

1.1P: Preparing for OOP – Answer Sheet

1. Explain the following terminal instructions:
 - a. `cd`: Changes the current directory.
 - b. `ls`: Lists all the files and folders in the current directory.
 - c. `pwd`: Displays the complete path to the 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	int
A phone number	string (there might be leading zeros)
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 text message between two people
Integer	The number of students in a class
Float	The distance between two points
Boolean	Whether the user is older than 13

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	int
True		True	boolean

a	a = 2.5	2.5	float
1 + 2 * 3		7	int
a and False	a = True	False	boolean
a or False	a = True	True	boolean
a + b	a = 1 b = 2	3	int
2 * a	a = 3	6	int
a * 2 + b	a = 2.5 b = 2	7	float
a + 2 * b	a = 2.5 b = 2	7.5	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 that declaring a variable makes the computer aware of its existence in the program as well as its name and data type, while initialising a variable just gives it an initial value.

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 named variable that identifies a value passed into a function or method. This value can then be used inside that function or method.

```

class MainClass
{
    static void Main(string[] args)
    {
        Greet("Tung"); // Prints "Hello Tung" in the console.
        Console.ReadLine();
    }

    static void Greet(string name)
    {
        Console.WriteLine("Hello " + name);
    }
}

```

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

Scope refers to where a variable or function is accessible inside a program. For instance, in C#, if a variable is scoped inside a class, it is only accessible to code inside that class.

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.

```

static float Average(int[] numbers)
{
    int sum = 0;
    for (int i = 0; i < numbers.Length; i++)
    {
        sum += numbers[i];
    }
    float average = (float)sum / numbers.Length;
    return average;
}

```

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

```
using System;

namespace Average
{
    class MainClass
    {
        static void Main(string[] args)
        {
            int[] array = { 1, 5, 5 };
            Console.WriteLine(Average(array));
            Console.ReadLine();
        }

        static float Average(int[] numbers)
        {
            int sum = 0;
            for (int i = 0; i < numbers.Length; i++)
            {
                sum += numbers[i];
            }
            float average = (float)sum / numbers.Length;
            return average;
        }
    }
}
```

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.

```
using System;

namespace Average
{
    class MainClass
    {
        static void Main(string[] args)
        {
            int[] array = { 1, 5, 5 };
            float average = Average(array);
            Console.WriteLine(average);

            if (average >= 10)
            {
                Console.WriteLine("Double digits");
            } else
            {
                Console.WriteLine("Single digits");
            }

            Console.ReadLine();
        }

        static float Average(int[] numbers)
        {
            int sum = 0;
            for (int i = 0; i < numbers.Length; i++)
            {
                sum += numbers[i];
            }
            float average = (float)sum / numbers.Length;
            return average;
        }
    }
}
```

C:\Program Files\dotnet\dotnet.exe

3.666667

Single digits