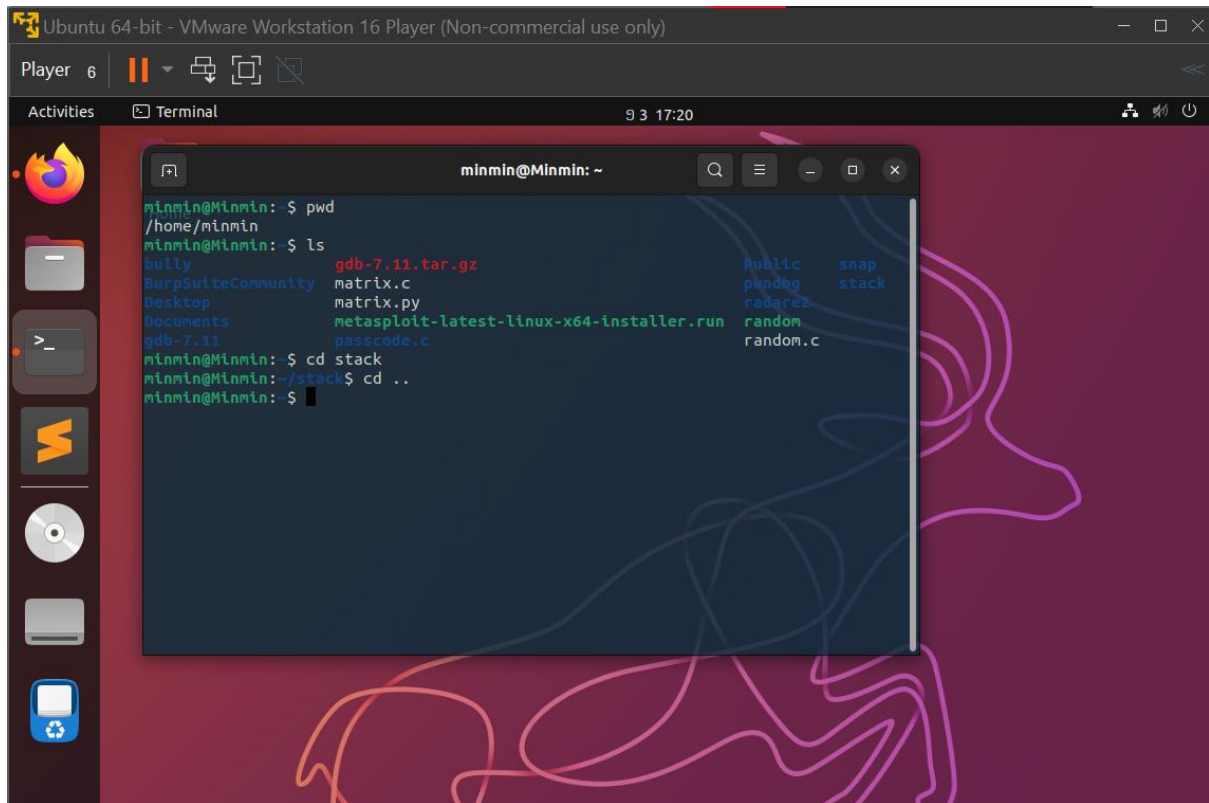


1.1P: Preparing for OOP – Answer Sheet

1. Explain the following terminal instructions:
 - a. cd: is the command line for showing the current directory



The screenshot shows a terminal window titled 'minmin@Minmin: ~' within a VMware Workstation 16 Player. The terminal displays the following commands and output:

```
minmin@Minmin:~$ pwd
/home/minmin
minmin@Minmin:~$ ls
bully          gdb-7.11.tar.gz      public        snap
BurpSuiteCommunity  matrix.c             pwndbg       stack
Desktop        matrix.py            radare2
Documents      metasploit-latest-linux-x64-installer.run  random
gdb-7.11       passcode.c           random.c
```

Subsequent commands show directory navigation:

```
minmin@Minmin:~$ cd stack
minmin@Minmin:~/stack$ cd ..
minmin@Minmin:~$
```

- b. ls: is the command line for listing o showing all the items (files, folders) in the current directory

Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)

Player 6

Activities Terminal 9 3 17:18

```
minmin@Minmin: ~  
minmin@Minmin:~$ pwd  
/home/minmin  
minmin@Minmin:~$ ls  
bully          gdb-7.11.tar.gz      Public          snap  
BurpSuiteCommunity  matrix.c             pwndbg         stack  
Desktop        matrix.py            radare2  
Documents      metasploit-latest-linux-x64-installer.run  random  
gdb-7.11       passcode.c           random.c  
minmin@Minmin:~$ s
```

c. pwd: is the command line for showing the current directory.

Ubuntu 64-bit - VMware Workstation 16 Player (Non-commercial use only)

Player 6

Activities Terminal 9 3 17:16

```
minmin@Minmin: ~  
minmin@Minmin:~$ pwd  
/home/minmin  
minmin@Minmin:~$ s
```

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	Integer
A temperature in Celsius	Float
The average age of a group of people	integer
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	Address: Luong Dinh Cua Street
Integer	Numbers of time I have been vaccinated: 3
Float	My height: 186,5 cm
Boolean	Did you finish the homework ? No, I didn't

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	I = 6	6	integer
True	J = True	True	Boolean
a	a = 2.5	2.5	float
1 + 2 * 3	A = 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	Integer
$a + 2 * b$	$a = 2.5$ $b = 2$	6.5	Float
$(a + b) * c$	$a = 1$ $b = 1$ $c = 5$	10	Integer
"Fred" + " Smith"		Fred Smith	String
$a + \text{" Smith"}$	$a = \text{"Wilma"}$	Wilma Smith	String

5. Using an example, explain the difference between **declaring** and **initialising** a variable.

The difference between the two is if the value is not assigned to the variable so the process is only called as declaration. Declaration defines the existence of a variable and its location.

declare example : `int val1, val2;`

Initialising example: `int val1 = 10, val2 = 20;`

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.

