EECS 1710
LAB 1 :: Data Types & Simple Expressions

<u>Prerequisite</u> (Lab 0) – please ensure that you have setup your EECS account, and can log into the lab machines prior to starting this lab. Also make sure you are familiar with the Linux terminal and the submit command (remember you can only use the submit command from the terminal IN THE LAB MACHINE or via REMOTE LAB, if you do this lab on your laptop (not via REMOTE LAB), you need to submit using WEB SUBMIT – see instructions at the end of this lab).

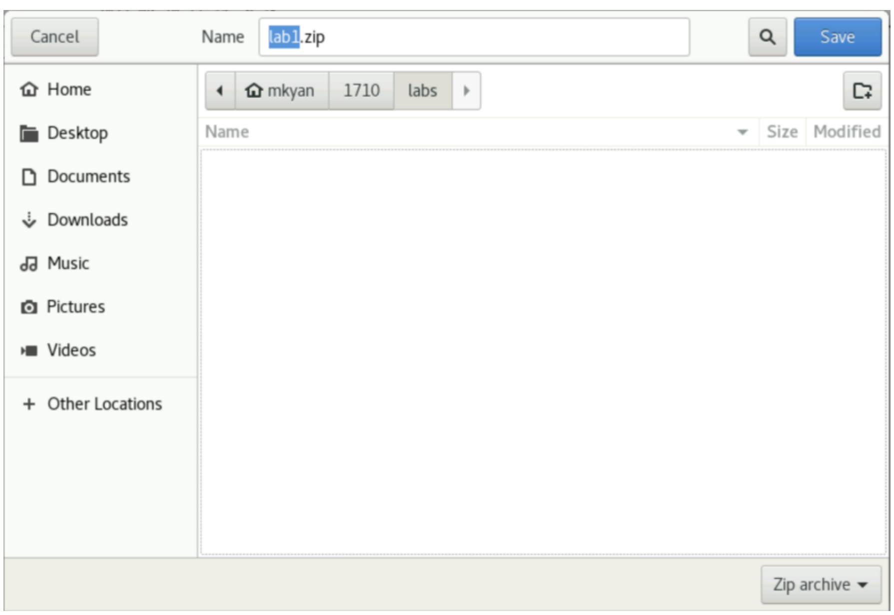
STEP 1: Importing and unzipping the starter code

In this step, you will import an archived zip folder containing the starter code for the lab. Each question refers to a separate *.pde sketch folder + file in the unzipped "labX" parent folder (where X = the lab number).

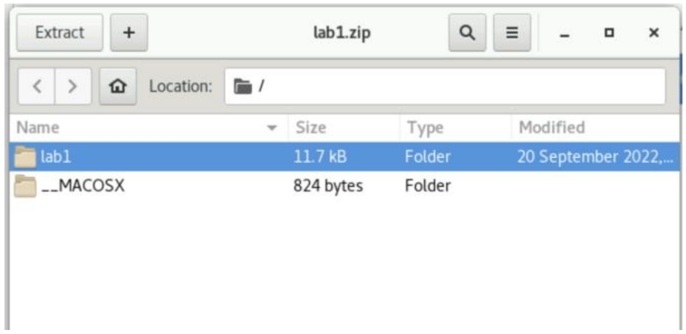
a. You can download the Lab 1 starter code from the following link:
   http://www.eecs.yorku.ca/course_archive/2022-23/F/1710/labs/lab1/lab1.zip

   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**.

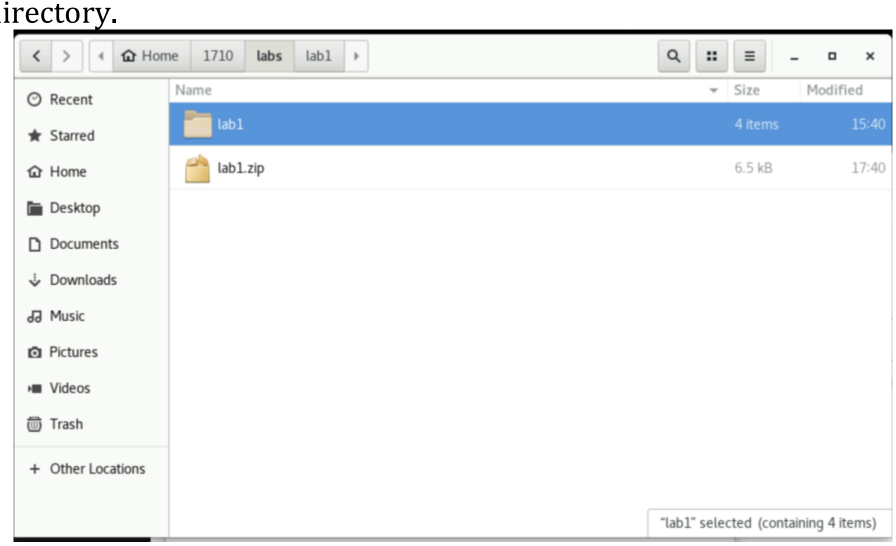
   You should then choose to download this straight into your **1710->labs** subfolder that you setup in lab0 (within your home directory), or to the equivalent **labs** subfolder you have setup within your main sketchbook folder.



