EECS 1720 - W2023

LAB 1 :: Reviewing Java Basics & Working with Simple Classes

Prerequisite - please ensure that you have setup your EECS account (see lab0) and can log into the lab machines prior to starting this lab! You need to be able to access websubmit (log in) to be able to submit your work.

The steps for importing are given for you in STEP 1 below.
The steps for navigating to and submitting online via websubmit are given in STEP 3.

Lab Resources:

Java API: https://docs.oracle.com/javase/8/docs/api/

```
java.lang.Math      // in the above link, pick the package java.lang
                    // then click on the class Math (to access its documentation)
                    // similarly for the following classes
java.lang.String
java.lang.Integer
java.lang.Double
java.lang.Character
java.lang.String
java.lang.StringBuilder
java.lang.ArrayList
java.util.Scanner
```

Lab Files: http://www.eecs.yorku.ca/course_archive/2022-23/W/1720/labs/lab1/lab1.zip

Lab Files API:
https://www.eecs.yorku.ca/course_archive/2022-23/W/1720/labs/lab1/doc/lab1/package-summary.html

STEP 1: Importing an Archived Project

In this step, you will import an archived project into your Eclipse workspace, (the files for each lab exercise are embedded). Each question refers to a separate java file. The instructions for what is required for each question is included in the document below (STEP 2 onward). It is a good idea to create a separate workspace (e.g. "EECS1720") for this course.

a. You can download the lab project file from the link above (see Resources)
   This file is a *zip file*, also known as an *archive file*. Click on the link below to open the URL in your browser. In the dialog that opens, choose **Save**, not **Open**. This file is typically saved into your *home* or *downloads* folder.

b. Open Eclipse and import this archive (as per the method given in lab0, TASK 4).

IMPORTANT:* *this process is the same as the process you will use in programming tests, so please make sure you follow it, so that you do not have submission difficulties during the lab tests.

DO NOT DOUBLE CLICK ON THE FILE TO OPEN. By importing this project, you will add it into your *EECS1720* workspace (no matter where you had downloaded the file). If you DOUBLE CLICK, the file will open where you downloaded it (so won't be organized as conveniently into your workspace). Do this by:

i. Open Eclipse (**Applications → Programming → Eclipse**) & set/choose your workspace
ii. Close "Welcome" tab, and navigate to **File → Import → General → Existing Projects into Workspace.** Hit "next"
iii. Choose the archive file (zip file you downloaded in (a)). After selecting, hit **Finish**, and the file will open up as a project in your Project Explorer.
iv. We are now ready to proceed with the Lab Exercises.

STEP 2: Lab Exercises

The exercises in this lab are pitched at improving your ability to comprehend API's and complete a set of utility methods that review basic Java concepts (dealt with in EECS1710, however now using static methods and pure java features.

In the process, we will make use of Wrapper classes (Integer, Double, Character) to more conveniently handle and work with primitive types. Lastly, we will learn how to run some included JUnit test files (that will assess whether the methods are complete/working). We are not learning how to write test files in this course per se, just utilizing them where possible to help guide us when coding.

JUnit is a framework that allows you to define and setup a collection of "UNIT" of tests for your code (tests that run methods with a given input and expected output, checking to see if the output is indeed correct, and generating a report of which tests passed and failed). The instructor will run through the use of these tests in the lab session (and will post a recording of this for future reference).

There are 3 exercises that you should address in the following order (NumUtils.java, LogicUtils.java, StrUtils.java) in which you will complete methods using knowledge from EECS1710, and some ideas introduced in the first lectures of EECS1720 (namely using wrappers, utility classes, and getting data into a java program via command line/Scanner).

