COM4013 – Introduction to Software Development
Week 9 – Classes and Enums

*Always select "**Enable Editing**" if prompted by Microsoft Word*

Lab Exercises

Always refer to the lecture notes for examples and guidance.
Also look at your previous work, many exercises are similar.

There are two "stages" marked in the lab sheet. Stage 1 is the *absolute minimum* point you should reach. Hopefully, you can reach it in the lab session. Reaching stage 2 suggests you are on target for a top-class mark for the module. Set your goals sensibly - do not just aim for the minimum or you may struggle to pass the module.

I use the term integer, string and float, and Boolean to infer what the input/output looks like. Unlike many other languages Python does not expect a data type before the variable names.

So if I ask that you declare an integer num1 to 1, then do as follows:

```
num1 = 1
```

For declaring a string and assigning the value hello

```
greeting = "hello"
```

For floats we can declare it like this

```
Money = 2.50
```

For Booleans (bool) we can declare it like this

`isHeavy = True` **Exam - More Examples**

Person Main

- Create a new Jupyter Notebook project called **Lab 9 Exercises**. Refer to Week 1's lecture/worksheet/video if you have forgotten how to create a new project/file/program or use the shell code.

- Write some code to assign the values for your name, age, gender, address, phone number, national insurance number and a secret password property to variables

- Print these onto the screen like this:

```
Name:         Umar Arif
Age:          27
Gender:       Male
Address:      68 Lovell Street
Phone Number: 075658633678
NINO:         PZ225683E
Password:     ThoughtsAreDangerous-100%
```

- You'll notice that this requires a lot of effort on your part, if I was to ask you do this for a second, third, or even 100th time, you'd find this a difficult task to complete (even with copy-paste).

- So, let's create a template or `class` for this instead... Since most people share these properties.

Person Class

- In the same cell as above – above main create a CPerson class.

```
class CPerson
```

- This class must contain the same properties as above.
    - Note that properties must have the prefix m for all class member variables (look at the style guide for this).

      ```
      mAge = ???
      ```

    - Also note that the NINO and password must be a private member variables for this exercise.
        - We can make it private by placing two _ before its name, like this (e.g., __mVariableName)

- Write the constructor `__init__` method to set the member variable values.
    - Before you do this, you must write a SetNINO method inside of the class. It must take one string value as a parameter.
        - You will need to call this in your constructor when assigning the value of nino to `self.__mNINO`.

  - You will need to pass the `self` keyword here to get access to the member variables.

- Initialise a `CPerson` object in `main` (below your previous code) and assign it to a variable named after yourself.
  - You must provide all the constructor values you defined in the `__init__` method (except self).

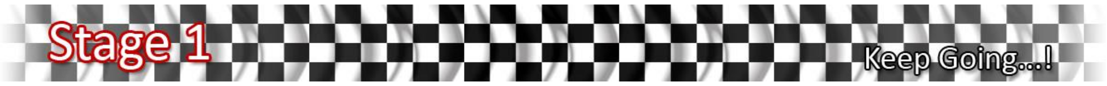
```
yourName = CPerson( comma separated constructor values )
```

- Write an `__str__` method that returns an f string containing the class's member variables in a manner consistent with the earlier approach used in the main function.

- Print the class object out in main and run the cell – You should have two identical prints.

```
print( yourName )
```

- Try to print the value of __mNINO for yourName CPerson object.
  - You should get an error.
  - Write a GetNino function in between the __init__ and the __str__ methods in your CPerson class. This must return the value of the NINO member variable.



People Class Part 2

- In the same cell as above create the following methods.

- Implement setter and getter methods for every member variable within the CPerson class. Adhering to best practices, it is recommended to engage with class variables through the designated interface rather than directly accessing the underlying variables.

- The user must be asked to enter their current password before they can set the value of their password to something else.
  - The user will only be given 3 attempts, before a message telling them that they must contact an administrator is displayed.

  - Use a for in range loop for this and print the number of attempts the user has left after each incorrect input.

- The user must also be asked to enter their password to get the value of their NINO. Implement the same restraints as above.

- Implement a method called GetAgeCategory that returns the age category of the person based on their age (e.g., “Child”, “Teenager”, “Adult”, or “pensioner” etc.).
  - You will need if statements for this.
  - Use the get functions to access the mAge variable in your if statements

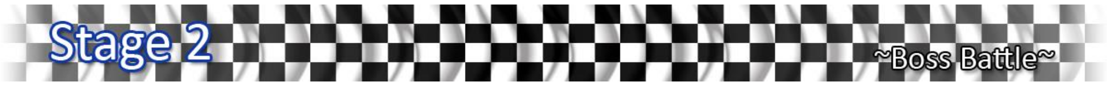
- Implement a method that increments the users age variable and then displays their new age – with a happy birthday message. Call the method Happy Birthday.
  - Create a variable called age and use the GetAge method you implemented earlier to get the current value of age. Increment this by 1.
  - Print the following message in the happy birthday method.

```
Happy Birthday Umar Arif! You are 28 years old!
```

  - Test this method in the main function – call it three times.

```
Happy Birthday Umar Arif! You are 28 years old!
Happy Birthday Umar Arif! You are 29 years old!
Happy Birthday Umar Arif! You are 30 years old!
```

- FULLY COMMENT THIS CLASS – It is good practice to comment classes well – look up examples online to understand the level of commenting required.



Stage 2 ~Boss Battle~

Enum Method

- Look at the slides and create an enum class called AgeCategory above you CPerson class.

- You need 5 values, CHILD, TEENDAGER, ADULT, PENSIONER, UNKNOWN.

- Edit the GetAgeCategory method to return these enum values instead.

- In main print out a statement based on the returned value from this method.
  - Use if statements and the enum values in your conditions i.e., `AgeCategory.ADULT` etc.
  - The output should look like this.

```
You are a teenager
```

- Copy the code from above and replace the enum values i.e., AgeCategory. PENSIONER with the values you set in the enum class i.e., 2.

- Run your code. Notice that you have the same output. Which is easier to read and understand?

Read the Assignment Breif and Ask Questions – I may not be teaching you next semester and you may want to know what's what.

THAT IS ALL FOR NOW

Submission

1. Upload your work to the GitHub classroom – Here
   a. Classroom Link for Week 9