Now double-click on the downloaded **lab1.zip** file and select the **lab1** folder in the archive and press the Extract button (this is all we need to extract):



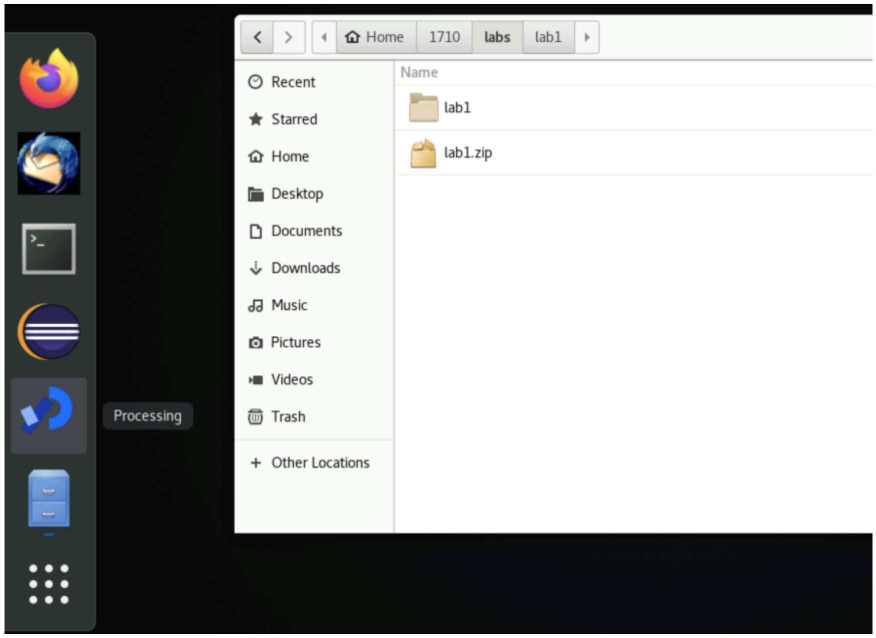
The lab folder (**lab1**) will now be extracted to your **labs** sub-folder within your **1710** directory.



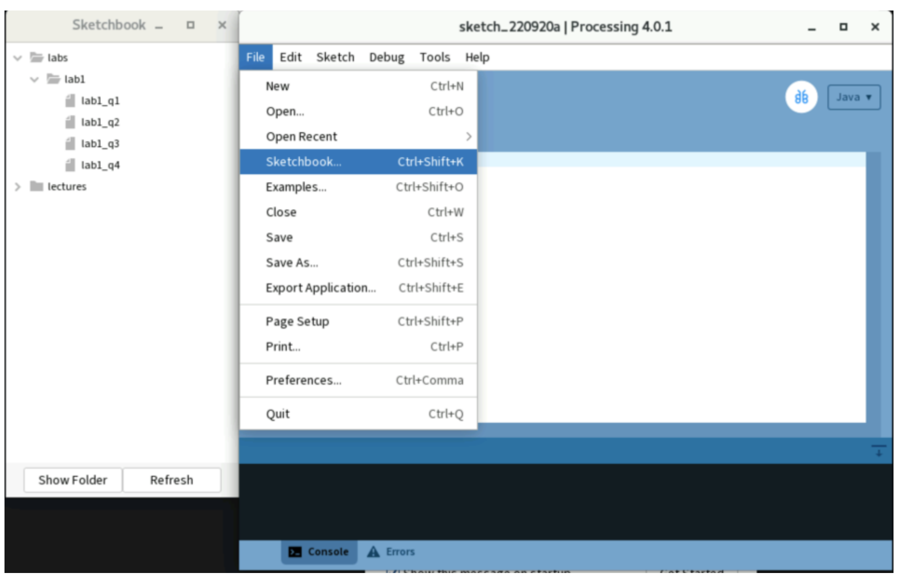
We will repeat this procedure for each new lab (so that all of your labs will reside in the labs sub-folder of your 1710 directory). You can delete the **lab1.zip** file.

