COMP1811 Lab 1.02
Python Conditionals

Aim

Practice using conditional statements in programs to make decisions and control execution flow. You will write short programs that can respond to different inputs in different ways and make decisions based on a set of given logic and relational rules.

Tasks

1. Revise the Python Conditionals notebook on Moodle and repeat the activities within until you are competent in coding them.
2. Test your knowledge on Python Conditionals. Complete this quiz individually**.**
3. Complete the Python coding exercises below. Ensure that you have a COMP1811 folder on your University One Drive. Then go to Moodle and download and unzip the code needed for these exercises into your COMP1811 folder in a new subfolder named **Week_2_Conditionals_code**.

Complete lab tasks 1 and 3 in pairs. Attempt all exercises during your lab session and ask your tutor if you are stuck. **Remember you must discuss your work with your tutor before you leave**.

Python Coding Exercises

a. Exercise 1

Examine the script in **02Conditionals.py** and try to understand the basic structure of a two-way conditional statement.

The script distinguishes even and odd numbers entered by the uses. It tests for even numbers by using the modulus operator "%" to calculate the remainder when the number is divided by 2. If the remainder is 0, the number is even.

b. Exercise 2

For each Python script in files prefixed with numbers from **03-05**:

- Go through the script and predict what it does and what the output will be.
- Run the script.
- Check whether you were right.

c. Exercise 3

1) Create a new project in your COMP1811 **Week_2_Conditionals_code** folder and call it **Conditionals**. Then create a Python file in this project called **exercise_3** or **highest_num**.
2) In **highest_num**, write a program to find the greatest of three numbers entered by the user. Remember to comment your code and to use meaningful variable names.

d. Exercise 4

It is not always possible to form a triangle given any three lines! For example, if one of the lines is 9cm long, one of the others is 3cm long and the third is 2cm long, you will not be able to get the edges of all three lines to meet and you cannot therefore form a triangle.

There is a simple test using the Triangle Inequality Theorem to check if it is possible to form a triangle given three line lengths:

- If the sum of any two lengths is always greater than the third, then a triangle can be formed using those lengths. Otherwise, it cannot.
- If the sum of any two lengths equals the third, then a "degenerate" triangle is formed instead.

1) Create a new Python file in the same project above and call it **exercise_4** or **is_triangle**.
2) In **is_triangle**, Write Python code to prompt the user to input the lengths for three lines, and then determine whether those lines can form a triangle. Display the result to the user on the console. Remember to comment your code and to use meaningful variable names.

e. Exercise 5

The Magic Kingdom theme park has some rides that are restricted by age and height. The owner of the park wants to reduce the delays and disappointment caused by park-goers not knowing the restrictions before queuing for the rides. He wants to install software on terminals around the park so that people can find out which rides they may go on and he provides the following information:

Ride name	Age Restrictions	Height Restrictions
1 Carnival Carousel	Min 5 years old	None
2 Dodgems	Min 12 years old	Min 1.3m
3 Pandemonium	Min 16 and Max 70 years old	Min 1.4m and Max 2m
4 Phantom Ghost Zone	Min 8 years old	None
5 Scenic River Cruise	None	None

1) Create a new Python file in the same project above and call it **exercise_5** or **can_ride**.
2) In **can_ride**, write a program that displays available ride names and code numbers as shown in the table above and asks users to enter the code number of the ride they've selected. The program should then ask for their age and height, and accordingly display a message indicating whether or not they can go on their chosen ride. The park owner is happy with text input for now.
   Remember to comment your code and to use meaningful variable names.