

## //LAB 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.

```
package CIEp;

public class Student{
    public int usn;
    public String name;
    public int sem;
    public Student(int usn,String name,int sem){
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}

package CIEp;

public class Internals extends Student{
    public int[] cieMarks = new int[5];
    public Internals(int usn,String name,int sem,int[] cieMarks){
        super(usn,name,sem);
        this.cieMarks = cieMarks;
    }
}

package SEEp;
```

```
import CIEp.*;
```

```
public class Externals extends Student{  
    public int[] seeMarks = new int[5];  
    public Externals(int usn,String name,int sem,int[] seeMarks){  
        super(usn,name,sem);  
        this.seeMarks = seeMarks;  
    }  
}
```

```
import CIEp.*;
```

```
import SEEp.*;
```

```
import java.util.*;
```

```
public class Main7{  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        Externals[] e = new Externals[2];  
        Internals[] in = new Internals[2];  
        for(int i=0;i<2;i++){  
            int usn1 = input.nextInt();  
            String name1 = input.next();  
            int sem1 = input.nextInt();  
            int[][] cie = new int[2][5];  
            int[][] see = new int[2][5];  
            for(int j=0;j<5;j++){  
                cie[i][j] = input.nextInt();  
            }  
            for(int j=0;j<5;j++){  
                see[i][j] = input.nextInt();  
            }  
            e[i] = new Externals(usn1,name1,sem1,see[i]);  
        }  
    }  
}
```

```
in[i] = new Internals(usn1,name1,sem1,cie[i]);  
int total = 0;  
System.out.println("Name: "+e[i].name);  
System.out.println("USN: "+e[i].usn);  
System.out.println("sem: "+e[i].sem);  
for(int j=0;j<5;j++){  
    total = e[i].seeMarks[j]+in[i].cieMarks[j];  
    System.out.print("Final marks: "+total+" ");  
}  
System.out.println();  
}  
}  
}
```

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

(Ans Program - 6)

In the CIE package (Student class)

```
package cie;
public class student {
    public int usn;
    public String name;
    public int sem;
    public student (int usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}
```

&

// internals class

```
package cie;
public class internals extends student {
    public int[] cie marks = new int[5];
    public internals (int usn, String name,
        int sem, int[] cie marks) {
        super (usn, name, sem);
        this.cie marks = cie marks;
    }
}
```

SEE package

(external class)

```
package see;
import cie.*;
public class externals extends student {
    int[] see marks = new int[5];
    public externals (int usn, String name, int sem,
        int[] see marks) {
        super (usn, name, sem);
        this.see marks = see marks;
    }
}
```

In Default package  
import sec.\*;  
import uc.\*;  
class main {

public static void main (String[] args) {

int usm = 141;

String name = "Samarth";

int sem = 3;

int[] uc = { 50, 64, 48, 84, 54 };

int[] sec = { 2, 26, 40, 38, 10 };

Internals in = new Internals (usm, name, sem, uc);

Externals ex = new Externals (usm, name, sem, sec);

System.out.println("Name: " + in.name);

System.out.println("USM: " + ex.usm);

System.out.println("Sem: " + in.sem);

int total = 0;

for (int i = 0; i < 5; i++) {

total = in.ucMarks[i] + ex.secMarks[i];

System.out.print("Final marks: " + total + " ");

}

System.out.println();

}

```
Command Prompt java main7
C:\Users\Samarth\Documents\pac2>javac Main7.java
C:\Users\Samarth\Documents\pac2>java Main7
Enter the Number of students : 10
Enter the details of the student 1:
Enter usn of the student : 141
Enter name of the student : SAM
Enter semester of the student : 3
Enter the CIE marks :
Enter marks of the course 1: 12
Enter marks of the course 2: 13
Enter marks of the course 3: 14
Enter the SEE marks :
Enter the SEE marks of the course 1: 60
Enter the SEE marks of the course 2: 45
Enter the SEE marks of the course 3: 67
Enter the details of the student 2:
Enter usn of the student : 120
Enter name of the student : shivanshu
Enter semester of the student : 3
Enter the CIE marks :
Enter marks of the course 1: 15
Enter marks of the course 2: 20
Enter marks of the course 3: 38
Enter the SEE marks :
Enter the SEE marks of the course 1: 89
Enter the SEE marks of the course 2: 78
Enter the SEE marks of the course 3: 90
```

