1.1P: Preparing for OOP – Answer Sheet

1. Explain the following terminal instructions:
    a. cd: Change directory
    b. ls: Lists files and directories
    c. pwd: Show current directory

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	Integer
A phone number	String
A temperature in Celsius	Float
The average age of a group of people	Float
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 person's email
Integer	Number of person in a room
Float	A person's weight in kilogram
Boolean	Whether a task has been completed

4. Fill out 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	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` `b = 2`	3	Integer
`2 * a`	`a = 3`	6	Integer
`a * 2 + b`	`a = 2.5` `b = 2`	7.0	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
`a + " Smith"`	`a = "Wilma"`	Wilma Smith	String

5. Using an example, explain the difference between **declaring** and **initialising** a variable. All code are written in Swift

```
var myAge: Integer // Declaration
myAge = 19 // Initialization
```

The difference between the two is declaring is the declaration of the creation of variables while initialization assigns initial values to variables.

6. Explain the term **parameter**. Write some code that demonstrates a simple 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 kind of variable that is used to pass information between functions or procedures.

```
func hello(name: String) -> String {
    return "Hello \(name)!"
}
print(hello(name: "Hoang"))
```

7. Using an example, describe the term **scope**.

   Scope refers to the visibility and accessibility of variables, functions, and other identifiers within a specific region of code. It determines where in the program a variable or identifier can be accessed and used.

```
func number() {
    let x = 5
    print(sum)
}

number()
// Output: 15

print(x)
// Since x is defined within the function's scope, it
cannot be accessed outside the function
```

8. In any procedural 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.

```
func average(_ numbers: [Int]) -> Double {
    var sum = 0
    for number in numbers {
        sum += number
    }

    let count = numbers.count
    if count > 0 {
```

```
            return Double(sum) / Double(count)
        } else {
            return 0
        }
    }
```

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

```
let numbers = [2, 4, 6, 8, 10]
let result = average(numbers)
print("The average is: \(result)")
```

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.

```
if result >= 10 {
    print("The average is double digits")
} else {
    print("The average is single digits")
}
```