ENGR 102 - Lab Assignment #1

TEXAS A&M UNIVERSITY

Fall 2020

For this course, we will be using Python 3 (Anaconda distribution), and the PyCharm IDE. This week you will install the software, and begin some simple programming.

Activity 1: Getting Set Up (Individual)

☑ *The purpose of this activity is to ensure all students can actively participate in the course.*

Your initial task is to install both Python 3 and the PyCharm IDE on your own laptop if you haven't done so already. Please follow the instructions provided on eCampus for installation and set-up procedures.

Activity 2: Writing Your First Programs (Individual)

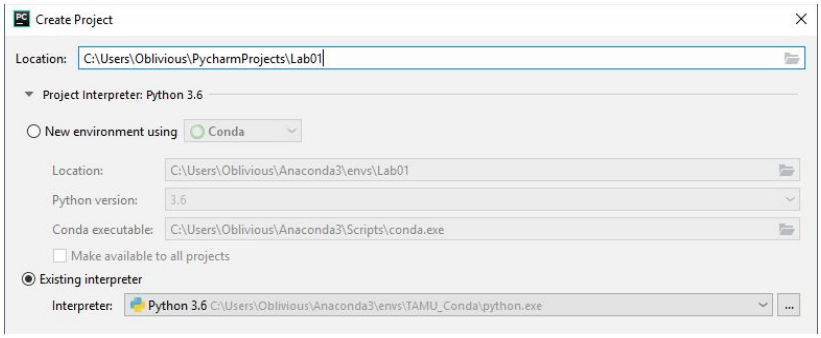
☑ *This activity is designed to verify installation, show how to create and execute a new program, write a print statement, run an import command, use simple math functions, and see what errors may look like.*

"Howdy, World!"

a) From the File menu, choose "New Project" (yes, again)

b) Name the project, perhaps "Lab01"

Important: select the arrow next to Project Interpreter, and choose "Existing interpreter". Select the environment you created during your set-up procedure. This should become the default behavior, and will keep you from unwittingly creating dozens of environments.



c) Create a new Python file: from the file menu, select "New" then "Python File" (give it a name!)
You should now see a blank area labeled with the name you picked, followed by .py

d) In the blank area labeled with the given name, type the command: `print("Howdy, World!")`.
Notice that the IDE makes helpful suggestions, like matching quotation marks and parentheses.

e) From the "Run" menu, select "Run" (it should have a green arrow in front of it).
A new window should appear at the bottom of the PyCharm screen, titled "Run", and the name of your program at the top.

i. This window should show what was executed, the Python interpreter command, and the name of your Python file

ii. Then, you should see the output of your code, the text: `Howdy, World!`

iii. Finally, you should see a line saying something like "`Process finished with exit code 0`" – code `0` is good, it means your program completed without an error.

f) Change the text to print other things, and run it again. *(To run again, click on the green arrow by the run window, or the green arrow at the upper right of the PyCharm environment.)*

Checking NumPy and MatPlotLib

a. Within your current project (not a new project), create another Python File from the file menu. Give it a new name, like "InstallTest".

   You should see a second tab in the main PyCharm window with the name of this new file in it, and a blank screen. *Note: you can switch between these files by clicking the tabs or by following the hierarchy that is usually shown at the left of the PyCharm environment.*

b. Create a simple program, with just two lines:

```
import numpy
import matplotlib
```

c. Run this program* and ensure that it gives a "Process finished with exit code 0" output.

   ** Note: if you just "Run" program again, it will run the Howdy World program from last time! Instead, you need to specify you want to run the InstallTest (or whatever name you picked) Python program. There are a couple of ways to do this:*

   i. *In front of the green arrow at the upper right is a drop-down box. Select the file from the dropdown box, and then press the green arrow. This will execute the selected file.*

   ii. *From the "Run" menu at the top of the screen, pick "Run". When you do this, it will give you options of which file to execute.*

   If it gives an error about a missing module, you either did not successfully add numpy and matplotlib to the environment when creating it (while following the installation and set-up guide), or a different environment was selected for the current project than what had been created.

   The installation and set-up guide can help you add the modules to the environment if you missed that step initially. Once the modules are added through the Anaconda Navigator, the project should work correctly.

   If you selected the wrong environment for the current project, you can change the selected environment as follows:

   i. From the "File" menu, select the "Settings" menu option (in Windows).

      Note: NOT "Settings for New Projects"!

   ii. Expand the "Project: XXXX" option on the left, and select "Project Interpreter"

   iii. Select the correct environment from the drop-down list at the top of the page. If it doesn't show, click on the gear on the far right side of the screen, and select "Show all", and then select the correct environment.

   Call for help if none of this is working.

Math Functions and Errors

a. Create another new Python File

b. Write the following program, with three lines:

```
from math import sin, cos
print(cos(0))
print(sin(0))
```

c. Now, run this new program.

   You should see two lines output, one with 1.0 (the cosine of 0), one with 0.0 (the sine of 0).

d. What is the result if you divide 1 by 0? Modify your previous program by adding the following line *after* the previous lines:

```
print(1/0)
```

   This is the sort of error you will see if you have a bug, like dividing by 0.

e. What happens if the new line comes *before* the others in your program?

f. After experimenting a bit, remove this line, so you again have a working program

Activity 3: Unreadable Clock (Team)

☑ *This activity will have you work as a team, write print statements, and use math functions in Python.*

Background

One "geek" item that you can buy is a wall clock where the hours of the clock are labeled by some sort of mathematical formula (e.g., the number 12 might be printed as `24/2` on the clock face). To see examples of these clocks, perform an internet search for "Math Wall Clock" or similar.

Project

As a team, with 3 other students in your class, write a program as a sequence of print statements. Each statement should contain a mathematical formula. Running your team's program should output the numbers from 1 to 12, in order.

Divide the 12 numbers of the clock among the 4 individuals at the table. Each person should be responsible for coming up with three formulas.

- Let each person verify his or her three formulas independently.
- Have one person merge the results from all team members into a single file. Add a comment for each number indicating which member completed that formula.
- Include the standard header at the beginning of the file, and write each member's name.
- Share the file with all team members. Each member will submit a copy of this teamwork.

Rules

- Formulas should be non-trivial, and you should not zero-out sections of formulas. For example, the following would *not* be good formulas for the number 2:
  a. `1 + 1`
  b. `(giant messy complicated expression here)*0 + 2`
- Formulas should be diverse. That is, do not substantially reuse portions of a formula. For example, while these might be individually acceptable formulas for 4 or 5, they are not appropriate to both be used:
  a. 4: `3**2-(3+2)`
  b. 5: `3**2-(3+1)`
- Among your 12 formulas, use each of the following categories at least once (you may certainly use other functions, or use these more than once):
  a. Each of the built in operations: `+`, `-`, `*`, `/`, `**`, `//`, `%` (7 things to include)
  b. Parentheses to override order of operations (1 thing to include)
  c. Each of the three main trigonometric functions: `sin()`, `cos()`, `tan()` (3 things to include)
  d. Square root and absolute value: `sqrt()` and `fabs()` (2 things to include)
  e. *e* exponentiation and logarithm: `exp()` and either `log()` or `log10()` (2 things to include)
- Each output may be formatted with a decimal ("4.0") or without a decimal ("4").

Example Output:

```
1.0
2
3.0
...(statements for numbers 4 to 12)...
Process finished with exit code 0
```