



Introduction to programming

Making decisions and repetition - Lab

Exercise 1 – advanced_login.py

In Exercise 2 last week you had to write part of a login program that displayed **True** or **False** depending on whether the user entered a correct password. Revisit this program and update it to use an `if .. else` statement to display an appropriate message if the user enters the correct password. Test that your program produces the correct result and save this as **advanced_login.py** in your Week 3 folder.

Exercise 2 – bus_fares.py

You are working for a company that writes software that takes payments for journeys on local buses. You have been asked to write a section of a program that calculates the correct fare for a traveller based on their age. Your program should output a message to the driver to inform them of the correct fare once they have entered the passenger's age.

The standard bus fare is £2.30

If the passenger is under 18, the fare is half price.

If the passenger is 65 or over, the fare is free.

Write a program that takes the passenger's age as an input and displays the correct fare to the driver. Make sure your program is well commented and save it as **bus_fares.py** in your Week 3 folder. Test your program to ensure all the values are correctly displayed.

Exercise 3 – times_table.py

Write a program that produces the seven times table up to the value of 10. You can format the output however you like for example:

$$1 \times 7 = 7$$

$$2 \times 7 = 14$$

...

$$10 \times 7 = 70$$

To create a maintainable program, you will need to use a **while** or **for** loop to repeatedly print out the lines of the table. Save your file as **times_table.py** in your Week 3 folder.



Extension: asking the user which times table they want to display and also how many lines they want to go up to.

Exercise 4 – exam_results.py

You have been given a list of student's exam results for a particular subject. You have been asked if you can write a simple program that can calculate the average of these exam results and then display the result to the user.

Exam results are as follows:

```
21, 11, 4, 96, 48, 5, 13, 64, 28, 33, 43, 20, 70, 24, 88, 57, 31, 9, 35, 47, 56, 45, 14,
74, 35, 6, 79, 62, 17, 83, 5, 8, 44, 56, 60, 47, 17, 23, 96, 66, 17, 43, 7, 21, 18, 100,
30, 8, 15, 15
```

You will need to loop through the values to calculate the total and then divide this by the number of items in the list. You should find the result gives an average result of 38. Save your file as **exam_results.py** in your Week 4 folder.

Extension task – guessing_game.py

You are working in a team of programmers who are building a set of Python games. You have been asked to write a guessing game that thinks of a random number between 1 and 10 and then asks the user to guess it. If the user is wrong, a message is displayed and the program repeats until a correct answer is given. When a correct answer is given, a success message is displayed and the program finishes.

Put the following snippet of code at the top of your program to generate a random number and store it in a variable called **my_number**

```
import random
my_number = int(random.random() * 10) + 1
```

Test your program to see if it works as expected and save it as **guessing_game.py** in your Week 3 folder.