*If you are completing/working on this lab from your own home computer/laptop, you will need to extract this into a sub-folder within your main sketchbook. It is recommended to setup a sub-folder called **1710** to mirror your setup on the eecs lab / remote-lab machines. Ultimately you will need to locate your files after completing the lab tasks, so that you may submit the files for marking (using submit command or websubmit portal – as introduced in lab0)*

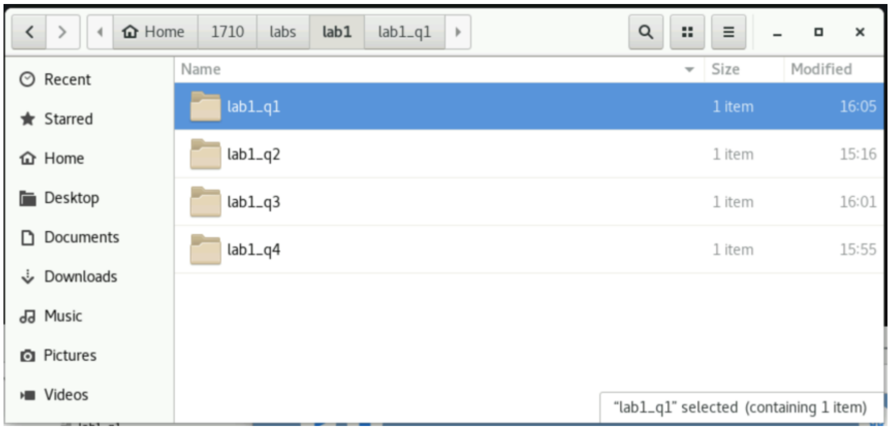
b. Launch Processing (setup as an icon in your favourites in lab0).



c. In Processing, a new empty sketch will open up first (we can close this later). To open the *.pde files for this lab, either open the sketchbook (File->Sketchbook) and navigate to the lab1 folder, or double click on the relevant *.pde file directly in the File Browser:



OR



d. Open a terminal, and navigate to the **labs** folder. Use the **cd** command to do this (see lab0). If you have followed instructions in lab0, then it should be located in the following folder in <u>your</u> *home* directory):

   e.g. the source folder for lab1 should be here:

```
cd
cd 1710/labs
```

   For now, you can minimize this terminal window (do not close). We will use it when ready to submit.

<u>STEP 2: Exercises</u>

Open the first question **lab1_q1** from the sketchbook, or locate and double click the **lab1_q1.pde** file in your File Browser. Do this for each of the 4 questions (explained below), when you are working on each (do them one by one in order is best).

Each question requires you to create a short simple program to compute and expression and output a result to the console (or create a result visually in an application window). Instructions for each program are given below.

<u>Question01 (time/distance calculation)</u>

In this task, you are asked to create any necessary **variables** and **expressions**, then use them to write a static program (i.e. no setup or draw blocks) to achieve the following task.

In ***lab1/lab1_q1/lab1_q1.pde***, write code to:

a. Calculate how long it will take person A (in hours) to travel 540.8km (from Toronto to Montreal) at an average speed of 78km/hr.
b. Calculate how much shorter it will take person B (in minutes) if they were travelling at an average speed of 112km/hr.
c. Output the results according to the format specified below

The results for these two computations should be displayed exactly as shown below, except that both `XX` and `YY` are each substituted with a real-valued number representing the resulting times for parts (a) and (b) respectively. *Take note, part (b) requires a time in minutes, not hours.*

Example output:

```
The time for Person A to travel 540.8km @ 78km/hr is XX hours.
If travelling @ 112km/hr, Person B will save YY minutes!
```

Make sure you save your work frequently. To submit your work, please refer to the submission instructions at the end of this lab document.

Question 2 (Formatting Output for Question 1)

You may have noticed that in *Question 1*, the numeric value that you print to the screen has many decimal places. In such cases, we can format the output in a way that we can control the spacing/decimal places or rounding to be used when printing. We can do this using the `nf()`, `int()` or `round()` commands (see Processing reference library for more info) to control the look of these numeric values.