## //LAB 7

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

generics

```
import java.util.*;
```

```
class Genrics<T>{
```

```
    T var1;
```

```
    void Genirics(T gvar){
```

```
        var1 = gvar;
```

```
    }
```

```
    T Gdisplay(){
```

```
        return var1;
```

```
    }
```

```
}
```

```
public class App {  
    public static void main(String[] args) throws Exception {  
        System.out.println("Hello, World!");  
  
        Scanner Minp = new Scanner(System.in);  
  
        Genrics<Integer> Rollno= new Genrics<Integer>();  
        Genrics<String> Name = new Genrics<String>();  
  
        System.out.println("Enter Name of Student");  
        String Sname = Minp.nextLine();  
        Name.Genirics(Sname);  
  
        System.out.println("Enter USN of Student");  
        int Sno = Minp.nextInt();  
        Rollno.Genirics(Sno);  
  
        System.out.println("The student details are :");  
        System.out.println("Name : "+ Name.Gdisplay());  
        System.out.println("USN : "+ Rollno.Gdisplay());  
  
        Minp.close();  
    }  
}
```



### Program 7:

```
import java.util.*;
```

```
class Genrics <T> {  
    T var1;  
    void Genrics (T gvar) {  
        var1 = gvar;  
    }  
    T a display() {  
        return var1;  
    }  
}
```

```
public class App {  
    public static void main (String[] args)  
        throws Exception {  
        System.out.println("Hello, World!");
```

```
        Scanner Minp = new Scanner (System.in);  
        Genrics <Integer> Rollno = new Genrics (Integer);  
        Genrics <String> Name = new Genrics (String);  
        System.out.println ("Enter name of Student");  
        String Sname = Minp.nextLine();  
        Name.Genrics (Sname);  
        System.out.println ("Enter USN of Student");  
        int Sno = Minp.nextInt();  
        Rollno.Genrics (Sno);  
        System.out.println ("The student details are");  
        System.out.println ("Name: " + Name.display());  
        System.out.println ("USN: " + Rollno.display());  
        Minp.close();  
    }  
}
```



```
Command Prompt
Microsoft Windows [Version 10.0.19041.685]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Samarth>cd C:\Users\Samarth\Documents
C:\Users\Samarth\Documents>javac App.java
C:\Users\Samarth\Documents>java App
STRING INPUT
Praveen
INT INPUT
123
THE OUTPUT GOT USING GENERICS IS:123 Praveen
C:\Users\Samarth\Documents>exit
```

## //LAB 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 takes both father and son's age and throws an exception if son's age is >= father's age.

```
import java.util.*;
```

```
class ageException extends Exception{
    int detail;
    ageException(int a){
        detail = a;
    }
    public String toString(){
        return "Exception :"+detail+" the entered age does not match the category";
    }
}
```

```

}

class Father{
    int age;

    Father(int age) throws ageException{
        this.age = age;
        if(this.age<=0){
            throw new ageException(this.age);
        }
    }

    void display(){
        System.out.println("Father's age:"+this.age);
    }
}

```

```

class Son extends Father{
    Father f;

    Son(int age,Father f) throws ageException{
        super(age);
        this.f = f;
        if(this.age>=this.f.age){
            //System.out.println(f.age);
            throw new ageException(this.age);
        }
    }

    void display(){
        this.f.display();
        System.out.println("Son's age:"+this.age);
    }
}

