```
Package CIE;
import java.util.Scanner;
Public class internals {
  Public double cieMarks[];
  Public double cieSum = 0.0;
  int i;

Public void laccept () {
Scanner in = new Scanner (System.in);
cieMarks = new double [5];
System.out.println (" Enter cie marks of 5
courses out of 50 ");
for ( i= 0; i<5; i++) {
  System.out.println (" Enter cie marks [" + (i+1)+ "]
  cieMarks [i] = in.nextDouble();

}
}
}.
```

```
Package CIE;
import java.util.*;
Public class Student {.
  Public String USN;
  Public String name;
  Public int sem;
```

```
Public void accept(){
Scanner in = new Scanner(System.in);
System.out.println("enter details
USN = in.next();
Name = in.next();
Sem = in.nextInt();
}
Public void display(){
System.out.println("USN: "+USN+"\n name: "+name+
"\n sem: "+sem);
}
}
import CIE.*;
import SEE.*;
import java.util.*;
Class FinalMarks{
Public static void main(String args[]){
int FS;
Scanner in = new Scanner(System.in);
System.out.println("Enter no. of students");
int n = in.nextInt();
CIE.Student[] cs = new CIE.Student[n];
CIE.Internals[] ci = new CIE.Student[n];
SEE.Externals[] se = new Externals[n];
```

```
for (int j=0; j<n; j++)
cs[j] = new CIE.Student();
cs[j].accept();
ci[j] = new CIE.Internals();
ci[j].iaccept();
se[j] = new Externals();
se[j].Eaccept();

}
for(int j=0; j<n; j++){
  cs[j].display();
  System.out.println("Final marks of student
  "+(j+1));
  for(int i=0; i<5; i++){
  FS = (int)(ci[j].cieMarks[i] + se[j].seeMarks[i]/2);
  System.out.println("In course: "+(i+1)+"is: "+(FS));
  }
}
}

Package SEE
import java.util.Scanner;
import CIE.*;
Public class Externals extends CIE.Student {
  Public double seeMarks[];
  int i;
```

```
Public void Eaccept() {.
  Scanner E = new Scanner (System.in);
  SeMarks = new double[5];
  Credits = new int[5];
  System.out.println ("Enter See Marks of 5 courses
  out of 100");
  for (i=0; i<5; i++) {.
  System.out.println ("Enter See marks [" + (i+1) + "]");
  SeeMarks[i] = E.nextDouble();
  }
}
}.
```

Program 7

```
class TwoGen<T> {.
  T obj1;
TwoGen (T o1) {.
  ob1 = o1;
}.
  void showtype() {.
  System.out.println ("Type of T is " + ob1.getClass().
                      getName()
                      );
}
T getob1() {.
  return ob1;
}
```

```
class simple {
Public static void main (String args[]) {
TwoGen <Integer> tgobj = new TwoGen <Integer> (88)
                                                  ;
tgobj.showTypes();
int v = tgobj.getobj();
System.out.println("value: " + v);
TwoGen <String> tgObj2 = new TwoGen <String>
                          ("This is Praveen");

tgobj2.showtypes();
String str = tgObj2.getObj();
System.out.println("value: " + str);
TwoGen <Float> ob3 = new TwoGen <Float> (23.2f
ob3.showtypes();
Float flo = ob3.getobj();
System.out.println("value: " + flo);

}
}
```

```
class Father {
static void acceptName F (int inputAge) throws
                                ArithmeticException.

{
 try {
 if (inputAge<0){.
 throw new {ArithmeticException}("Wrong age");
 }
}
 catch (ArithmeticException e) {.
  System.out.println ("caught "+e);
}!
);

class Son extends Father {
static void checkSFage (int S_Age , int F_Age)
throws ArithmeticException .{
try {.
if (S_Age >=F_Age)
throw new ArithmeticException ("Son's age should
be smaller than father's age");
System.out.println (" Son's age is "+ S_Age + "Father's
age is " + F_Age);
}
catch (ArithmeticException e) {.
  System.out.println("Fcaught " +e);
}
```

```
}
}.
Public class Exception {.
  Public static void main (String args[]){.
    father. acceptName(10);
    son. checkAge(30, 20);
  }
}.
```

