1.1P: Preparing for OOP – Answer Sheet

1. Explain the following terminal instructions:
    a. cd: Used to change directory
    b. ls: Used to list files and folders in the current directory
    c. pwd: Prints the current working 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
A temperature in Celsius	float
The average age of a group of people	float
Whether a person has eaten lunch	bool

3. Aside from the examples already given, come up with an example of information that could be stored as:

Data type	**Suggested Information**
String	Address
Integer	Items in cart
Float	Account balance
Boolean	Is account locked or unlocked

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**
`5`	`5`	5	int
`True`	`True`	True	bool

`a`	`a = 2.5`	2.5	float

`1 + 2 * 3`	`1`
	`2`
	`3`
		7	int

`a and False`	`a = True`	False	bool
`a or False`	`a = True`	True	bool
`a + b`	`a = 1` `b = 2`	3	int
`2 * a`	`a = 3`	6	int
`a * 2 + b`	`a = 1.5` `b = 2`	5	int
`a + 2 * b`	`a = 1.5` `b = 2`	5.5	float

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

`"Fred" + " Smith"`			string
`a + " Smith"`	`a = "Wilma"`	Wilma Smith	string

5. Explain the difference between **declaring** and **initialising** a variable.

The difference between the two is declaring a variable is when you specify the type (int) and identifier (a for example), but not assigning a value to it, whilst defining a variable is when you assign a value to the variable, for example, int a = 5, int is the type, a is the identifier, and 5 is the value.

6. Explain the term **parameter**. Write some code that demonstrates a simple of use of a parameter.

A parameter is a variable inside a method. We can assign values and call these parameters from the method.

```
1 reference
static void ParaMethod(string name)
{
    Console.Write(name + " Doe");
}

0 references
static void Main(string[] args)
{
    ParaMethod("John");
}
```

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

Scope refers to a region of code where variables can be called and used. These are usually class-level variables, which can be accessed from any method in the class, and method-level variables which are exclusively accessed in their methods and cannot be accessed outside them.

```
0 references
public class Scopes
{

    0 references
    private int classlevel = 2;

    0 references
    public void MethodScope()
    {
        int methlevel = 1;

        Console.WriteLine(methlevel);
    }
```

Here we have classlevel variable which is defined within the Scope class but outside of the other methods. So it can be called from anywhere within the class, whilst methlevel variable is defined inside the MethodScope method and is restricted inside there.

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. 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 task.

```
public static double Average(int[] numbers)
{
    int sum = 0;

    foreach (int number in numbers)
    {
        sum += number;
    }

    return (double)sum / numbers.Length;
}
```

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

```
public static void Main()
{
    int[] numbers = {15, 16, 17, 18, 19};

    double avg = Average(numbers);

    Console.WriteLine($"Average is: {avg}");
}
```

10. To the code from 9, add code to print the message “Double digits” if the average is above 10. Otherwise, print the message “Single digits”.

```
5   public class Program
6   {
7
        0 references
8       public static void Main()
9       {
10          int[] numbers = {15, 16, 17, 18, 19};
11
12          double avg = Average(numbers);
13
14          Console.WriteLine($"Average is: {avg}");
15          if (avg > 10)
16          {
17              Console.WriteLine("Double Digits");
18          }
19          else
20          {
21              Console.WriteLine("Single Digits");
22          }
23      }
24
25
        1 reference
26      public static double Average(int[] numbers)
27      {
28          int sum = 0;
29
30          foreach (int number in numbers)
31          {
32              sum += number;
33          }
34
35          return (double)sum / numbers.Length;
36      }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Average is: 17
Double Digits
PS C:\Users\Mason\Documents\Bachelor\OOP\Projects>
```