EECS 1720
LAB 2 :: Working with Exceptions

Prerequisite - please ensure that you have setup your PRISM account and can log into the lab machines prior to starting this lab, and you have completed lab1!

Please refer to lecture notes lectures L5, L6 – week3 (on eClass) to work through this material. Instructions are given in the blue/green comments embedded in the code (question files), they will guide you through.

PLEASE READ AND FOLLOW THE COMMENTS

Lab Resources:

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

Lab Files: http://www.eecs.yorku.ca/course_archive/2022-23/W/1720/labs/lab2/lab2.zip

Lab API: (instructions for questions in API format)
http://www.eecs.yorku.ca/course_archive/2022-23/W/1720/labs/lab2/doc/index.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 2 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
   IMPORTANT: this process is the same as the process you will use in lab 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. 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

Question 01:

Goal: To prompt the user to input the current time via a standard time formatted string, and then output the number of minutes that have elapsed since midnight.

In this exercise, you are to complete the "elapsedMins1()" method, to process an input string that provides a formatted representation of the time that is in a 12hr format: e.g. "h:m" where h = hour, and m = minutes.

where:
h ranges between: $0 < h \leq 12$
m ranges between: $0 \leq m \leq 59$

See https://en.wikipedia.org/wiki/12-hour_clock

** There are two ways to run this program:

1. You will notice that the main has some code already in place to accept and parse a line of text typed at runtime by the user. If you run the question file directly (i.e. run Question01.java), then the main() method will be run, and the program will prompt you to enter the time string. After entering, and pressing return, the program will try to call the elapsedMins1() method to process the string and interpret the time (further instructions in API link, and blue/green comments in the question file). Please follow all instructions inside the main/elapsedMins1() method. Run the file this way to manually test inputs to your method

2. The second way to run is to run the TestQuestions.java file. This will run a tester that attempts to test all of the questions from the lab. In reality it is running each questions version of the elapsedMinsX() method – against various inputs. This is a good guide to use to check your method(s) work

More detailed instructions are provided in the provided *Question01.java* file in the imported project. You do not need to provide any Exception handling for Question01. Complete this method first (when working, all testQ1_* tests should pass).

The API for this lab is given at the top of this file (URL link)

** please note, in this project, a `PrintStream` object is used to represent `System.out` (this saves some typing when you want to print outputs to standard out)

Question 02:

Goal: to implement some increased functionality to the application created in Question01 with exception-handling to the application to satisfy requirements defined in the "App Behaviour" section of the comments.

Read and follow the instructions given in the comments of Question02.java

** there are some written questions posed in this and remaining Questions. Please provide answers in your code (within comments) where requested.

Questions 03-05:

Goal: Additional behaviours for the application are now defined for each of the remaining questions. You need to develop your solution further to handle these new behaviours (again, by increasing the use of Exceptions in your code).

Read and follow the instructions given in the comments of each file Question03, Question04 & Question05.java

---

STEP 3: Submission (Deadline: Tuesday 31st January, 11:59pm)

You will formally submit the java files: **Question01-Question05.java** ONLY

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.