SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Established in collaboration with MIT

Introduction to Information System and Programming

Week 3: Cohort Session 2:

Title: ModularProgram

Q1. Write a program that performs modular operation of two integer values, for example x % y. Prompt the user to enter two integer values. Catch an exception thrown if modular by zero. Also, catch all other exceptions using Exception. Your program should be in a loop until not 'y' is entered.

Possible answer:

```
Welcome to the modulus computer
Enter two integer values
2
0
Sorry, cannot compute mod by 0
Do another pair of values ? (y)
y
Enter two integer values
1
1.1
Sorry, you must enter two integer values
Do another pair of values ? (y)
N
```

Title: TestScores

Q2. Write a program that checks the scores stored in an array. The method, TestScores, will throw IllegalArgumentException("Element: " + i + " Score: " + s[i]) if the values in array *s* are less than 0 or bigger than 100. Use try … catch block in main() to handle this exception.

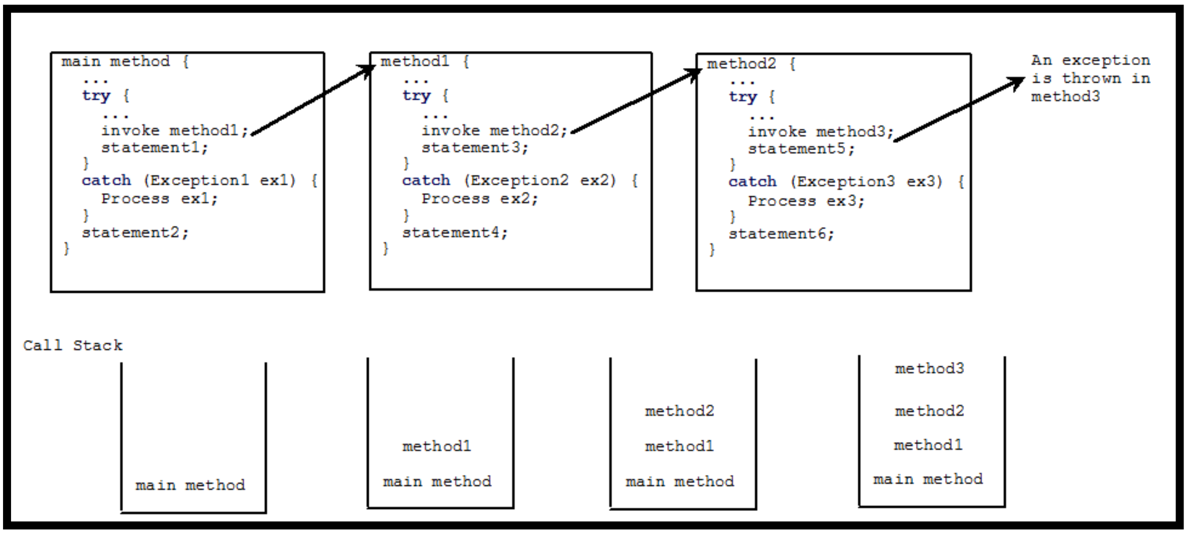
```
import java.util.Scanner;
public class TestScores {
    public static void TestScores(double[] s)  {
              Type your codes here

}
public static void main(String [] args) {
   try {
        double[] values = {1.2,1.3,1,102};
       TestScores(values);
       }
  catch (IllegalArgumentException ex) {
     System.out.println(ex);
     System.out.println("Exception: Values cannot be less than 0 or bigger than 100 ");
     }
System.out.println("Execution continues here");
} }
```

Possible answer:

```
java.lang.IllegalArgumentException: Element: 3 Score: 102.0
Exception: Values cannot be less than 0 or bigger than 100
Execution continues here
```

Title: <u>CatchExceptions</u>

Q3.



```
main method {
  ...
  try {
    ...
    invoke method1;
    statement1;
  }
  catch (Exception1 ex1) {
    Process ex1;
  }
  statement2;
}
```

```
method1 {
  ...
  try {
    ...
    invoke method2;
    statement3;
  }
  catch (Exception2 ex2) {
    Process ex2;
  }
  statement4;
}
```

```
method2 {
  ...
  try {
    ...
    invoke method3;
    statement5;
  }
  catch (Exception3 ex3) {
    Process ex3;
  }
  statement6;
}
```

Implement the diagram above. Then execute the following scenarios:
(a) An exception is thrown in method3 and is caught in method1. Print the statements that are executed. // use CatchingExceptionsv4.java
(b) An exception is thrown in method3 and is not caught in method2, method1, and method. Print the statements that are executed. // use CatchingExceptionsv6.java

```
import java.util.*;
public class CatchingExceptionsv4 {

        public static void main(String[] args) throws Throwable {
                try {
                        method1();
                        System.out.println("Statement1");
                }
                catch (NullPointerException ex) {
                        System.out.println(ex);
                }
                System.out.println("Statement2");
        }
        public static void method1() throws Exception {
                try {
                        method2();
                        System.out.println("Statement3");
                }
                catch ________________________ {
                      ________________________

                }
                System.out.println("Statement4");
        }
        public static void method2() throws Exception {
                try {
                        method3();
                        System.out.println("Statement5");
                }
                catch _____________________ {
                      _____________________

                }
                System.out.println("Statement6");
        }
        public static void method3() throws Exception {
                throw new IllegalArgumentException("Info from method2");
        }

}
```

Possible answer:

```
java.lang.IllegalArgumentException: Info from method2
Statement4
Statement1
Statement2
```

```
import java.util.*;
public class CatchingExceptionsv6 {

        public static void main(String[] args) throws Throwable {
                try {
                        method1();
                        System.out.println("Statement1");
                }
                catch (NullPointerException ex) {
                        System.out.println(ex);
                }
                finally  {
                           ___________________________
                }

                System.out.println("Statement2");
        }
        public static void method1() throws Exception {
                try {
                        method2();
                        System.out.println("Statement3");
                }
                  catch _____________________ {

                        ______________________
        }
        System.out.println("Statement4");
    }
    public static void method2() throws Exception {
        try {
            method3();
            System.out.println("Statement5");
        }
        catch ________________________ {

          _________________________

        }
        System.out.println("Statement6");
    }
    public static void method3() throws Exception {
        throw new IllegalArgumentException("Info from method2");
    }

}
```

Possible answer:

```
Last statement
Exception in thread "main" java.lang.IllegalArgumentException: Info from method2
	at CatchingExceptionsv6.method3(CatchingExceptionsv6.java:43)
	at CatchingExceptionsv6.method2(CatchingExceptionsv6.java:33)
	at CatchingExceptionsv6.method1(CatchingExceptionsv6.java:21)
	at CatchingExceptionsv6.main(CatchingExceptionsv6.java:6)
```