

## COM4013 – Introduction to Software Development

### Week 11 – Intro to Algorithms in Python

*Always select “Enable Editing” if prompted by Microsoft Word*

#### Lab Exercises

*Always refer to the lecture notes for examples and guidance.  
Also look at your previous work, many exercises are similar.*

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. Set your goals sensibly - do not just aim for the minimum or you may struggle to pass the module.

**I use the term integer, string and float, and Boolean to infer what the input/output looks like. Unlike many other languages Python does not expect a data type before the variable names.**

So if I ask that you declare an integer num1 to value of 1, then do as follows:

```
num1 = 1
```

For declaring a string and assigning the value hello

```
greeting = "hello"
```

For floats we can declare it like this

```
money = 2.50
```

For Booleans (bool) we can declare it like this

```
isHeavy = true
```

## First Program

- Type (do not copy) this code into the IDE

```
def main():  
    a = 4  
    b = 6  
    result = a * b  
  
    print(f"The result is {result}")  
  
if __name__ == "__main__": main()
```

- Now compile and run the program.
- Compilation and debugging are the same in C++ as in the Python environment.
- Write a function that takes in two integer parameters and returns the product (multiplication) of both.

## Programming Exercise 1

Use a for loop to write out the 2 times table. Your program needs to output the following to the screen:

```
1 times 2 = 2  
2 times 2 = 4  
3 times 2 = 6  
4 times 2 = 8  
5 times 2 = 10  
6 times 2 = 12  
7 times 2 = 14  
8 times 2 = 16  
9 times 2 = 18  
10 times 2 = 20  
11 times 2 = 22  
12 times 2 = 24
```

You must use a for loop. You are not allowed to use any other type of loop construct.

## Keyboard input

- In Python you get input from the keyboard using the `input` command.
- Create a string called `name` and ask the user to input their name.
- Print the name to the console using the `print` format statement.

## More Programming Exercises

- Write a program which prompts the user a number and then outputs the square of the number (the square of a number is the number multiplied by itself).
- The length of a string can be found by using the `"len"` method, e.g. if we had a string variable called `"word"` then the length of the string would be given by the following code:

```
len(word)
```

- Write a program which prompts the user for their name and then tells them how many letters there are in their name.



## Drawing Squares

- Write a program that prompts the user for an integer value. Using this value draw a square made out of asterisks the size of the value given, e.g. a value of 1 would give:

```
*
```

- A value of 4 would give:

```
****
****
****
****
```

## Type No to End

- Write a while loop which prints out the message `"Type 'no' to end"`, and then only ends if the user does indeed enter the word `"no"`. Make sure you carefully test your program with both `"no"` and some other words.

### Age If Statements

- In a new cell, use the shell code (from semester 1). Write a program which asks the user for their age. Display a different message depending on whether they are a child, adult or pensioner. You can decide the age ranges.

### Output Number of Characters

- Set up a while loop so that program reads in character values from the keyboard until an asterix is entered. You will need to count the number of characters entered. Do not include the asterix in your count.
- Output the number of characters entered.

### Guess the Secret Number

- Write a program that initialises a "secret" number to a value between 1 and 10. Ask the user to enter a number between 1 and 10 in order to guess what the number is. If the number entered is correct, output "Well done, you guessed it!" or words to that effect. If the guess is incorrect, output "Wrong, try again" or similar. The program should terminate when they guess the number.
- Now modify the program so that it gives the user a clue as to guessing the number by outputting "No it is higher than that" or "No it is lower than that".

### Nested For Loop

- Use a nested loop to create and display the following 3 by 3 matrix  
1 2 3  
4 5 6  
7 8 9

### Programming Exercise 1

Read integer values from the keyboard values until you encounter a zero. Print out the sum (an integer) and the average (a float).

### Programming Exercise 2

Write a program (using a while loop) that repeatedly requests two numbers to be entered by the user and then adds the numbers together. If, at any time, the sum of the numbers is greater than 100 then the program stops.

### Programming Exercise 3: Bicycle Gears

The gear for a pedal cycle is calculated as:

$$\text{wheel radius} * \text{teeth on front sprocket} / \text{teeth on rear sprocket}$$

Write a program to read 3 floating point numbers:

the wheel radius

the number of teeth on the front sprocket

the number of teeth on the rear sprocket

Then calculate and subsequently print to screen the gear.

Example input: 27.00 52 22

Calculation:  $\text{gear} = 27.00 * 52.00 / 22.00$

Example output: gear is 63.8182 inches

Although the last two numbers are read in as integers you will need to do a floating point calculation. Why? What happens if the calculation is an integer calculation?

#### **Programming Exercise 4: larger and lowest value**

Write a program that asks the user for five numbers. Tell the user what the highest number was and what the lowest number was.

---

### **Advanced Tasks**

#### **Programming Exercise 5**

A builder needs to measure the height of a wall. Unfortunately, they have left their ladders at home. Luckily, they do have a long tape measure and a long piece of wood. The wood is always guaranteed to be longer than the height of the wall.

The length of the sides of a right-angled triangle are given using the equation:  $a^2 = b^2 + c^2$ , where a, b and c are the three lengths of the triangle. How would the builder calculate the height of the wall?.

Test with:

Distance from wall to wood=5m Length of wood=13m

This should give you a height of the wall as 12m

#### **Programming Exercise 2**

Write a program that will generate tables question such as:

What is 5 x 7?

If the user enters 35 they are told they are correct otherwise they are incorrect. The user should be given the option to “Answer another question? (Y/N)”. If ‘Y’ is entered, the process should be repeated.

You will need a way to generate random numbers. The function given below will generate a random number between



### Advanced Tasks

- Here are some tasks for those who want to stretch themselves.

#### Cube of a Number

- Obtaining the cube of a number. Write a function called cubed which takes a single integer as a parameter and then returns the cube ( $n * n * n$ ) of the number.

#### Min Function

- Write a function which, given two integers, discovers the smaller of the two integers and then returns this value.

#### Max Function

- Write a function which will return the maximum of three given integer values.

#### Area of a Circle

- Calculating the area of a circle. The area of a circle is calculated using pi (3.14) times the square of the radius:

$$\text{area} = 3.14 * \text{radius} * \text{radius}$$

- Write a function which calculates the area of a circle. The function takes one parameter of type float and returns a value of type float.
- Include some error checking (try-catch not try-except) in your code. If the value of the variable passed over to the function is negative, then produce an error message and return a value of -1.

## **Advanced Task – (PORTFOLIO PIECE)**

### **Calculator Tool**

- Throughout the previous semester, we extensively covered the topic of calculators. As a continuation of our exploration, I task you with creating a sophisticated text-based scientific calculator.
- Start by incorporating the fundamental operations such as addition (+), subtraction (-), multiplication (\*), division (/), and modulus (%).
- Then incorporate advanced functions like factorials and exponentiation (squared, cubed).
- Feel free to organise it with a calculator class, a user-friendly menu system for efficiency, and some level of input validation.
- Draw inspiration from your PC's scientific calculator for additional features.

**THAT IS ALL FOR NOW**