

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

1.1P: Preparing for Object Oriented Programming

PDF generated at 20:07 on Tuesday 7th March, 2023

1.1P: Preparing for OOP – Answer Sheet

1. Explain the following terminal instructions:

a. `cd`:

allows you to move between directories

b. `ls`:

writes to standard output the contents of each specified Directory or the name of each specified File

c. `pwd`:

Your current working directory is the directory where your commands are being executed.

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	Integer
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	Name of a place
Integer	Number of students in a class
Float	Someone's height in Cm
Boolean	If a person has siblings

4. Fill out the last two columns of 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	True	Boolean
a or False	a = True	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	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 + " 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 you declare a variable you give it a name and data type whereas initializing a variable assigns a value to the variable.

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 that is passed into a function.

```
1 a = 10
2 b = 20
3
4 def main(arg1, arg2)
5
6   puts arg1
7   puts arg2
8
9 end
10 main(a, b)
11
```

7. Using an example, describe the term **scope** as it is used in procedural programming (not in business or project management). Make sure you explain the different kinds of scope.

Scope refers to the extent in which a variable can be accessed. Local scopes can be accessed within that specific function whereas in global scoping variables can typically be accessed by a larger space.

8. In a procedural style, in any 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.

```

1 numbers_array = [2,4,5,6,7]
2
3 def average(numbers_array)
4
5     sum = 0
6     index = 0
7
8     while index < numbers_array.length
9         sum = sum + numbers_array[index]
10
11         index += 1
12     end
13     avg = sum.div(numbers_array.length)
14     return avg
15
16 end

```

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

```

19 def main(numbers_array)
20     avg = average(numbers_array)
21     print avg
22
23
24 end
25 main(numbers_array)
26

```

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.

```
19 def main(numbers_array)
20     avg = average(numbers_array)
21     print avg
22
23     if 10 <= avg
24         puts "\nDouble digets"
25     else
26         puts "\nSingle digets"
27     end
28
29 end
30 main(numbers_array)
```

Console

Terminal

4
Single digets

✓ Program exited with code 0