```

```

public class Lab{

```

```
public static void main(String[] args){  
    try{  
        Scanner input = new Scanner(System.in);  
        Father f = new Father(input.nextInt());  
        Son s = new Son(input.nextInt(),f);  
        s.display();  
    }catch(Exception e){  
        System.out.println(e);  
    }  
}
```

### Program - 8

```
import java.util.*;
```

```
class ageException extends Exception {  
    int detail;  
    ageException (int a) {  
        detail = a;  
    }
```

```
    public String toString() {  
        return "Exception : " + detail + "  
        entered age does not match  
        the Category";  
    }
```

```
}
```

```
class Father {
```

```
    int age;
```

```
    Father (int age) throws ageException {  
        this.age = age;
```

```
        if (this.age <= 0) {
```

```
            throw new ageException (this.age);  
        }
```

```
    }
```

```
    void display() {
```

```
        System.out.println("Father's age :  
        this.age);  
    }
```

```
}
```

```
class Son extends Father {
```

```
    Father f;
```

```
    Son (int age, Father f) throws ageException {  
        super (age);
```

```
        this.f = f;
```

```
        if (this.age >= this.f.age) {
```

```
        throw new ageException(this.age);
    }
}

void display() {
    this.f.display();
    System.out.println("Son's age: " + this.age);
}

}

public class Lab8 {
    public static void main(String[] args) {
        try {
            Scanner input = new Scanner(System.in);
            Father f = new Father(input.nextInt());
            Son s = new Son(input.nextInt(), f);
            s.display();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}

}
```

```
Command Prompt
Microsoft Windows [Version 10.0.19041.685]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Samarth>cd C:\Users\Samarth\Documents
C:\Users\Samarth\Documents>javac Lab.java
C:\Users\Samarth\Documents>java Lab
50
19
Father's age:50
Son's age:19
C:\Users\Samarth\Documents>java Lab
2
1
Father's age:2
Son's age:1
C:\Users\Samarth\Documents>java Lab
1
2
Exception :2 the entered age does not match the category
C:\Users\Samarth\Documents>
```

## // LAB 9

```
class NewThread implements Runnable {
    Thread t;

    NewThread() {
        t = new Thread(this, "Demo Thread");
        t.start();
    }

    public void run() {
        try {
            for(int i = 25; i > 0; i--) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("thread2 interrupted.");
        }
        System.out.println("Exiting thread2.");
    }
}

class Thread1 {
```

```
public static void main(String args[]) {  
    new NewThread();  
    try {  
        for(int i = 5; i > 0; i--) {  
            System.out.println("BMS College of Engineering");  
            Thread.sleep(10000);  
        } } catch (InterruptedException e) {  
            System.out.println("Main thread interrupted.");  
        }  
        System.out.println("Main thread exiting.");  
    } }
```



LAB - 9

Write a program which creates 2 threads, one thread displaying "BMS college of Engineering" one every 10 sec and another display "CSE" every 2 seconds.

```
class New Thread implements Runnable {
```

```
Thread t;
```

```
new Thread() {
```

```
t = new Thread (this, "Demo thread");
```

```
t.start();
```

```
}
```

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

```
try {
```

```
for (int i=25; i>0; i--) {
```

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

```
Thread.sleep(2000);
```

```
}
```

```
};
```

```
} catch (InterruptedException e) {
```

```
System.out.println("thread2 interrupted.");
```

```
}
```

```
System.out.println("Exiting thread 2.");
```

```
}
```

```
}
```

```
class Thread1 {
```

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

```
new NewThread();
```

```
try {
```

```
for (int i=5; i>0; i--) {
```

```
System.out.println("BMS college of Engineering");
```

```
Thread.sleep(10000);
```

```
} } catch (InterruptedException e) {
```

```
System.out.println("Main thread interrupted.");
```

```
};
```

```
System.out.println("Main thread exiting.");
```

```
Command Prompt
Microsoft Windows [Version 10.0.19041.630]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Samarth>cd C:\Users\Samarth\Documents
C:\Users\Samarth\Documents>javac Thread1.java
C:\Users\Samarth\Documents>java Thread1
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
CSE
CSE
CSE
Main thread exiting.
Exiting thread2.
```

