Swinburne University of Technology

COS20007 Object Oriented Programming

# 1.1P: Preparing for Object Oriented Programming

PDF generated at 21:58 on Tuesday 7th March, 2023

# 1.1P: Preparing for OOP – Answer Sheet

1. Explain the following terminal instructions:
    a. cd: go to a directory
    b. ls: list files in directory
    c. pwd: password

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 | 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 | Team name |
| Integer | Team number |
| Float | Pi |
| Boolean | Whether a computer is functioning |

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` | false | Boolean |
| `a or False` | `a = True` | true | Boolean |
| `a + b` | `a = 1`<br>`b = 2` | 3 | Integer |
| `2 * a` | `a = 3` | 6 | Integer |
| `a * 2 + b` | `a = 2.5`<br>`b = 2` | 7.0 | float |
| `a + 2 * b` | `a = 2.5`<br>`b = 2` | 7.5 | Float |
| `(a + b) * c` | `a = 1`<br>`b = 1`<br>`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 that declaring a variable only means a variable has a name and has its data type assigned to it, while initialising a variable would mean to assign a value to the variable as well.

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 where input data is passed on to a function.

```ruby
def average(array)
    sum = 0
    array.each do |num|
        sum = sum + num
    end
    return sum / array.length
end
```

average is receiving an argument and uses that value in its function.

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 is the extent of which a variable or constant can be identified by the compiler. There's local scope, which is an identifier restricted within its function; and global scope, which can be used in any function.

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.

```ruby
def average(array)
    sum = 0
    array.each do |num|
        sum = sum + num
    end
    return sum / array.length
end
```

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

```ruby
def average(array)
    sum = 0
    array.each do |num|
        sum = sum + num
    end
    return sum / array.length
end

def main
    print average([1, 2, 3, 4, 5])
end

main()
```

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.

```ruby

def average(array)
    sum = 0
    array.each do |num|
        sum = sum + num
    end
    return sum / array.length
end

def main
    average = average([1, 2, 3, 4, 5])
    print average

    if average >= 10
        print "\nDouble digits"
    else
        print "\nsingle digits"
    end
end

main()


```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
single digits
C:\Users\shahn\Documents\code\OOP\week1>ruby test.rb
14
Double digits
C:\Users\shahn\Documents\code\OOP\week1>ruby test.rb
3
single digits
C:\Users\shahn\Documents\code\OOP\week1>
```