

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

---

## 1.1P: Preparing for Object Oriented Programming

---

PDF generated at 14:29 on Monday 6<sup>th</sup> March, 2023

# 1.1P: Preparing for OOP – Answer Sheet

1. Explain the following terminal instructions:
  - a. `cd`: `cd` stands for “change directory”. It allows you to change your file directory/location by stating “`cd`” and the file location.
  - b. `ls`: `ls` means list files, it allows you to see all the files the directory you are currently in
  - c. `pwd`: `pwd` stands for “print working directory”, and it tells you the current directory you are in.
2. Consider the following kinds of information, and suggest the most appropriate data type to store or represent each:

Information	Suggested Data Type
A person's name	string
A person's age in years	int
A phone number	int
A temperature in Celsius	double
The average age of a group of people	double
Whether a person has eaten lunch	bool

3. Aside from the examples already provided in question 2, come up with an example of information that could be stored as:

Data type	Suggested Information
String	Names of restaurants
Integer	Number of Redbull's consumed for exams
Float	The average exam scores
Boolean	Whether someone has worked out today

4. Fill out the last two columns of the following table, evaluating the value of each expression and identifying the data type the value is most likely to be:

Expression	Given	Value	Data Type
6		6	int
True		True	Bool
a	a = 2.5	2.5	float
1 + 2 * 3		7	int
a and False	a = True	False	Bool
a or False	a = True	True	Bool
a + b	a = 1 b = 2	3	int
2 * a	a = 3	6	int
a * 2 + b	a = 2.5 b = 2	7	int
a + 2 * b	a = 2.5 b = 2	6.5	float
(a + b) * c	a = 1 b = 1 c = 5	10	int
"Fred" + " Smith"		"Fred Smith"	String
a + " Smith"	a = "Wilma"	"Wilma Smith"	String

5. Using an example, explain the difference between **declaring** and **initialising** a variable.

The difference between the two is....<finish the sentence>

#### **Declaring a variable**

Declaring a variable is when you state that a variable of certain data type exists, but it has no value yet, memory is allocated at a specific location for the variable but it doesn't hold any value yet.

