

5COSC019W - Tutorial 9 Exercises

1 How Exceptions change the flow of control of your program

Attempt the interactive exercise below:

<https://codecheck.io/assignment/201115224672ami00gbpy5xsk7a7y5zh6dc>

1. You do not need to share the ID with your tutor!
2. If you would like to redo the exercise make sure that you click the ‘Clear ID now’ button.

2 File Opening Exception

Attempt the interactive exercise below:

<https://codecheck.io/assignment/2011152130695jffjxwpx7jh4lkt7sgf0g28>

1. You do not need to share the ID with your tutor!
2. If you would like to redo the exercise make sure that you click the ‘Clear ID now’ button.

3 Throwing Exceptions

Attempt the interactive exercise below. Make sure you drag the lines of code from the left to the right in the right order and also that you decide the right indentation of your code:

<https://codecheck.io/assignment/2011152333cistjsllqogqrsy0t7ihlg3r0>

- You do not need to share the ID with your tutor!
- If you would like to redo the exercise make sure that you click the ‘Clear ID now’ button.

4 Throwing Exceptions (cont’ed)

This is a continuation from the previous exercise:

<https://codecheck.io/assignment/201115233969ifc871o03j9dzhyy5nym5np>

5 More on Learning to Deal with Exceptions

Attempt the interactive exercise below. Make sure you drag the lines of code from the left to the right in the right order and also that you decide the right indentation of your code:

<https://codecheck.io/assignment/20111523558pakcewgyoru3i0txlyo5atvn>

- You do not need to share the ID with your tutor!
- If you would like to redo the exercise make sure that you click the ‘‘Clear ID now’’ button.

6 Testing your Knowledge on the Types of Exceptions

Attempt the interactive exercise below:

<https://codecheck.io/assignment/2011152359xv5sqjblth3aimsbtwj1rza>

- You do not need to share the ID with your tutor!
- If you would like to redo the exercise make sure that you click the ‘‘Clear ID now’’ button.

7 Dealing with Exceptions

Add a `readList` method to the following code `ListOfNumbers.java`. This method should read in int values from a file, print each value, and append them to the end of the vector. You should catch all appropriate errors. You will also need a text file containing numbers to read in.

```
/*
 * Copyright (c) 1995, 2008, Oracle and/or its affiliates. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * - Redistributions of source code must retain the above copyright
 *   notice, this list of conditions and the following disclaimer.
 *
 * - Redistributions in binary form must reproduce the above copyright
 *   notice, this list of conditions and the following disclaimer in the
 *   documentation and/or other materials provided with the distribution.
 *
 * - Neither the name of Oracle or the names of its
 *   contributors may be used to endorse or promote products derived
 *   from this software without specific prior written permission.
 *
 */
```

```

* THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
* IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
* EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
* PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
* PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
* LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
* NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
* SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*/

```

```

import java.io.*;
import java.util.List;
import java.util.ArrayList;

public class ListOfNumbers {
    private List<Integer> list;
    private static final int SIZE = 10;

    public ListOfNumbers () {
        list = new ArrayList<Integer>(SIZE);
        for (int i = 0; i < SIZE; i++)
            list.add(new Integer(i));
    }

    public void writeList() {
        PrintWriter out = null;

        try {
            System.out.println("Entering try statement");
            out = new PrintWriter(new FileWriter("OutFile.txt"));

            for (int i = 0; i < list.size(); i++)
                out.println("Value at: " + i + " = " + list.get(i));
        } catch (IndexOutOfBoundsException e) {
            System.err.println("Caught IndexOutOfBoundsException: " +
                e.getMessage());
        } catch (IOException e) {
            System.err.println("Caught IOException: " + e.getMessage());
        } finally {
            if (out != null) {
                System.out.println("Closing PrintWriter");
                out.close();
            } else {
                System.out.println("PrintWriter not open");
            }
        }
    }
}

```

```

    }

    // ... method readList goes here ...

    public static void main(String[] args) {
        ListOfNumbers program = new ListOfNumbers();
        program.readList();
        program.writeList();
    }
}

```

8 More on Exceptions

Modify the following `cat` method so that it will compile.

```

public static void cat(File file) {
    RandomAccessFile input = null;
    String line = null;

    try {
        input = new RandomAccessFile(file, "r");
        while ((line = input.readLine()) != null) {
            System.out.println(line);
        }
        return;
    } finally {
        if (input != null) {
            input.close();
        }
    }
}
}

```