Luong Trac Duc Anh – ID: 103488117

COS10009: Pass Task 1

1. In the terminal, we have some commonly used commands to navigate around file folders, some of which are:

- cd (change directory): change the directory to either the child folders or <cd ..> to return the directory to the parent folder (as screenshotted below)

```
PS C:\Users\Pro> cd .\Documents\
PS C:\Users\Pro\Documents> cd ..
PS C:\Users\Pro> D
```

- ls (list): list out files and folders within the directory that you are at (as screenshotted below).

```
PS C:\Users\Pro> ls


    Directory: C:\Users\Pro


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         4/11/2022  12:38 AM                .astropy
d-----         11/6/2021   1:10 PM                .cache
d-----         9/11/2021  10:57 AM                .cisco
d-----         4/19/2022   9:31 PM                .conda
d-----          5/4/2022  11:58 PM                .config
d-----         4/10/2022   4:20 PM                .continuum
d-----          5/3/2022  12:40 AM                .dotnet
d-----         4/10/2022   3:41 PM                .eclipse
d-----         1/29/2022  10:14 AM                .gem
d-----         4/11/2022  12:34 AM                .idlerc
d-----         4/12/2022  10:20 PM                .ipython
d-----         4/11/2022   1:14 AM                .jupyter
d-----          4/5/2022   5:27 PM                .keras
d-----         4/11/2022  12:43 AM                .matplotlib
d-----          5/3/2022  12:40 AM                .nuget
d-----         4/20/2022   1:15 PM                .p2
d-----          5/3/2022  12:26 AM                .splashkit
d-----         4/19/2022   9:27 PM                .spyder-py3
```

- pwd (print working directory): print out the current directory that you are at for easier navigation throughout the terminal (as screenshotted below).

```
PS C:\Users\Pro> pwd

Path
----
C:\Users\Pro
```

2. The most appropriate data type to store and represent the given information is:

- A person's name: string
- A person's age in years: integer

- A phone number: string (for example, +84 393563321 contains the character +, which cannot be stored as an integer)
- A temperature in Celsius: float
- The average age of a group of people: float
- Whether a person has eaten lunch: Boolean (True/False)

3. An example of information that could be stored as:

- A string data type: A person's gender
- An integer data type: The number of products a person purchased.
- A float data type: The height of a person in meters (1.7).
- A Boolean data type: Whether a student has enrolled in the class or not.

4. Fill out the following table:

Expression	**Given**	**Value**	**Data Type**
5		5	integer
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 and b = 2	3	integer
2 * a	a = 3	6	integer
a * 2 + b	a = 1.5 and b = 2	5	integer
a + 2 * b	a = 1.5 and b = 2	5.5	float
(a + b) * c	a = 1, b = 1, and c = 5	10	integer
"Fred" + " Smith"		Fred Smith	string
a + " Smith"	a = "Wilma"	Wilma Smith	string

5. Difference between declaring and initializing a variable:

- Declaration of a variable in a computer programming language is a statement used to specify the variable name and its data type. Declaration tells the compiler about the existence of an entity in the program and its location. When you declare a variable, you should also initialize it.
- Initialization is the process of assigning a value to the Variable. Every programming language has its own method of initializing the variable. If the value is not assigned to the Variable, then the process is only called a Declaration.

Example:

```
int a, b, c;                // declare 3 variables.

int z = 35;                 // declare and initialize variable z with value 35.

double pi = 3.14159;        // declare an approximation of pi.
```

```
char x, y = ‘y’, z;          // declare 3 variables, initialize char y with value ‘y’.
char str[30];                // declare an array named ‘str’ which holds 30 characters.
```

6. A parameter is a named variable passed into a function. Parameter variables are used to import arguments into functions.

Example in Python:

```
def greet_user(first_name, last_name): # first_name and last_name are parameters
    print(f'Hi {first_name} {last_name}!')
print("Start")
greet_user("John", "Smith") # "John" and "Smith" are arguments
```

7. Example of scope:

```
def print_name():
    name = "Duc Anh"
    print(name) # print command number 1
print_name()
print(name) # print command number 2
```

If you run the above code with the print command number 1, the code will be executed successfully. In this case, the print command is within the same scope as the string name, both exist in the function print_name(). However, if we remove the print command number 1 and write the print command number 2, the program will display an error: “NameError: name 'name' is not defined”. This is because the scope of the string name does not exist outside of the print_name() function, and the program cannot access it unless we use a parameter to pass the value in like the number 6 question.

8. Code:

```
# write a function called Average, which accepts an array of integers and returns the average of those integers:
def Average():
    number = int(input("Enter a number: "))
    numbers = []
```

```
    while number != 0:
        numbers.append(number)
        number = int(input("Enter a number: "))
    average = sum(numbers) / len(numbers)
    return average
```

9. Code:

```
def Average():
    number = int(input("Enter a number: "))
    numbers = []
    while number != 0:
        numbers.append(number)
        number = int(input("Enter a number: "))
    average = sum(numbers) / len(numbers)
    return average
# call the function and print out the result:
print(Average())
```

10. Code:

```
def Average():
    number = int(input("Enter a number: "))
    numbers = []
    while number != 0:
        numbers.append(number)
        number = int(input("Enter a number: "))
    average = sum(numbers) / len(numbers)
    # print the message “Double digits” if the average is above 10. Otherwise, print the message “Single digits”:
    if average >= 10:
```

```
        print("Double digits")
    else:
        print("Single digits")
    return average
print(Average())
```