + this.mem2);
    }
```

```
}
```

```
class FamousMain
```

```
{
```

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

```
        famous<String, String> std1 = new famous<String, String>("Name", "Elon musk");
```

```
        famous<String, Double> std2 = new famous<String, Double>
```

```
( "Networth in Billions ", 127.9 );
```

```
        famous<String, Integer> std3 = new famous<String, Integer>("Age", 49);
```

```
        std1.displayItems();
```

```
        std2.displayItems();
```

```
        std3.displayItems();
```

```
}
```

```
}
```

```
The value of Name is: Elon Musk  
The value of Net Worth in Billions is: 127.9  
The value of Age is: 49  
PS D:\Java Programs\Week 10> 
```

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.

## Lab program 8 :-

```
import java.util.*;  
class AgeException extends Exception  
{  
    public String toString()  
    {  
        return "Age of the Father cannot be lesser than the age of the  
        Son";  
    }  
}
```

~~class~~  
class Father

```
{  
    public Scanner ss = new Scanner (System.in);  
    int f-age;
```

Father()

{

```
    System.out.println ("Enter the age of the Father: ");
```

```
    f-age = ss.nextInt();
```

}

}

class Son extends Father

{

```
    int s-age;
```

Son()

{

```
    System.out.println ("Enter the Age of the Son: ");
```

```
    s-age = ss.nextInt();
```

}

}

class FamilyMain

{

    static void WrongAge (int fage, int sage) throws AgeException

{

        if (fage < sage)

{

            throw new AgeException();

}

    System.out.println ("There ~~were~~ were no other problems  
        arised during execution");

    System.out.println ("The Age of the Father is: " + fage);

} System.out.println ("The Age of the Son is: " + sage);

public static void main (String args [3])

{

    Son sn = new Son();

    try

    {

        WrongAge (sn.fage, sn.sage);

}

    catch (AgeException e)

{

        System.out.println ("An ~~Age~~ exception popped up: " + e);

}

}

}

```
Enter the Age of the Father:  
34  
Enter the Age of the Son:  
12  
There were no other problems arised during execution  
The age of the Father is: 34  
The age of the Son is: 12  
PS D:\Java Programs\Week 10> cd 'd:\Java Programs\Week 10'; & 'c:\Users\khush\.vscode\exten  
JDK\jdk-11.0.8.10-hotspot\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,susp  
Storage\2b4e8dbe371b3dd5a4fa074dc92557af\redhat.java\jdt_ws\Week 10_d36af370\bin' 'Family  
Enter the Age of the Father:  
12  
Enter the Age of the Son:  
34  
An Exception popped up: Age of the Father Cannot be lesser than the age of the Son  
PS D:\Java Programs\Week 10> []
```

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 program 9 :-

class NewThread1 implements Runnable

{

    Thread t;

    String ~~sth~~ name;

    long time;

    NewThread1(String stname, long thtime)

{

        time = thtime;

        name = stname;

        t = new Thread(this, name);

        t.start();

}

    public void run()

{

        try

{

            for (int n=10; n>0; n--)

{

                System.out.println(t.getName());

                Thread.sleep(time);

}

}

    catch (InterruptedException ie)

{

        System.out.println("Child Thread Interrupted");

}

        System.out.println("Child Thread Quitting....");

}

}