In ***lab1/lab1_q2/lab1_q2.pde***, we will use the `nf()` command to print out and format the decimal places/ significant digits displayed in the time values from *Question 1* (i.e. `XX` and `YY`).

Copy and paste the *completed* code inside your main method from ***lab1_q2.pde*** into your ***lab1_q2.pde*** file (where indicated).

Now modify your code so that:

a. the time associated with **XX** is displayed with **3 decimal places**
b. the time associated with **YY** is displayed as **an integer (no decimal places)**

for example, if XX = 14.123167888, and YY = 3.4103333333
Your output would now look something like this:

```
The time for Person A to travel 540.8km @ 78km/hr is 14.123 hours.
If travelling @ 112km/hr, Person B will save 3 minutes!
```

Question 3 (colour to grayscale conversion)

Imagine that your Programming sketch has access to the 3 separate colour components of an image pixel: i.e. red, green and blue. Assume that of these components can store an integer value between 0-255, each of which is stored in separate variables (r, g, b) for the colours (red, green and blue) respectively.

This representation of a pixel is known as the RGB colour space.

In ***lab1/lab1_q3/lab1_q3.pde***, you are to convert an RGB representation (3 variables) into a single equivalent luminance (grayscale) value. This calculation is used for example to convert each pixel in a colour image to an equivalent grayscale (black and white) value for printing on a black and white printer.

The formula for converting (r,g,b) to a single luminance value (y) is:

$$\mathbf{y} = 0.2989\mathbf{r} + 0.5870\mathbf{g} + 0.1140\mathbf{b}$$

Note: y should be an integer value once the calculation is done, and it should store values between 0-255. You should also use constants to remove any 'magic numbers' from your code (see discussion in lecture notes)

Pick an appropriate data type and write code to do the above calculation. You should use variables for `r`,`g`,`b` and `y` so that you can modify them to explore the results if different values of `r`,`g`,`b` are used. You may assign your own values to these variables in order to test your program. Try some of those shown in the example output below.

Example outputs to the console (note, each line results from running the program a separate time with a different values assigned for `r`,`g`,`b`)

```
The pixel (r = 24, g = 16, b = 100) has a luminance of (y = 27)
The pixel (r = 150, g = 60, b = 33) has a luminance of (y = 83)
The pixel (r = 250, g = 120, b = 150) has a luminance of (y = 162)
The pixel (r = 255, g = 255, b = 255) has a luminance of (y = 254)
The pixel (r = 0, g = 0, b = 0) has a luminance of (y = 0)
```

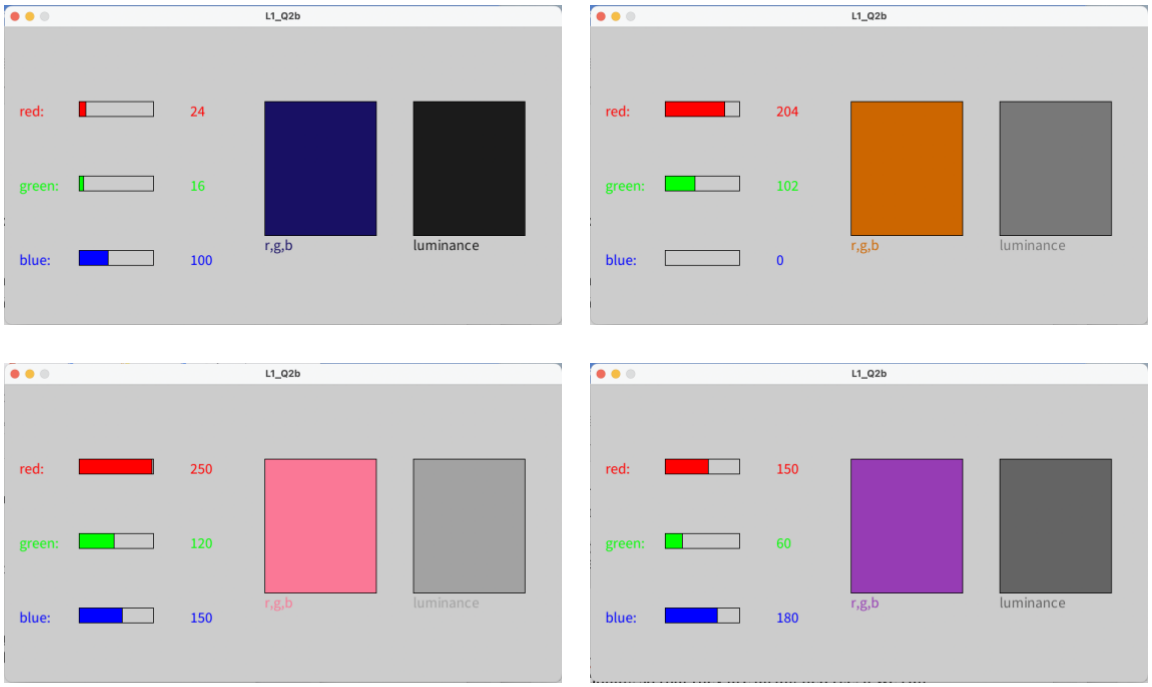
Question 4 (create a visualization of the colour + conversion in Question 3)

