1.1P: Preparing for OOP – Answer Sheet

1. Explain the following terminal instructions:
    a. cd: stands for “change directory” that allows us to navigate between directories in the desktop such as go back to previous folder or go to the order shown in the PATH

For example:

```
(base) Luans-MacBook-Pro:~ luannguyen$ cd Desktop
(base) Luans-MacBook-Pro:Desktop luannguyen$ ls
```

    b. ls: show the list of files that is included in the current file

```
(base) Luans-MacBook-Pro:~ luannguyen$ ls
Applications                    Projects
Cisco Packet Tracer 8.1.1       Public
Creative Cloud Files            Server-file
Definition.docs.docx            Untitled-1.php
Desktop                         VS CODES
Documents                       bin
Downloads                       dotnet
Library                         lib
Movies                          logs
Music                           luannguyen.csproj
One Drive                       myeasylog.log
P4DS4D2                         obj
Pictures                        opt
Program.cs                      ~.bash_profile.save
```

    c. pwd: stands for “print working directory” that will show the directory(Path) of the current file

```
(base) Luans-MacBook-Pro:Desktop luannguyen$ pwd
/Users/luannguyen/Desktop
```

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	FLOAT
The average age of a group of people	FLOAT

Commented [NN1]: fixed

Whether a person has eaten lunch	BOOLEAN

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	A text message to a friend
Integer	The number of student in a classroom
Float	The pi number
Boolean	Whether a person have pet

Commented [NN2]: fixed

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	Boolean
`a`	`a = 2.5`	2.5	Float
`1 + 2 * 3`		7	Integer
`a and False`	`a = True`	False	Boolean
`a or False`	`a = True`	True	Boolean
`a + b`	`a = 1` `b = 2`	3	Integer
`2 * a`	`a = 3`	6	Integer
`a * 2 + b`	`a = 2.5` `b = 2`	7	Float
`a + 2 * b`	`a = 2.5` `b = 2`	6.5	Float

`(a + b) * c`	`a = 1`
	`b = 1`
	`c = 5`
		10	Integer

`"Fred" + " Smith"`		Fred Smith	String

Commented [NN3]: fixed

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 that declaring a variable means defining the variable's data type and name so that it can be used in the program. When you declare a variable, you are telling the program that a particular variable of a specific data type exists and can be used to store data. And then when initialising a variable, you just give it a initial value before programming

Commented [NN4]: added

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 a special type of variable that is used in the definition of a function or method to represent the input that the function or method expects to receive.

```
def sumwith9(a):
    print(a+9)


sumwith9(3)
sumwith9(12)
```

7. 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.

The scope is the visibility and accessibility of variables and other program elements within a specific part of the program.

Local variables exist inside a block of code or a function that will no longer be accessible when the function is finished executing.

Global variables exist outside the blocks of code and can be accessed by every single function that is existed and will no longer be accessible when the program is ended. For example

```python
#Global variable created outside main()
a = 3
def main():
#Local variable created in main()
    b = 1
    print(a + b)
main()
```

8. 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.

```python
def average(arr):
    if len(arr) == 0:
        return 0
    total = 0
    for num in arr:
        total += num
    return total / len(arr)
```

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

```python
def main():
    x = input('Input an array:')
    arr = list(map(float, x.split()))
    avg = average(arr)
    print('The result is:', avg)
def average(arr):
    if len(arr) == 0:
        return 0
    total = 0
    for num in arr:
        total += num
    return total / len(arr)


main()
```

```
Input the array of numbers that you want to count the average (separate the numbers by spa
ces): 3 4 5
The result is: 4.0
(base) luannguyen@Luans-MacBook-Pro 1.1P %
```

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.

```python
def main():
    x = input('Input an array:')
    arr = list(map(float, x.split()))
    avg = average(arr)
    compare(avg)
def average(arr):
    if len(arr) == 0:
        return 0
    total = 0
    for num in arr:
        total += num
    return total / len(arr)
def compare(x):
    if x < 10:
        print('Single digit')
    else:
        print('Double digit')

main()
```

```
 /usr/bin/env /usr/bin/python3 /Users/luannguyen/.vscode/extensions/ms-python.python-2
20.2/pythonFiles/lib/python/debugpy/adapter/../../debugpy/launcher 62661 -- /Users/lua
yen/Desktop/COS20007-2/1.1P/Average
Input an array:5 6 7
Single digit
(base) luannguyen@Luans-MacBook-Pro 1.1P %
```