Program - 9.

```
class cse extends Thread {.
  cse() {.
    super("cse thread");
  System.out.println("The thread is:" +this);
    start();
}
Public void run() {
  try {
    for(int i=0; i<5; i++) {.
      System.out.println("CSE");
        Thread.sleep(2000);
  }
}
  catch (InterruptedException e) {.
    System.out.println("Thread interrupted");
  }
  System.out.println("cse thread exiting");
}
}.
```

```
class Main {
  Public static void main(String args[]) {
    new cse();
    try {
      for(int i=0; i<5; i++) {
        System.out.println("BMS college of engineering");
        Thread.sleep(2000);
      }
    }
    catch (InterruptedException e) {
      System.out.println("Main thread interrupted");
    }
    System.out.println("Main thread exiting");
  }
}
```

Program - 10.

```
class Car_queue {
    String n;
    boolean ValueSet = false;
  synchronized String get() {
      while (!valueset) {
        try {
          wait();
        }
        catch (InterruptedException ie) { }
}.
```

```
System.out.println("Got: " + n);
valueSet = false;
notify();
return n;
}
synchronized void put(String n) {
while (valueSet) {
try {
wait();
} catch (InterruptedException e) { }
}
this.n = n; valueSet = true;
System.out.println("Put: " + n);
notify();
}
}
class Car_Owner implements Runnable {
Car_queue q;
Car_Owner(Car_queue q) {
this.q = q;
new Thread(this, "Owner").start();
}
public void run() {
while(true) {
q.put("Brakes");
}
}
}
class Car_Mechanic implements Runnable {
Car_queue q;
Car_Mechanic(Car_queue q)
this.q = q;
```

```
new Thread(this, "Mechanic").start();
}
Public void run() {
    while(true) {
        q.getcar();
    }
}
}
class inter{
    Public static void main(String args[]){
        car_queue q = new car_queue();
        new car_owner(q);
        new car_queue(q);
        System.out.println("Press control+c to stop");
    }
}
```

Mouse events.

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

Public class mouse extends Frame implements MouseListener,
MouseMotionListener {
    String msg = "";
    int mouseX = 0, mouseY = 0;

  Public mouse() {
    .addMouseListener();
    addMouseMotionListener();
    addWindowListener(new MyWindowAdapter());
.}
  Public void mouseClicked(MouseEvent me) {
      msg = msg + " Click recieved";
      repaint();
  }
  Public void mouseEntered(MouseEvent me) {
      mouseX = 100;
      mouseY = 100;
      msg = "Mouse Entered";
      repaint();
}
  Public void mouseExited(MouseEvent me) {
      mouseX = 100;
      mouseY = 100;
      msg = "Mouse Exited"
      repaint();
.}.
```

```
Public void mouseReleased (MouseEvent me) {
    mousex = me.getX();
    mousey = me.getY();
    msg = "Button Released";
    Repaint();
}
Public void mouseDragged (MouseEvent me) {
    mousex = me.getx();
    mousey = me.getY();
    msg = "*" + "mouse at " + mousex + ", " + mousey;
    Repaint();
}
Public void mouseMoved (MouseEvent me) {
    mousex = me.getx();
    mousey = me.getY();
    msg = "Mouse moving at " + mousex + mousey;
    Repaint();
}
Public void paint (Graphics g) {
    g.drawString (msg, mousex, mousey);
}
Public static void main (String args[]) {
    mouse appwin = new mouse();
    appwin.setSize (new Dimension (300, 200));
    appwin.setVisible (true);
    appwin.setTitle ("Mouse Events");
}
```

```
class MyWindowAdapter extends WindowAdapter {
    Public void windowClosing (WindowEvent we) {
        System.exit(0);
    }
}.
```

Arithmetic.java.

```
import java.awt.*;
import java.awt.event.*;
class Arithmetic extends Frame implements ActionListener {
    TextField t1, t2, t3, t4;
    Label lf1, lf2, l3, lt4;
    Button b;
Public Arithmetic () {
    setLayout (new FlowLayout());
    Label lf1 = new Label ("Field 1", Label.RIGHT);
    Label lf2 = new Label ("Field 2", Label.RIGHT);
    ~~Label lf3 = new Label ("Operation", Label.RIGHT);~~
    Label l = new Label ("Result", Label.LEFT);
    t1 = new TextField (12);
    t2 = new TextField (12);
    ~~t3 = new TextField (12);~~
    ~~t4 = new TextField (12);~~
    b = new Button ("Divide");
    add (lf1);      add (t1);
    add (lf2);      add (t2);
    ~~add (lf3);~~      ~~add (t3);~~
    add (l);        add (t4);
                                add(b);
```