A parameter is a variable with a name passed into a function. For example: person in the below image

The screenshot shows a Ruby IDE with two tabs: `hello.rb` and `test.rb`. The `test.rb` tab is active, displaying the following code:

```
1 def hello(person)
2   print('Hello, ' + person + ' !')
3 end
4 def main
5   puts("Hello, Who are you?")
6   person = gets.chomp.to_s
7   hello(person)
8 end
9 main()
```

Below the code editor, the status bar shows `/home/test.rb 5:28 Spaces: 4 (Auto)`. At the bottom, the `Terminal` tab is active, showing the execution of the script:

```
[user@sahara ~]$ ruby test.rb
Hello, Who are you?
Minh
Hello, Minh ![user@sahara ~]$
```

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

Scope is referring to the named variable which can be came into 2 types of local variable and global variable.

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.

```

using System;

namespace _1._1
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine("Enter the numbers that you want to calculate: ");
            int i = Convert.ToInt32(Console.ReadLine());
            Avera(i);
            static float Avera(int i)
            {
                int[] value; // define array and create variable name
                int j, sum = 0;
                value = new int[i]; // number of values will be stored
                for (j = 0; j < i; j++)
                {
                    Console.WriteLine($"Enter {j + 1} number: ");
                    value[j] = Convert.ToInt32(Console.ReadLine());
                    sum += value[j];
                }
                Console.WriteLine("Sum of all numbers: " + sum);
                float average = (float)((float)sum / (float)i);
                Console.WriteLine("Average: " + average);
                return average;
            }
        }
    }
}

```

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

```
Microsoft Visual Studio Debug Console

Enter the numbers that you want to calculate:
5
Enter 1 number:
1
Enter 2 number:
2
Enter 3 number:
3
Enter 4 number:
4
Enter 5 number:
5
Sum of all numbers: 15
Average: 3

C:\SWINBURNE\COS\20007 - Object Oriented Programming\Week 1\1.1\bin\Debug
ith code 0.
To automatically close the console when debugging stops, enable Tools->C
le when debugging stops.
Press any key to close this window . . .
```

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 _1._1
```

```
{
```

```
    0 references
```

```
    class Program
```

```
    {
```

```
        0 references
```

```
        static void Main(string[] args)
```

```
        {
```

```
            Console.WriteLine("Enter the numbers that you want to calculate: ");
```

```
            int i = Convert.ToInt32(Console.ReadLine());
```

```
            Avera(i);
```

```
            static float Avera(int i)
```

```
            {
```

```
                int[] value; // define array and create variable name
```

```
                int j, sum = 0;
```

```
                value = new int[i]; // number of values will be stored
```

```
                for (j = 0; j < i; j++)
```

```
                {
```

```
                    Console.WriteLine($"Enter {j + 1} number: ");
```

```
                    value[j] = Convert.ToInt32(Console.ReadLine());
```

```
                    sum += value[j];
```

```
                }
```

```
                Console.WriteLine("Sum of all numbers: " + sum);
```

```
                float average = (float)((float)sum / (float)i);
```

```
                Console.WriteLine("Average: " + average);
```

```
                Check(average);
```

```
                return average;
```

```
            }
```

```
            static void Check(float average)
```

```
            {
```

```
                if (average >= 10)
```

```
                {
```

```
                    Console.WriteLine("Double digits");
```

```
                }
```

```
                else
```

```
                {
```

```
                    Console.WriteLine("Single digit");
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
}
```


Microsoft Visual Studio Debug Console

Enter the numbers that you want to calculate:

5

Enter 1 number:

1

Enter 2 number:

2

Enter 3 number:

3

Enter 4 number:

4

Enter 5 number:

5

Sum of all numbers: 15

Average: 3

Single digit

C:\SWINBURNE\COS\20007 - Object Oriented Programming\Week 1\1.1\bin\Debug\netcoreapp3.1\1.1.exe (process 18744) exited with code 0.

To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.

Press any key to close this window . . .

Microsoft Visual Studio Debug Console

Enter the numbers that you want to calculate:

3

Enter 1 number:

10

Enter 2 number:

15

Enter 3 number:

20

Sum of all numbers: 45


Average: 15

Double digits

C:\SWINBURNE\COS\20007 - Object Oriented Programming\Week 1\1.1\bin\Debug\netcoreapp3.1\1.1.exe (process 18744) exited with code 0.

To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.

Press any key to close this window . . .

 Microsoft Visual Studio Debug Console

Enter the numbers that you want to calculate:

3

Enter 1 number:

1

Enter 2 number:

3

Enter 3 number:

3

Sum of all numbers: 7

Average: 2.3333333

Single digit

C:\SWINBURNE\COS\20007 - Object Oriented Programming\Week 1\1.1\bin\Debug
ith code 0.

To automatically close the console when debugging stops, enable Tools->
le when debugging stops.

Press any key to close this window . . .

■