In this last question, you will create a visualization of Question 3. There is a creative component involved here – as you will get to design the look and feel (building on last week's drawing commands, however, now we want to leverage variables and computations utilized in Question 3.

Copy the completed code from the main method in your ***lab1_q3.pde*** file, over to your ***labs/lab1_q4/lab1_q4.pde*** file. Then modify it so that it illustrates (visually in an application window), the various values represented in the r,g,b variables (as a percentage/proportion of their maximum value: 255).

You must include text labels of both the variables, their numeric value and a graphic illustrating the ***proportion*** of colour from each variable that makes up the final colour.

You have creative freedom over all of this and how to represent things pictorially. Below is a very mundane (boring) version of what a visualization might look like (ignore the window titles):



Note that the combined colour (according to the r,g,b values) is also visualized, along with the luminance value that corresponds to the conversion.

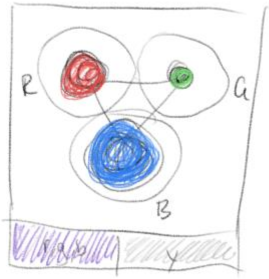
Note – the luminance can be used to generate a colour (as a fill or stroke or text colour) by using (y,y,y) in place of (r,g,b).

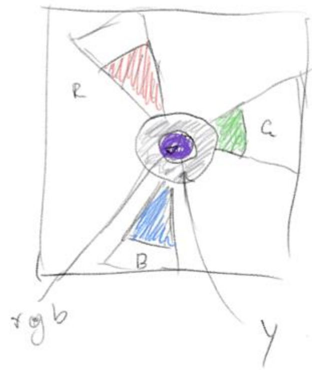
YOUR JOB IN THIS TASK IS TO MAKE A FAR MORE JAZZY VISUALIZATION THAN THE ABOVE!!!

For example, something like the classic Apple watch visualization (for fitness/health apps), might show the colours and their proportions using arcs/rings (below left).

OR,

perhaps think of some other representations (line art/shape combinations – as quickly/roughly sketched below right):





STEP 3: SUBMISSION (DUE: Tuesday Sept 27, 2022 – 5:00pm)

You will formally submit the ***.pde** files associated with lab1 -> Questions01-04

To do this, we need to re-open the terminal (STEP 1d). Once you have navigated to the **labs** subdirectory in which your project sketch folders/files are located, you may submit all your work with the following command (on the lab computer). This will submit the entire lab1 folder and all of its sketches:

```
submit 1710 lab1 lab1
```

Recall: if you submit multiple times, the files previously submitted will be overwritten with the newer versions of the files. You can submit an unlimited number of times up until the deadline for this lab.

To check the files you have currently submitted, type the following on a lab computer:

```
submit –l 1710 lab1
```

You should see something similar to this (in terminal):

```
The following files have been submitted:
lab1_q1                                         25 bytes
lab1_q1/lab1_q1.pde                             1202 bytes
lab1_q2                                         25 bytes
lab1_q2/lab1_q2.pde                             854 bytes
lab1_q3                                         25 bytes
lab1_q3/lab1_q3.pde                             1157 bytes
lab1_q4                                         25 bytes
lab1_q4/lab1_q4.pde                             2294 bytes
```

NOTE: REMEMBER, you can choose to use the web-submit function (see Lab 0 for walkthrough). You will need to find and upload your lab1 files independently. Web-submit can be used to check your submission from the terminal, and can be found at the link: https://webapp.eecs.yorku.ca/submit/