COM4013 – Introduction to Software Development
Week 2 – Variables, Operators and Conditions

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

Lab Exercises

The exercises in this lab follow the topics in the lecture notes. Refer to the lecture notes for examples and guidance.

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. Going beyond stage 2 sets you apart as an expert. Set your goals sensibly - do not just aim for the minimum or you may struggle to pass the module.

Variables Declarations

- For each of the following variable declarations, decide if it is:
  - **Good**: *A valid declaration with a readable variable name (see lecture)*
  - **Poor**: *A valid declaration, but a poor choice of variable name*
  - **Invalid**: *The declaration has an error - it will not compile*

  Write your answers on paper, or into the worksheet and save it later for reference. Check and explain your answers with the tutor.

```
numberStudents
totalCash = 210.50
num = 45.5
myID = "G1423"
myName = "Bob"
1stPlaceScore
```

```
wall_length
wall height
pi = 3.14159
AccountBalance
firstLetter = 'A'
secondLetter = "B"
```

Variables and Operators

- Create a new Jupyter Notebook project called **Lab 2 Exercises**. Refer to last week's worksheet if you have forgotten how to create a new project/file/program.
- Copy the code below into the correct place in the shell code. Again, refer to last week's lecture / lab or ask the tutor if you are not sure where to put the code:

```
itemPrice = float(input("Enter the item price: "))
itemTax = 8.5
totalPrice = itemPrice + itemTax
```

```
print("Tax on the item is", itemTax)
print("Total item price is", total_price)
```

- Run the code. You will immediately see that Jupyter spots an error in the code. Fix the error so that the program will run.
- The program adds a fixed amount of tax to the price, which is not very realistic. Update the value of `itemTax` to be 20% of `itemPrice`. This involves a calculation using addition and division. Try to do this on your own, but if you get stuck there is an example in the lecture notes on the "Updating Variables..." slide.
- After displaying the total value with tax, add some extra code to the program to ask the user "How many items do you want to buy?". Read an integer from the user for their answer, then calculate the total cost (totalPrice * number of items). Output a final message in this style:

  ```
  The price for 4 items is 57.60
  ```

- Add some sensible comments to your code.

An 'if' Statement

- Create a new cell in Jupyter Notebook (use the shell code).
- Copy the code below into the correct place in the shell code:

```
year = int(input("Enter the year you were born: "))

if year % 4 == 0:
      print("You were born in a leap year")
```

- Run the program and it will tell you whether your birth year was a leap year. It works by seeing if there is a 0 remainder after dividing the year by 4, which is the same as saying that the year divides by 4 exactly - the main rule for leap years (there are other rules for leap years, but they do not affect any recent years).
- Add an 'else' statement to display the message "You were not born in a leap year" appropriately. Check the lecture notes to see where to put an else statement. Be precise and try to line things up in the same way as the lecture.
- Add another 'if-else' statement that checks if the user was born in the same year as yourself. It should display "You were born in the same year as me!" or "You weren't born in the same year as me".
- **OPTIONAL:** Can you work out how to update your code to say one of "You're older than me", "You're younger than me" or "You're the same age as me"?

A Common Error: Integer divided by Integer (Dynamic typing to the rescue…)

- Create a new cell in Jupyter Notebook (use the shell code).
- Copy the code below into the correct place in the shell code:

```
half = int(1/2) # Casted to integer yo make a point
print(f"One half (1/2) = {half}")
```

- Run the program. That's strange... it says 1/2 = 0
- Why does this happen? The computer calculates the division 1/2, but as both 1 and 2 are integers (there is no decimal point) it thinks you want an integer as an answer. So, it rounds the result 0.5 down to 0. The computer does this *before* it sees that the result is being put into a float (remember *computers are stupid*).
- Luckily for us if we remove the casting then Python will automatically convert our integers to floating point numbers. So, we do not need to use a decimal point to indicate that our variable is a float.
- Add `print(type(half))` after the half variable. A great way to tell if your code is working as expected. You should see type float.
- To be more precise we should use 1.0 and 2.0 for readability. Edit the programme to be more readable by doing this.

Fahrenheit to Celsius

- Create a new cell in Jupyter Notebook (use the shell code).
- Write a program that asks the user for a temperature in Celsius. Convert this temperature to Fahrenheit using the formula:

  Fahrenheit = Celsius * (9/5) + 32

  Watch the brackets. Display the result as such (25C = 77.00F)

Currency Conversion

- Create a new cell in Jupyter Notebook (use the shell code).
- Write a program that inputs from the user an amount in pounds (£) and converts it to an amount in dollars ($), displaying the result. At first assume that the exchange rate is $1.5 for £1. Check your result carefully and make sure you are converting in the correct direction.
- Update your program so the user can first enter the exchange rate to use instead of just using 1.5.
- Extend the program further by first prompting the user "Type 1 to convert from £ to $, or type 2 to convert from $ to £". Then read the user's choice (as an integer). Use an 'if-else' statement that tests if this integer is equal to 1 as its condition. In the first block of the if statement, put the code to convert from £ to $, and in the second block ('else') add code to convert the other way, from $ to £. The code to convert the other way is the same.

- **OPTIONAL:** Clearly there is a lot of similar code between the if and else blocks here. Try to reduce the amount of duplicate code to the minimum. You can get it down to only one line in each block of the 'if-else' statement, although it will need some thought.

Maximum Number

- Write a program that asks the user to input five different integers. Using 'if' statements work out which is the maximum of the five numbers and display it. You will need several 'if' statements to do this exercise. Ask me for advice if necessary.

Sorting (!)

- Create a new cell in Jupyter Notebook (use the shell code).
- Write another program that asks the user to input five different integers. Using 'if' or 'if-else' statements ***only***, display the numbers in increasing order. E.g., user inputs: 5, 2, 11, 1, 4. Then the program outputs: 1, 2, 4, 5, 11

  This exercise is *very* difficult, but you have already reached the first-class stage for this lab sheet so challenges here are supposed to be very tough. You will gain experience just by thinking about this exercise.

  Possible approach: Compare the 2nd, 3rd, 4th and 5th values in turn with the 1st value. If any value is less than the 1st value, then swap it with the first value. Think carefully how to swap two variables. Now do the same comparing/swapping with values 3,4 & 5 against value 2. Then compare/swap 4, 5 against 3, then finally 5 against 4. There is a pattern there. Ten 'if' statements in total. Good luck!

Submission

1. Save your completed Jupyter Notebook or Python script.
2. Create a GitHub account if you do not already have one.
3. You will receive an invitation to join a GitHub class repository. Once you accept it, create a new repository for your project within that class repository.
4. Upload and share your file with your instructor in your newly created repository for evaluation.
5. Please complete these steps before our next session. If you have not received a GitHub invitation by then, feel free to send me a message on Teams, and I will assist you in getting set up.