Exercises – Getting Hands On With Python

Exercise 1

For this first exercise, you are tasked with creating the famous 'Hello World' application.

```
print('Hello World')
```

Instead of just writing a print statement and passing it one argument (the string 'Hello World') as we did above, please try to replicate this output in the following three different ways:

1. Concatenate the following three strings together using the + operator
    a. 'Hello'
    b. ' '
    c. 'World'
2. Store the string 'Hello' in a variable called greeting. This can then be concatenated to ' World' using the + operator
3. Pass two separate arguments to the print function:
    a. 'Hello'
    b. 'World'

Exercise 2

You are creating an Electronic Medical Records application and you are currently working on the personal details portion of the app.

You need to create the following variables. Please input data that you feel is appropriate for these variables. e.g. first_name = "Geoff" and so on…

- first_name
- last_name
- age
- gender
- weight

- bmi
- diagnoses
- medications
- next_of_kin
- has_medical_card
- contact_number

After storing some data in the above variables, you can paste in the following print statement:

print(type(first_name), type(last_name), type(age), type(gender), type(weight), type(bmi), type(diagnoses), type(medications), type(next_of_kin), type(has_medical_card), type(contact_number), sep="\n")

This print statement uses the type function to return the data types of each of these variables.

Sep=”\n” allows us to improve the presentation, by separating each printed argument with a new line. Without this additional parameter, the arguments would be printed separated by a single space.

The output that we are looking for is displayed below. If your output matches this output, then you have stored the correct type of data in the variables:

```
<class 'str'>
<class 'str'>
<class 'int'>
<class 'str'>
<class 'float'>
<class 'float'>
<class 'list'>
<class 'list'>
<class 'str'>
<class 'bool'>
<class 'str'>
```