```
class ThreadMain
{
    public static void main(String args[])
    {
        NewThreads1 n1 = new NewThreads1("BMS College of Engineering", 1000);
        NewThreads1 n2 = new NewThreads1("CSE", 2000);
    }
}
```

BMS College of Engineering.

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering.

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering.

Child Thread Quitting.....

BMS College of Engineering.

Child Thread Quitting.....

---

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 ArithmeticException. Display the exception in a message dialog box.

## Lab program 10 :- (Week 12)

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

class NumException extends Exception {
    public String toString() {
        return "There is an Arithmetic Exception.";
    }
}
```

```
class FormatException extends Exception {
    public String toString() {
        return "There is an Format Exception.";
    }
}
```

```
public class dividenum extends frame implements ActionListener {
```

```
    JTextField num1, num2;
```

```
    Button div;
```

```
    double result;
```

```
    String msg = "The result is: 0.0";
```

```
    public dividenum()
```

```
{
```

```
    setLayout(new FlowLayout());
```

```
    Label num1n = new Label("Numerator: ", Label.RIGHT);
```

```
    Label num2n = new Label("Denominator: ", Label.RIGHT);
```

```
    Button div = new Button("Divide");
```

```
    num1 = new JTextField(5);
```

```
    num2 = new JTextField(5);
```

```
    add(num1n);
```

```
    add(num1);
```

```
    add(num2n);
```

```
    add(num2);
```

```
    add(div);
```

```
num1.addActionListener(this);
num2.addActionListener(this);
div.addActionListener(this);
```

```
addWindowListener(new WindowAdapter()
{
```

```
public void windowClosing(WindowEvent we)
{
```

```
    System.exit(0);
}
```

```
} );
```

```
public boolean isDouble (double num)
{
```

```
    double dec;
```

```
    dec = num - (int) num;
```

```
    if (dec == 0.0)
```

```
        return true;
```

```
    else
```

```
        return false;
    }
```

```
public double divide (double a, double b) throws -
```

```
- NumException , FormatException
```

```
{
```

```
if (b == 0.0)
```

```
{
```

```
    throw new NumException();
```

```
}
```

```
else if (isDouble(a) || isDouble(b))
```

```
{
```

```
    throw new FormatException();
```

```
}
```

```
    return (double) a/b;
```

```
}
```

```
public void actionPerformed (ActionEvent ae)
```

```
{
```

```
    double a, b;
```

```
    a = Double.parseDouble (num1.getText ());
```

```
    b = Double.parseDouble (num2.getText ());
```

```
    try {
```

```
        result = divide (a, b);
```

```
        msg = ("The result is: " + result);
```

```
}
```

```
    catch (NumberFormatException ne)
```

```
{
```

```
        msg = ne.toString ();
```

```
}
```

```
    catch (FormatException fe)
```

```
{
```

```
        msg = fe.toString ();
```

```
}
```

```
    repaint ();
```

```
}
```

```
public void paint (Graphics g)
```

```
{ ResultDialog d = new ResultDialog (this, "Result")
```

~~```
d.setVisible (true);
```~~

```
}
```

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

```
{
```

```
    dividenum appwin = new dividenum ();
```

```
    appwin.setSize (new Dimension (350, 300));
```

```
    appwin.setTitle ("Divide Two Numbers");
```

```
    appwin.setVisible (true);
```

```
}
```

```
class ResultDialog extends Dialog implements ActionListener {  
    divide nums pt;
```

```
    ResultDialog (Frame parent, String title)  
    {
```

```
        Super (parent, title, false);
```

```
        pt = (divide nums) parent;
```

```
        setLayout (new Flow Layout());
```

```
        setSize (250,100);
```

```
        Button b = new Button ("OK");
```

```
        add (new Label (pt.nsg));
```

~~```
        b.addActionListener (this);
```~~

```
        add (b);
```

```
        addWindowListener (new WindowAdapter ()) {
```

```
            public void windowClosing (WindowEvent we) {
```

```
                dispose();
```

```
            }
```

```
        };
```

```
}
```

```
        public void actionPerformed (ActionEvent ae)
```

```
{
```

```
            dispose();
```

```
}
```

```
}
```



## Divide Two Numbers



Numerator:

12

Denominator:

3

Divide



## Result



The result is: 4.0

OK



## Divide Two Numbers

-



Numerator:

Denominator:

Divide



Result



There is an Arithmetic Exception.

OK



## Divide Two Numbers



Numerator:

Denominator:

Divide



### Result



There is an Formal Exception.

OK