1.1P: Preparing for OOP – Answer Sheet

1. Explain the following terminal instructions:
    a. cd: Change Directory
    b. ls: Lists all the files in the current directory
    c. pwd: prints the working directory (aka current file path)

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 (could be float)
A phone number	Int
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	Car manufacturer
Integer	Student ID
Float	Velocity
Boolean	Has student passed unit?

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	Datatype
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 = false	False	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
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 When a variable is declared it becomes known to the compiler and memory is allocated, a variable does not become initialized until there's a value associated with it.

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.

Parameters are variables which are passed into a function and allow functions to access data they otherwise wouldn't be able to.

```
def swap(a, b)
  c = a
  a = b
  b = c

  puts "In swap a is: " + a.to_s
  puts "In swap b is: " + b.to_s

end

def main

  a = 100
  b = 50

  puts "a is: " + a.to_s
  puts "b is: " + b.to_s

  swap(a, b)

  puts "a is: " + a.to_s
  puts "b is: " + b.to_s

end

main
```

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

The scope of a variable refers to where in the code that variable can be accessed. In Ruby, for example, variables can have global scope,

```
$$globalVariable1
```

Variables can also have a local scope in which case they must be passed through as parameters in order to be access by other functions.

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.

```python
# Thomas Horsley - 103071494

def average():
    floatArray = []
    array = input('Please input array: ')
    array = array.split(',')

    i = 0
    total = 0

    for x in array:
        x = float(x)
        floatArray.append(x)
    length = len(floatArray)

    while i < length:
        total = total + floatArray[i]
        i += 1

    average = total/length
    return average

print(average())
```

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

   See 10

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
# Thomas Horsley - 103071494

def average():
    floatArray = []
    array = input('Please input array: ')
    array = array.split(',')

    i = 0
    total = 0

    for x in array:
        x = float(x)
        floatArray.append(x)
    length = len(floatArray)

    while i < length:
        total = total + floatArray[i]
        i += 1

    average = total/length

    if average >= 10:
        print('Double digits')
    else:
        print('Single digit')

    return average

print(average())
```

```
Please input array: 8,2,3,6
Single digit
4.75
```

```
Please input array: 12,53,76
Double digits
47.0
```