# **Chapter 02 Selection Lab**

## **Part 1**

## **Objective**

In this lab you'll gain the experience of using conditional **if…elif** control flow statements in Python.

**Duration:** 10-15 minutes

## **Steps**

1. In your directory PythonArea (or whatever you chose to call it) create a text file called ***selection.py***
2. Create an IF statement to see if the person is equal to 18 or over.
   1. Display '*You are in catergory A*'
3. Create an IF statement to see if the person is 16 or over
   1. Display '*You are in catergory B*'
4. Create another IF statement to see if the person is under 16 years of age.
   1. Display '*You are in catergory 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

### **Use elif**

1. 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.

1. Save and run your code using different values for age.

## **Part 2**

## **Objective**

In this major lab you'll create several pieces of code in 4 distinct tasks. You'll practise what you've learned and create more complex code. Try to complete as many of these tasks as you can.

**Duration**: 90 minutes

### **Task 1 - Create a Calculator**

Create a text file called ***calculator.py***

1. Inputs two numbers (int or float, your choice)  
   **Tip**: Use two **input()** statements. Don't forget to cast the text to either **int** or **float**.
2. Display a typical calculator menu such as  
     
      
     
     
     
     
     
    **Tip**: Use simple **print()** statements
3. **Add +**
4. **Subtract -**
5. **Multiply \***
6. **Divide /**
7. **Square s**
8. Ask the user to choose what operation to perform.   
   For example if they select **+** then you should display the sum of the two numbers.  
      
   **Tip**: Use **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.
9. Save and run.

## **Task 2 – Calculate exam grades**

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

1. Create a text file called e***xamgrade.py***
2. Input the exam mark for a student. The integer must be between 1..100   
   if the mark is less than 1 or greater than 100 you'll display a suitable message (see below)
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..100'

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

## **Task 3 – Calculate exam grades with levels**

In this part 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. Create a text file called ***examgrade2.py***
2. Input the exam mark for a student. The integer must be between 1..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 have two levels (1 or 2).
4. The rules for calculating a grade for **level 1** are as follows (same as in Task 2):

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

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

The rulesfor 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**

1. Save and run.

## **Task 4 - Exercise 9.10 – Pythagoras**

In this lab 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**  
*(n\*\*2 in Python will raise n to power of 2, i.e. square it. e.g. 3\*\*2 is 9.)*

**C**

**A**

**B**

1. Create a text file called ***pythagoras.py***
2. 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
4. Print the result.

If **‘1’** is entered, prompt for the length of sides: **B** and **C**, calculate the length of side: **A**

If **‘2’** is entered, prompt for the length of sides: **A** and **C**, calculate the length of side: **B**

If **‘3’** is entered, prompt for the length of sides: **A** and **B**, calculate the length of side: **C**

\*\* End