Guided Task 2: Selection

Objective

This guided task is made up of two parts.

In the first part, you'll use conditional if...elif control flow statements in Python.

In the second part, you'll create several pieces of code in four distinct exercises. You'll practice what you've learnt and create more complex code.

Duration: 10-15 minutes (Part 1), 90 minutes (Part 2)

Instructions

Part 1

1. Add a new code page to your project. Call it *selection_part1*.
2. Create an IF statement to see if the person's age is equal to 18 or over.
   a. Display '*You are in category A*'
3. Create an IF statement to see if the person's age is equal to 16 or over.
   a. Display *'You are in category B'*
4. Create another IF statement to see if the person is under 16 years of age.
   a. Display *'You are in category C'*
5. Save and run your code and enter 19 for age.
6. As you see, there are too many confusing messages!
   Simple IF statements work fine, but not in a chain of IF statements such as these. For chain, we need to use an if...elif statement.
7. Create an if...elif statement to examine the age in one statement. Follow this pattern:

   ```
   if person is 18 and over:
       display message
   elif person is 16 and over:
       display message
   else:
       display message
   ```

   Note: You must start with the highest age value first.
8. Save and run your code using different values for age.

Part 2

Exercise 1 - Create a calculator

1. Either create a new project or, preferably, add a new code page to an existing project.

   Name the new page *Calculator.py* and make it the startup file.

   If you're creating a new project by selecting the *File-New-Project* menus, make sure you give this new project a suitable name such as 'Chapter 6'. Take note of where it is saved, as you'll take away your work at the end of this course.

2. Input two numbers (int or float, your choice)
   Tip: Use two **input()** statements. Don't forget to cast the text to either **int** or **float**.

3. Display a typical calculator menu such as:

   ```
   1. Add        +
   2. Subtract   -
   3. Multiply   *
   4. Divide     /
   5. Square     s
   ```

   Tip: Use simple **print()** statements.

4. Ask the user to choose what operation to perform.
   For example, if they select **+** then you should display the sum of the two numbers (in Step 2).

   Tip: Use an **input()** statements. There is no need to cast the text.
   You'll need a single **if...elif** statement to examine the operator and display the result.

5. Save and run.

Exercise 2 – Calculate exam grades

In this part of the task, you'll write code to input a grade between 1 and 100 and display the exam grade according to a set of rules.

1. Add a new file to your project called *ExamGrade.py* and make it the startup file.

2. Input the exam mark for a student. The integer must be between 1 and 100. If the mark is less than 1 or greater than 100, you'll display a suitable message (see Step 3).

3. The rules for calculating a grade is as follows:
   If the mark is **less than 1** or **greater than 100**, display 'Error: marks must be

between 1 and 100'.

Less than **50**	Fail
between **50..60** (inclusive)	Pass
between **61..70** (inclusive)	Merit
between **71..100** (inclusive)	Distinction

Exercise 3 – Calculate exam grades with levels

In this part of the task, you'll write code to input a grade and calculate the grade, but this time you'll take into account the different levels of studies.

Tip: You'll need to make use of nested if statements, such as:

```
if(level == 1):
     if (grade >70):
          print(….)
```

You may decide to use separate if statements for levels or use **elif**.

1. Add a new file to your project called *ExamGrade2.py* and make it the startup file.
2. Input the exam mark for a student. The integer must be between 1 and 100. If the mark is less than 1 or greater than 100, you'll display a suitable message.
3. Input the student level. Currently we've got two levels (1 or 2).
4. The rules for calculating a grade for **level 1** are as follows (same as in Part 2):

   If the mark is **less than 1 or greater than 100**, display 'Error: marks must be between 1 and 100'.

Less than **50**	Fail
between **50..60** (inclusive)	Pass
between **61..70** (inclusive)	Merit
between **71..100** (inclusive)	Distinction

   The rules for calculating a grade for **level 2** are as follows:

Less than **40**	Fail
between **40..50** (inclusive)	Pass
between **51..65** (inclusive)	Merit
between **66..100** (inclusive)	Distinction

5. Save and run.

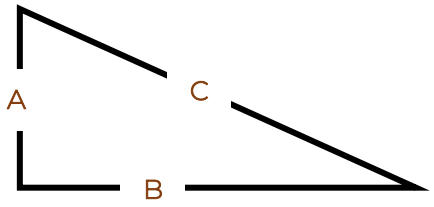
Exercise 4 - 9.10: Pythagoras

In this part, you'll write a program that calculates the lengths of sides of a triangle using Pythagoras's theorem.

Pythagoras' theorem states that the square of the long side (C) of a right-angled triangle is the sum of the squares of the two shorter sides (A and B).

C**2 = A**2 + B**2

*(**2 in Python will raise to power of 2)*



Add a new file: *Pythagoras.py* to your existing project and make it the startup file.

1. Print a menu:

> Pythagoras' Calculator
> 1. Find the length of A given B and C
> 2. Find the length of B given A and C
> 3. Find the length of C given A and B

2. Print the result.
3. If '1' is entered, prompt for the length of sides: B and C, calculate the length of side: A
4. If '2' is entered, prompt for the length of sides: A and C, calculate the length of side: B
5. If '3' is entered, prompt for the length of sides: A and B, calculate the length of side: C

Well done, you've completed this guided task!