Faculty of Science, Engineering and Technology

Object Oriented Programming

Pass Task 1.1: Preparing for Object Oriented Programming

Overview

We have designed this unit assuming that have already been exposed to some fundamental programming concepts. While we don't expect that you know anything about object oriented programming specifically, we do expect that you have a solid grasp on these pre-requisite concepts.

`Purpose:`	Demonstrate that you have the pre-requisite knowledge required for this unit.
`Task:`	Create a hello world program and extend it to output custom messages for different user names.
`Time:`	This task should be completed as soon as you can.

Submission Details

You must submit the following files to Doubtfire:

- A PDF document containing your written answers.

Instructions

1. Explain the terminal instructions **cd**, **ls**, and **pwd**. Provide a screenshot showing these instructions being used correctly.
2. Consider the following kinds of information, and suggest the most appropriate data type to store or represent each:
   - A person's name
   - A person's age in years
   - A phone number
   - A temperature in Celsius
   - The average age of a group of people
   - Whether a person has eaten lunch
3. Aside from the examples already given, come up with an example of information that could be stored as:
   - A string data type
   - An integer data type
   - A float data type
   - A boolean data type
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`			
`TRUE`			
`a`	`a = 2.5`		
`1 + 2 * 3`			
`a and FALSE`	`a = TRUE`		
`a or FALSE`	`a = TRUE`		
`a + b`	`a = 1` and `b = 2`		
`2 * a`	`a = 3`		
`a * 2 + b`	`a = 1.5` and `b = 2`		
`a + 2 * b`	`a = 1.5` and `b = 2`		
`(a + b) * c`	`a = 1, b = 1,` and `c = 5`		
`"Fred" + " Smith"`			
`a + " Smith"`	`a = "Wilma"`		

5. Explain the difference between **declaring** and **initialising** a variable.
6. Explain the term **parameter**. Write some code that demonstrates a simple use of a parameter.
7. Using an example, describe the term **scope**.
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.
9. In the same language, write the code you would need to call that function and print out the result.
10. To the code from 6, add code to print the message “Double digits” if the average is above 10. Otherwise, print the message “Single digits”.