======
Goal: Read the instructions provided in the comments of the `*.java` source files. These exercises are a review on your java basics (working with primitives, expressions,

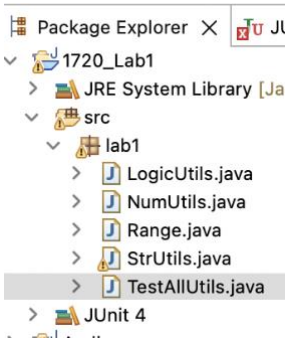
conditional logic, and simple objects such as Strings and the Wrapper Classes: e.g. Integer, Double, Character), and potentially String/StringBuilder classes, and the use of regular expressions (regex) for methods in the String class.

In these tasks, you are asked to complete several methods. A description of how the methods should work (API) can be found in the blue comments before each method. These comments form the basis for some documentation

JUnit Tester (TestAllUtils.java)

JUnit (a testing framework for java) is used in additional Test files (one for each java source). These can be opened and run to evaluate all three Utils classes you have to complete (with the exception of the last method in StrUtils.java, which does not have any tests – see TASK 3).

Once you have imported your project you should see the following in your package explorer (left margin). Open up the TestAllUtils.java file by double clicking on it, then run and you will see a new tab open in the left margin called "JUnit" – this is where you can see the results of the tests run.



Whenever you run this file, it will call each of the methods and run it with some test input then figure out if the method works or not. If it works, it will show as green (with a tick), when all methods are working, you will see a green message saying so.

The idea is to complete all the methods so that they are working and passing the tests. When a test is not working, the failure trace at the bottom of the JUnit tab will give a message indicating what was expected versus what the method returned. For example, in the above image, you can see that test01_maxInt failed, and reported that the method is returning 0 when it is expected to return 2147483647 (which is the max int value – a constant stored in Integer class). The method is expecting to return this value (when you fix it, the test will pass and you can move onto the next method)

TASK 1 – NumUtils.java

In this task, you need to complete the NumUtils class (by completing its methods).

The API for this class can be found here:
https://www.eecs.yorku.ca/course_archive/2022-23/W/1720/labs/lab1/doc/lab1/NumUtils.html

This API doc is also reflected in the comments in the file. Please read and follow the comments in the file to complete the methods. Run the tester to check if you have completed the methods correctly (they act as a guide).

TASK 2 – LogicUtils.java

In this task, you need to complete the LogicUtils class (by completing its methods). Follow the comments (in the API link below or in the file itself).

The API for this class can be found here:
https://www.eecs.yorku.ca/course_archive/2022-23/W/1720/labs/lab1/doc/lab1/LogicUtils.html

** Note the last two questions refer to a provided class **Range**:
https://www.eecs.yorku.ca/course_archive/2022-23/W/1720/labs/lab1/doc/lab1/Range.html

The API will indicate how you can use the methods of this class (you will need them to complete the methods “contains()” and “compareTo()”.

TASK 3 – StrUtils.java

In this task, you need to complete the StrUtils class (by completing its methods). Follow the comments (in the API link below or in the file itself).

The API for this class can be found here:
https://www.eecs.yorku.ca/course_archive/2022-23/W/1720/labs/lab1/doc/lab1/LogicUtils.html

Note, the Character wrapper class is useful for the "alternatingCaps" method. The last method windChillUserInputs() does not have a tester, and it is intended to be used to pass user input (keyboard typed or passed from command-line) into the windChill method from TASK 1. Lecture 3 notes will help with completing this method, you will need to call the method from the main() method. Which is included in StrUtils.java. Note this is the only method that can be run (TASK 1 and 2 both do not have main methods. If you want to run them, you must create a main method, and use it to call your methods). You don't need to do this as the tester runs these instead.

STEP 3: Submission (Deadline: Tuesday 24st January, 11:59pm)

You will formally submit the java files: **NumUtils.java**, **LogicUtils.java** & **StrUtils.java**

SUBMIT ONLY THE REQUESTED *.java FILES (**NO ZIP FILES OR CLASS FILES!!**).

Use the web-submit function.
Web-submit can be found at the link: https://webapp.eecs.yorku.ca/submit/

** submission will be closed after the deadline. Ensure you submit early and often up until you complete the lab, it is not suggested to wait until the last minute.