```
b. addActionListener(this);
addWindowListener(new WindowAdapter());
Public void actionPerformed(ActionEvent ae) {
    if(ae.getSource() == Divide) {
        try {
        resultTextField.setText(Integer.toString(Integer.parseInt(
        Num1.getText())
try {
    String n1 = d1.getText();
    String n2 = d2.getText();
    l.setText("Quotient: " + (Integer.parseInt(n1) /
                              Integer.parseInt(n2)));
}
catch (NumberFormatException ze) {
    l.setText("Cannot divide non integer values");
}
catch (ArithmeticException ze) {
    l.setText("Cannot Divide");
} catch(Exception en) {
    System.out.println(en);
}
}
Public static void main(String args[]) {
    Lab11 app = new Lab11();
    app.setSize(new Dimension(400, 400));
    app.setTitle("Division");
    app.setVisible(true);
```

LAB PROGRAMS CODE AND OUTPUT:

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.

Student.java

```
package CIE;
import java.util.Scanner;
public class Student
{
public String name;
public String usn;
public int sem;
public void display()
{
Scanner s=new Scanner(System.in);
System.out.print("Name:-");
name=s.next();
System.out.print("USN:-");
usn=s.next();
System.out.print("Semester:-");
sem=s.nextInt();
}
}
```

Internals.java

```
package CIE;
import java.util.Scanner;
public class Internals extends Student
{
public double ciem[];
public void display()
 {
ciem=new double[5];
Scanner t = new Scanner(System.in);
System.out.println("CIE Marks for 5 subjects(out of 50):");
for(int i=0;i<5;i++)
ciem[i]=t.nextDouble();
```

```
 }
}
```

Externals.java

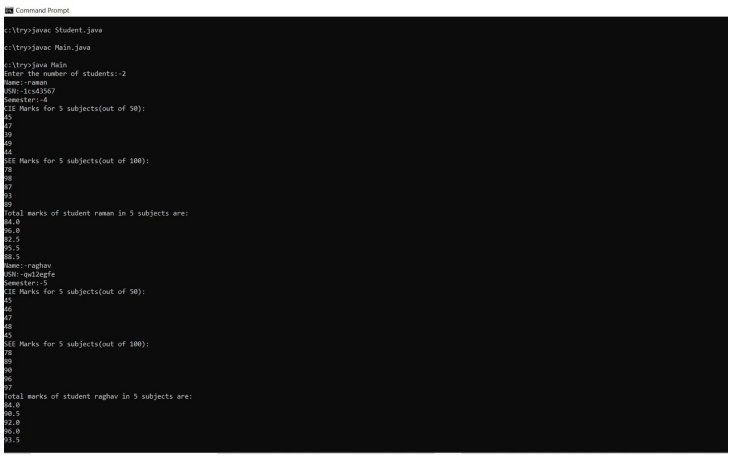
```
package SEE;
import java.util.*;
import CIE.*;
public class Externals extends Student
{
public double seem[];
public void display()
 {
seem=new double[5];
Scanner s=new Scanner(System.in);
System.out.println("SEE Marks for 5 subjects(out of 100):");
for(int i=0;i<5;i++)
seem[i]=s.nextDouble();
 }
}
```

Main.java

```
import CIE.*;
import SEE.*;
import java.util.Scanner;
public class Main
{
public static void main(String args[])
{
int n;
Scanner s=new Scanner(System.in);
System.out.print("Enter the number of students:-");
n=s.nextInt();
Student st[]=new Student[n];
Internals in[]=new Internals[n];
Externals e[]=new Externals[n];
for(int i=0;i<n;i++)
{
st[i]=new Student();
in[i]=new Internals();
e[i]=new Externals();
st[i].display();
in[i].display();
```

```
e[i].display();
System.out.println("Total marks of student "+st[i].name+" in 5 subjects are:");
for(int j=0;j<5;j++)
{
System.out.println(in[i].ciem[j]+(e[i].seem[j]/2));
}
}
}
}
```

OUTPUT:



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