**An example of this in C# is string name;**

## Initialising a variable

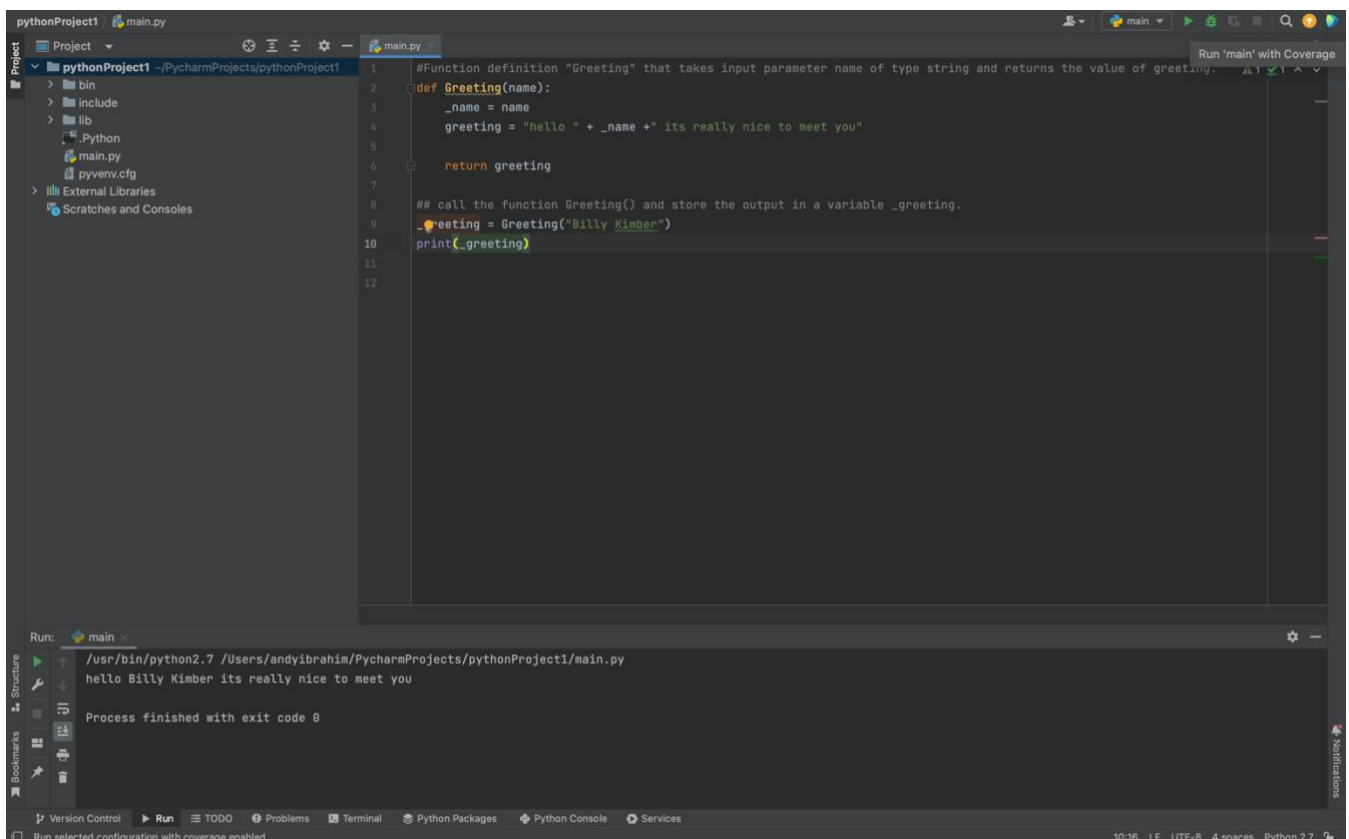
Initialising a variable is when a value has been allocated to the variable after its been declared (or memory being allocated).

**An example of this in C# is string name = "Johnny Walker";**

Now the variable has been initialised and holds a single value.

6. Explain the term **parameter**. Write some code that demonstrates a simple of use of a parameter. You should show a procedure or function that uses a parameter, and how you would call that procedure or function.

A parameter is...<When a function is declared, it may need to take some arguments as inputs for the function to compute what its intended to do. Those inputs for the function are referred to as parameters. They are a set of values that a function/method or process takes to complete the designated task. This process of a function having parameters is generalised so that the parameters can hold any value relative to what the function is going to compute, in saying that, even though the parameters are generalised, the domain or set of values the function can take is restricted. A function that takes a person's name as input and prints out a greeting with their name cannot take an integer value, so any integers are forbidden from being inputted cause it wouldn't make any sense. This is declared in the function definition >



The screenshot shows the PyCharm IDE interface. The main editor window displays a Python script named `main.py` with the following code:

