Winston Wang

Mr. Kuszmaul

9 November 2014

AP Computer Science

Exception Exercises

Questions:

1. Yes, the code is legal. The try block does not need a catch block as long as there is a finally block. It just means that there is code that must always be executed
2. This type of exception handler catches every exception. The problem with this catcher is that there is no way to differentiate between different types of exceptions. Thus, the program would have to have some other way of determining the type of exception in order to handle it correctly.
3. The problem is that the first catch block will catch all exceptions, including the exception that should be caught in the second catch block. Thus, the code in the second catch block would never be reached. Thus, the code does not compile.
4. 1. b  
   2. d  
   3. a  
   4. c

Exercises

1.

/\*

\* 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;

/\*\*

\* This is the constructor

\* **@author** winst13

\*

\*/

**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));

}

/\*\*

\* This method writes the the list

\*/

**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 < ***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");

}

}

}

/\*\*

\* This method is the method I wrote for exercise 1. It takes

\* the list at a given address and turns it into a list of numbers

\* **@param** filename

\*/

**public** **void** readlist(String filename)

{

String line = **null**;

RandomAccessFile file;

**try**

{

file = **new** RandomAccessFile(filename, "r");

**while** ((line = file.readLine()) != **null**)

{

Integer nextnum = **new** Integer(Integer.*parseInt*(line));

System.***out***.println(nextnum);

list.add(nextnum);

}

}

**catch**(FileNotFoundException e)

{

System.***err***.println("File: " + filename + " not found.");

}

**catch** (IOException e)

{

System.***err***.println(e.toString());

}

}

}

2.

**import** java.io.\*;

**public** **class** Cat

{

/\*\*

\* This method is the method I fixed for exercise 2. Previously

\* it had a couple uncaught exceptions

\* **@param** file

\*/

**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**;

}

**catch**(FileNotFoundException e)

{

System.***err***.println("File: "+file+"not found");

}

**catch**(IOException e)

{

System.***err***.println(e.toString());

}

**finally**

{

**if** (input != **null**)

{

**try**

{

input.close();

}

**catch**(IOException e)

{

}

}

}

}

}