//LAB10

**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.**

```
import java.awt.BorderLayout;

import java.awt.Button;

import java.awt.Color;

import java.awt.Dialog;

import java.awt.FlowLayout;

import java.awt.Frame;

import java.awt.Graphics;

import java.awt.Insets;

import java.awt.Label;

import java.awt.TextField;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.TextEvent;

import java.awt.event.TextListener;
```

```
import java.awt.event.WindowAdapter;
```

```
import java.awt.event.WindowEvent;
```

```
public class Lab10 extends Frame implements ActionListener{
```

```
    TextField t1,t2;
```

```
    String msg="";
```

```
    Button btn;
```

```
    Lab10(){
```

```
        Label l1 = new Label("First Number: ",Label.RIGHT);
```

```
        t1 = new TextField(10);
```

```
        Label l2 = new Label("Second Number: ",Label.RIGHT);
```

```
        t2 = new TextField(10);
```

```
        btn = new Button("Submit");
```

```
        //Label l = new Label("Updates:");
```

```
        l1.setBackground(Color.YELLOW);
```

```
        l2.setBackground(Color.YELLOW);
```

```
        //this.setResizable(false);
```

```
        this.add(l1);
```

```
        this.add(t1);
```

```
        this.add(l2);
```

```
        this.add(t2);
```

```
        //the following command will make sure that the input char is not visible to the user
```

```
        //(it has been added just to demonstrate). Can be used for passwords.
```

```
        //t1.setEchoChar('*');
```

```
        //t2.setEchoChar('#');
```

```
        this.add(btn,BorderLayout.CENTER);
```

```
        this.setVisible(true);
```

```
        this.setSize(600, 300);
```

```
        this.setLayout(new FlowLayout(FlowLayout.CENTER,20,10));
```

```
        //t1.addActionListener(this);
```

```

        btn.addActionListener(this);

        addWindowListener(new MyWindow());

        setBackground(Color.YELLOW);

        //System.out.println(BorderLayout.CENTER);
    }

    @Override
    public Insets getInsets() {
        return new Insets(50,10,10,20);
    }

    @Override
    public void actionPerformed(ActionEvent e) {

        String st1 = t1.getText();
        String st2 = t2.getText();

        double n1,n2;

        n1 = 0.0;
        n2 = 0.0;

        if(st1.equals("") || st2.equals("")) {

            msg="You cannot leave the text elements blank";
        }else{
            try {
                n1 = Double.parseDouble(st1);
                n2 = Double.parseDouble(st2);

                try {
                    double res = n1/n2;

                    msg = "Result of division: "+res;
                }catch(ArithmeticException e1) {
                    msg = e1.toString();
                }
            }
        }
    }

```

```

        }
        }catch(NumberFormatException e2) {
            msg = "Enter only numbers and not other things";
        }
    }
    new MyDialog(this,"Result Dialog",false,msg,n1,n2);
}
public static void main(String[] args) {
    new Lab10();
}
}

```

class MyDialog extends Dialog implements ActionListener{

```

{
    public MyDialog(Frame owner, String title, boolean modal,String msg, double n1, double n2)
    {
        super(owner, title, modal);
        this.setVisible(true);
        this.setSize(300, 400);
        this.setLayout(new FlowLayout());
        //System.out.println(owner);
        Label l1 = new Label("    Updates on the result:    ");
        //l1.setSize(300, 20);
        this.add(l1);
        this.add(new Label("First Number: "+n1));
        this.add(new Label("Second Number: "+n2));
        this.add(new Label(msg));

        Button b = new Button("Close");
        this.add(b);
        b.addActionListener(this);
    }
}