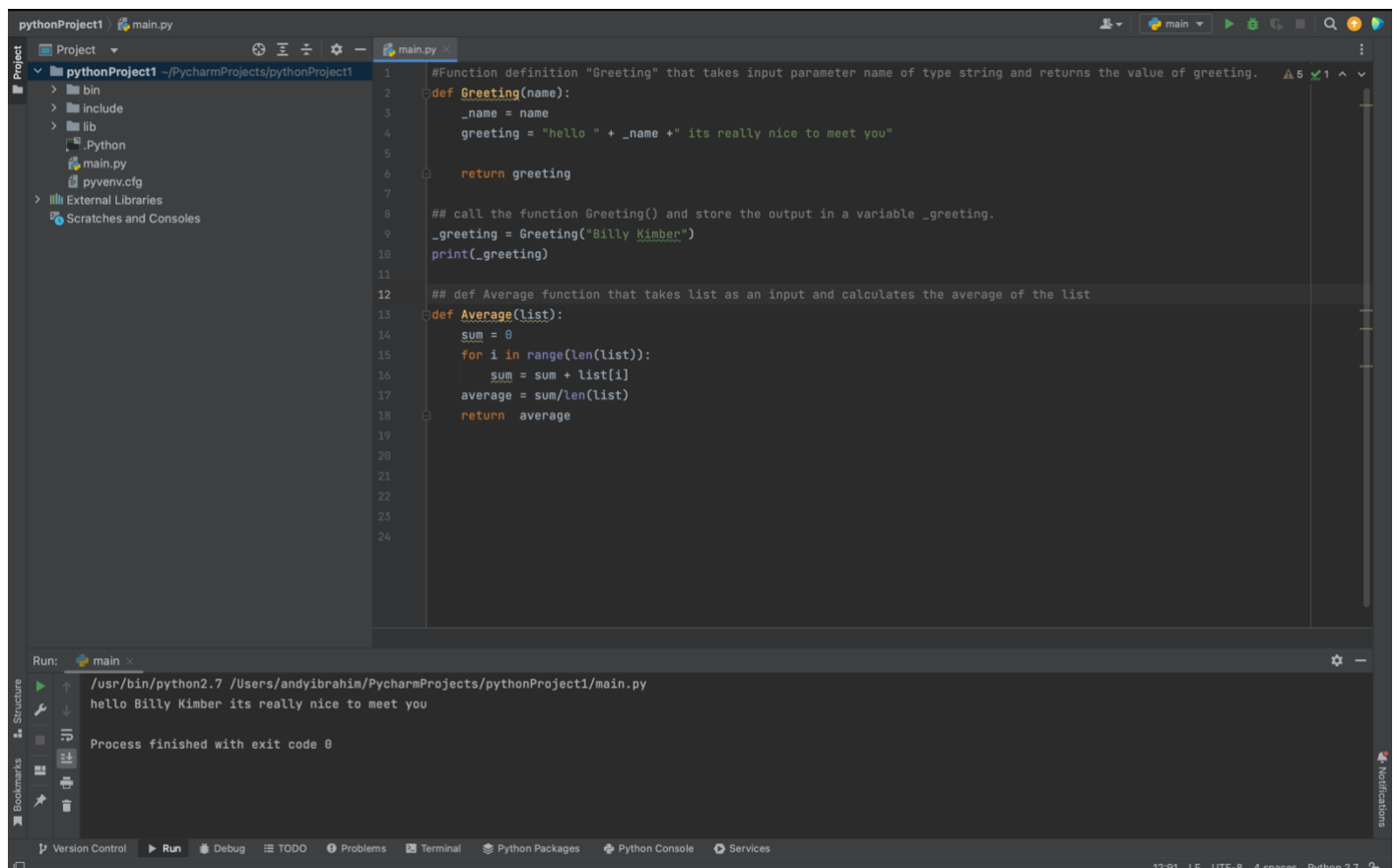
```
1 #Function definition "Greeting" that takes input parameter name of type string and returns the value of greeting.
2 def Greeting(name):
3     _name = name
4     greeting = "hello " + _name + " its really nice to meet you"
5
6     return greeting
7
8 ## call the function Greeting() and store the output in a variable _greeting.
9 _greeting = Greeting("Billy Kimber")
10 print(_greeting)
11
12
```

The left sidebar shows the project structure for `pythonProject1`, including `bin`, `include`, `lib`, `.Python`, `main.py`, and `pyenv.cfg`. The bottom panel shows the output of running the script, displaying the text `hello Billy Kimber its really nice to meet you` and indicating that the process finished with exit code 0.

- Using an example, describe the term **scope** as it is used in procedural programming (not in business or project management). Make sure you explain the different kinds of scope.