```
// A simple generic class with two type
// parameters: T and V.
class TwoGen<T, V> {
T ob1;
V ob2;
// Pass the constructor a reference to
// an object of type T and an object of type V.
TwoGen(T o1, V o2) {
ob1 = o1;
ob2 = o2;
 }
// Show types of T and V.
void showTypes() {
System.out.println("Type of T is " +
ob1.getClass().getName());
// Obtain and show values.
```

```
System.out.println("value: " +ob1);
System.out.println("Type of V is " +
ob2.getClass().getName());
System.out.println("value: " +ob2);
 }
}
// Demonstrate TwoGen.
class generics {
public static void main(String args[]) {
TwoGen<Integer, String> tgObj =new TwoGen<Integer, String>(88, "Generics");
// Show the types.
tgObj.showTypes();
 }
}
```

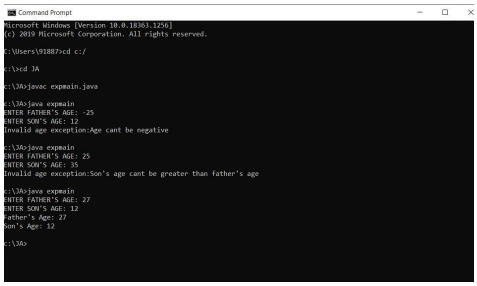
OUTPUT:

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.

```
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 x) throws WrongAge
```

```
{
age=x;
if(age<0)
 throw new WrongAge("Age cant be negative");
}
}
class son extends father{
 int age1;
 son(int fage,int sage)throws WrongAge{
 super(fage);
 age1=sage;
if(age1>=age)
 throw new WrongAge("Son's age cant be greater than father's age");
 }
}
class lab8
{
 public static void main(String args[])
 {
 Scanner s =new Scanner(System.in);
 System.out.print("ENTER FATHER'S AGE: ");
 int m=s.nextInt();
 System.out.print("ENTER SON'S AGE: ");
 int n=s.nextInt();
 try{
 son ob = new son(m,n);
 System.out.println("Father's Age: "+ob.age);
 System.out.println("Son's Age: "+ob.age1);
 }
 catch(WrongAge e)
 {
 System.out.println(e);
 }
 }
}
```

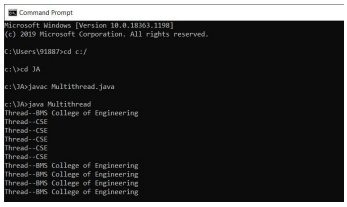
OUTPUT:



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.

```
class NewThread implements Runnable
{
 private String name;
 private int interval;
 private Thread t;
 NewThread(String threadname, int interval)
 {
 this.name = threadname;
 this.interval = interval;
 t = new Thread(this, name);
 t.start();
 }
 public void run()
 {
 try {
 for(int i=5;i>0;i--) {
 System.out.println("Thread--" + this.name);
 Thread.sleep(this.interval);
 }
 }
 catch (InterruptedException e) {
 System.out.println(name + "Interrupted");
 }
 }
}
class lab9
{
 public static void main(String args[])
 {
 new NewThread("BMS College of Engineering", 10000);
 new NewThread("CSE", 2000);
 }
}
```

OUTPUT:



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.

```
import java.awt.*;
import java.awt.event.*;
public class lab10 extends Frame implements ActionListener {
TextField num1, num2;
Label l;
Button n;
lab10() {
num1 = new TextField();
num1.setBounds(50, 50, 200, 25);
num2 = new TextField();
num2.setBounds(50, 100, 200, 25);
l = new Label();
l.setBounds(50, 150, 300, 50);
n = new Button("Divide");
n.setBounds(50, 200, 100, 50);
n.addActionListener(this);
add(n);
add(num1);
add(num2);
add(l);
setSize(800, 800);
setLayout(null);
setVisible(true);
}
public void actionPerformed(ActionEvent e) {
try {
String n1 = num1.getText();
String n2 = num2.getText();
l.setText("Quotient: " + (Integer.parseInt(n1) / Integer.parseInt(n2)));
```

```
} catch (NumberFormatException ze) {
l.setText("Cannot divide non-integer values");
} catch (ArithmeticException ze) {
l.setText("Cannot divide");
} catch (Exception ex) {
System.out.println(ex);
}
}
public static void main(String[] args) {
new lab10();
}
}
```

OUTPUT: