Exceptions Lab

This lab contains exercises related to Exceptions, and overriding methods of the Object Class.

Start Eclipse IDE. Create a new Java project named **Week4-Project,** and complete the following tasks:

Task 1: Handling unchecked exceptions

Create a class named ***TestUncheckedException*** with the following code:

```
public class TestUncheckedException {

   public static void test(int num) {

      int x = 10;

      int[] ia1 = null, ia2 = new int[2];

          if (num == 0) {

              x = 20 / num;

          } else if (num == 1) {

              x = ia1.length;

          } else if (num == 2) {

              x = ia2[2];

          } else {

              x = num;

          }

   }

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

```
        System.out.println("Please enter 0, 1, 2 or any other number: ");
        java.util.Scanner sc = new java.util.Scanner(System.in);
        test(sc.nextInt());
    }
}
```

Create exception handlers to catch the exceptions and print out what exception is caught in each case. Make sure the value of variable x is always printed.

Task 2: Handling checked exceptions

Create a class named ***TestCheckedException*** with the following code:

```
import java.awt.Desktop;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
public class TestCheckedException {
   public static void test1(String myURL) {
        URI uri = new URI(myURL);
        if (Desktop.isDesktopSupported()) {
            Desktop.getDesktop().browse(uri);
        }
   }
   public static void test2(String myURL) {
        URI uri = new URI(myURL);
        if (Desktop.isDesktopSupported()) {
```

```
            Desktop.getDesktop().browse(uri);

        }

    }

    public static void main(String[] args) {

        System.out.println("Please enter an URL (e.g.
https://ict.senecacollege.ca/): ");

        java.util.Scanner sc = new java.util.Scanner(System.in);

        String url = sc.nextLine();

        test1(url);

        test2(url);

    }

}
```

The file is not compiled because of the possible checked exceptions which are not handled. Update the code of method ***test1*** to handle the check exception(s) using the "catch" approach; update the code of method ***test2*** to handle the check exception(s) using the "specify" approach.

Task 3: Creating custom (user-defined) exception

Create a user-defined exception named ***UnexpectedUserInputException*** in the project.

Update the ***TestUncheckedException*** class in the Task 1. If the number user input is not one of the integers: 0, 1 or 2, the ***test*** method should throw an exception of ***UnexpectedUserInputException*****.** Then, handle the exception in the main method.

Task 4: Creating classes with overriding methods

Create a ***Course*** class that has a name, an id, and a description. Create a ***Book*** class that has a name, ISBN and price. The Course must have a field data of type array of ***Book***. Create constructors, getters and setters for the two classes, and override the toString(), equals(), hashCode() and clone() methods.