Scope is...<When declaring variable or function names, the scope refers to the parts of the program where the function or variable names are valid and can be referred to. There are various scopes but the main ones are global and local scopes. A global scope is when a variable or function name is declared at a location where the whole program has access to, this location is usually outside functions/methods etc. A local scope is when a variable or function is defined inside a function/method, thus it can only be accessed within the function/method and cannot be accessed outside the function.>

- In a procedural style, in any language you like, write a function called Average, which accepts an array of integers and returns the average of those integers. Do not use any libraries for calculating the average. You must demonstrate appropriate use of parameters, returning and assigning values, and use of a loop. Note — just write the function at this point, we'll use it in the next task. You shouldn't have a complete program or even code that outputs anything yet at the end of this question.



```
pythonProject1 main.py
Project
  pythonProject1 ~/PycharmProjects/pythonProject1
    bin
    include
    lib
    Python
    main.py
    pyvenv.cfg
    External Libraries
    Scratches and Consoles

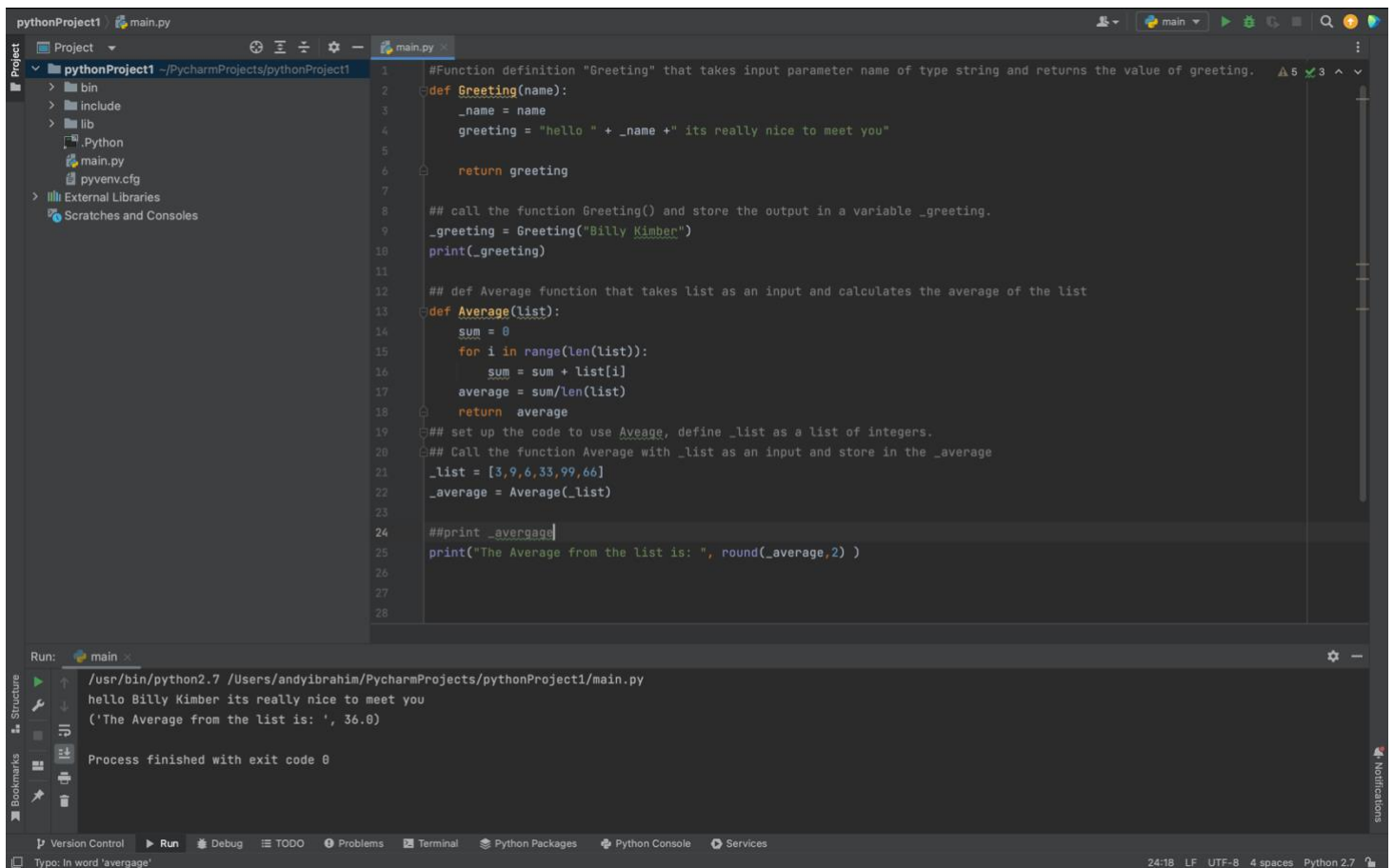
1  #Function definition "Greeting" that takes input parameter name of type string and returns the value of greeting.
2  def Greeting(name):
3      _name = name
4      greeting = "hello " + _name + " its really nice to meet you"
5
6      return greeting
7
8  ## call the function Greeting() and store the output in a variable _greeting.
9  _greeting = Greeting("Billy Kimber")
10 print(_greeting)
11
12 ## def Average function that takes list as an input and calculates the average of the list
13 def Average(list):
14     sum = 0
15     for i in range(len(list)):
16         sum = sum + list[i]
17     average = sum/len(list)
18     return average
19
20
21
22
23
24

Run: main
/usr/bin/python2.7 /Users/andyibrahim/PycharmProjects/pythonProject1/main.py
hello Billy Kimber its really nice to meet you

Process finished with exit code 0

Version Control Run Debug TODO Problems Terminal Python Packages Python Console Services
12:01 LF UTF-8 4 spaces Python 2.7
```