```

```
        this.addWindowListener(new WindowAdapter() {  
            public void windowClosing(WindowEvent e) {  
                dispose();  
            }  
        });  
    }  
}
```

```
    @Override  
    public void actionPerformed(ActionEvent e) {  
        dispose();  
    }  
}
```

```
}
```

```
class MyWindow extends WindowAdapter{  
    public void windowClosing(WindowEvent e) {  
        System.exit(0);  
    }  
}
```

## LAB 10

```
import java.awt. BorderLayout;
import java.awt. Button;
import java.awt. Color;
import java.awt. Dialog;
import java.awt. FlowLayout;
import java.awt. Frame;
import java.awt. Graphics;
import java.awt. Insets;
import java.awt. Label;
import java.awt. TextField;
import java.event. ActionEvent;
import java.event. ActionListener;
import java.awt.event. TextEvent;
import java.awt.event. TextListener;
import java.awt.event. WindowAdapter;
import java.awt.event. WindowEvent;
```

```
public class Lab10 extends Frame implements
    ActionListener {
```

```
    TextField t1, t2;
    String msg = " ";
    Button btn;
```

```
    Lab10() {
```

```
        Label l1 = new Label("First number");
```

```
        t1 = new TextField(10);
```

```
        Label l2 = new Label("Second number");
```

```
        t2 = new TextField(10);
```

```
        btn = new Button("Submit");
```

```
        l1. setBackground (Color. YELLOW);
```

```
        l2. setBackground (Color. YELLOW);
```

```
        this.add (l1);
```

```
        this.add (t1);
```



```

this.add (l2);
this.add (t2);
this.add (btn, BorderLayout-CENTER);
this.setVisible (true);
this.setSize (600, 300);
this.setLayout (new FlowLayout (FlowLayout.
                                (CENTER, 20, 10)));

```

```

btn.addActionListener (this);
addWindowListener (new MyWindow ());
setBackground (Color. YELLOW);
}

```

@ Override

```

public Insets getInsets () {
    return new Insets (50, 10, 10, 20);
}

```

@ Override

```

public void actionPerformed (ActionEvent e) {

```

```

    String st1 = t1.getText();
    String st2 = t2.getText();
    double n1, n2;
    n1 = 0.0;
    n2 = 0.0;
    if (st1.equals (" ") || st2.equals (" ")) {
        msg = "You cannot leave the
              text elements Blank";
    }

```

Left);

Right);

}

else

{

try {

```

    n1 = Double.parseDouble (st1);
    n2 = Double.parseDouble (st2);
    try {

```

```

        double res = n1/n2;
        msg = "Result of division
              : " + res;
    }
}

```

```

    }
    catch (ArithmeticException e1) {
        msg = e1.toString();
    }
    catch (NumberFormatException e2) {
        msg = "Enter only nos and not other things";
    }
}

new MyDialog (this, "Result Dialog", false, msg);
}

public static void main (String[] args) {
    new Lantio();
}

}

class MyDialog extends Dialog implements
    ActionListener {

    public MyDialog (Frame owner, String title,
        boolean modal, String msg, double n1,
        double n2) {
        super (owner, title, modal);
        this.setVissible (true);
        this.setSize (300, 400);
        this.setLayout (new FlowLayout ());
        Label l1 = new Label ("updates on the result");
        this.add (l1);
        this.add (new Label ("First no: " + n1));
        this.add (new Label ("Second no: " + n2));
        this.add (new Label (msg));

        Button b = new Button ("Close");
        this.add (b);
        b.addActionListener (this);
        this.addWindowListener (new WindowAdapter () {

```

```
        public void windowClosing (WindowEvent e) {  
            dispose();  
        }  
    }  
    @Override  
    public void actionPerformed (ActionEvent e) {  
        dispose();  
    }  
}  
class MyWindow extends WindowAdapter {  
    public void windowClosing (WindowEvent e) {  
        System.exit(0);  
    }  
}
```

});

