Work on this lab with a new partner: someone with whom you have not yet worked.

Write the code for the lab below and each partner must upload the zipped src folder to the learning hub (learn.bcit.ca 🡪 Activities 🡪 Assignments 🡪 Lab4) before the start of lesson 5.

Include your full name and your partner’s full name at the top of each file, using a Java Javadoc   
/\*\* @author \*/ tag (for example: /\*\* @author Paul Mills and Jason Wilder \*/).

Re-do lab 3 but this time the following methods must throw the following Exceptions given the following conditions:

Make sure your methods declare the Exceptions they throw (and Javadoc @throws too). For example:

/\*\*

\* @param parameter - some parameter

@throws IllegalArgumentException if parameter is bigger than 100, or is null

\*/

public void pretendMethod(String parameter) throws IllegalArgumentException

{

// code goes here

}

**printTitlesOfLength(int length)**

Must throw an IllegalArgumentException if length is 0 or negative with the message “bad length”

**printNameStartsEndsWith(String substring)**

Must throw a checked Exception (YOU create the class) named IllegalNameException with the message "bad name" if the substring parameter is null or blank

**printTitlesContaining(String substring, boolean caseSensitive)**

Must throw an IllegalArgumentException (with message “bad string”) if the substring parameter is blank or null

NOTE: when a method or constructor throws one of several different methods, the try/catch block is extended as follows:

try

{

// code to try

}

catch(ExceptionTypeOne e)

{

// handle exception type one here

}

catch(ExceptionTypeTwo e)

{

// handle exception type two here

}

Etc…

**getLongest(String property)**

Must throw an unchecked Exception (YOU create the class) named IllegalNovelPropertyException with the message "bad property" if its value is not "author" nor "title" (in any letter casing)

**main(String[] args)**

Must try/catch all of its methods now; in the event of an Exception being thrown, simply print the Exception object’s getMessage() method output.