9. In the same language, write the code you would need to call that function and print out the result.



The screenshot shows the PyCharm IDE with a Python project named 'pythonProject1'. The main file 'main.py' contains the following code:

```
1 #Function definition "Greeting" that takes input parameter name of type string and returns the value of greeting.
2 def Greeting(name):
3     _name = name
4     greeting = "hello " + _name + " its really nice to meet you"
5
6     return greeting
7
8 ## call the function Greeting() and store the output in a variable _greeting.
9 _greeting = Greeting("Billy Kimber")
10 print(_greeting)
11
12 ## def Average function that takes list as an input and calculates the average of the list
13 def Average(list):
14     sum = 0
15     for i in range(len(list)):
16         sum = sum + list[i]
17     average = sum/len(list)
18     return average
19
20 ## set up the code to use Average, define _list as a list of integers.
21 ## Call the function Average with _list as an input and store in the _average
22 _list = [3,9,6,33,99,66]
23 _average = Average(_list)
24
25 ##print _average
26 print("The Average from the list is: ", round(_average,2) )
27
28
```

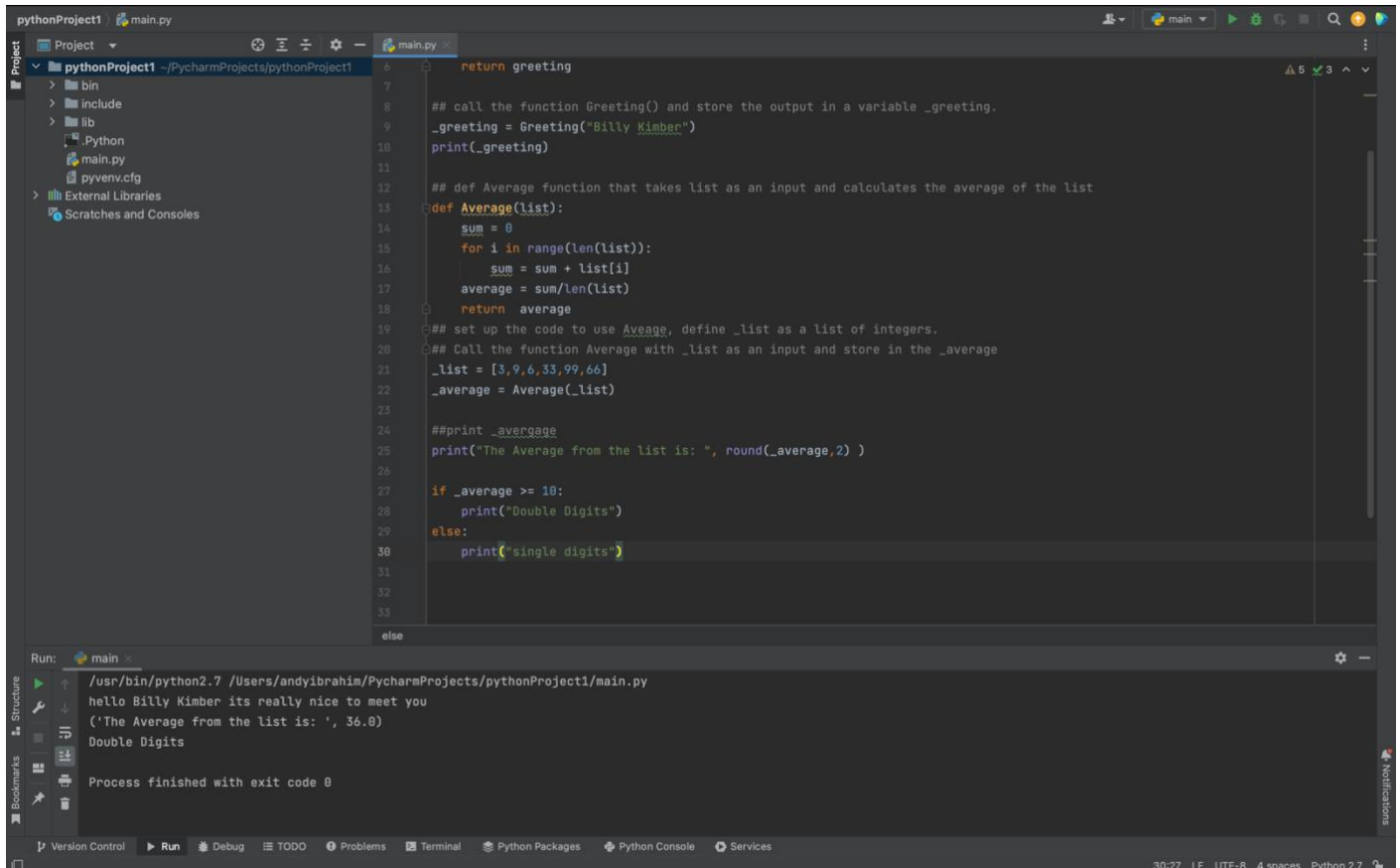
The Run window shows the execution output:

```
/usr/bin/python2.7 /Users/andyibrahim/PycharmProjects/pythonProject1/main.py
hello Billy Kimber its really nice to meet you
('The Average from the list is: ', 36.0)

Process finished with exit code 0
```

The status bar at the bottom indicates the file is 'main.py', the encoding is 'UTF-8', and the Python version is 'Python 2.7'.

10. To the code from 9, add code to print the message “Double digits” if the average is above or equal to 10. Otherwise, print the message “Single digits”. Provide a screenshot of your program running.



The screenshot displays the PyCharm IDE interface. The main editor window shows a Python script named `main.py` with the following code:

```
6     return greeting
7
8     ## call the function Greeting() and store the output in a variable _greeting.
9     _greeting = Greeting("Billy Kimber")
10    print(_greeting)
11
12    ## def Average function that takes list as an input and calculates the average of the list
13    def Average(list):
14        sum = 0
15        for i in range(len(list)):
16            sum = sum + list[i]
17        average = sum/len(list)
18        return average
19
20    ## set up the code to use Average, define _list as a list of integers.
21    ## Call the function Average with _list as an input and store in the _average
22    _list = [3,9,6,33,99,66]
23    _average = Average(_list)
24
25    ##print _average
26    print("The Average from the list is: ", round(_average,2) )
27
28    if _average >= 10:
29        print("Double Digits")
30    else:
31        print("single digits")
32
33    else
```

The Run window at the bottom shows the output of the program:

```
/usr/bin/python2.7 /Users/andyibrahim/PycharmProjects/pythonProject1/main.py
hello Billy Kimber its really nice to meet you
('The Average from the list is: ', 36.0)
Double Digits
Process finished with exit code 0
```

The status bar at the bottom indicates the file encoding is UTF-8, 4 spaces, and Python 2.7.