1.1P: Preparing for OOP – Answer Sheet

1. Explain the following terminal instructions:
   1. cd: change directory, navigate and redirect folder
   2. ls: list, displays the names of the files and directories present in the current directory.
   3. pwd: print working directory, display the current directory's full path
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 | float |
| Whether a person has eaten lunch | boolean |

1. 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 | A pet’s name |
| Integer | Someone’s home address |
| Float | The ml of a cup of water |
| Boolean | If anyone has eaten lunch |

1. 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:

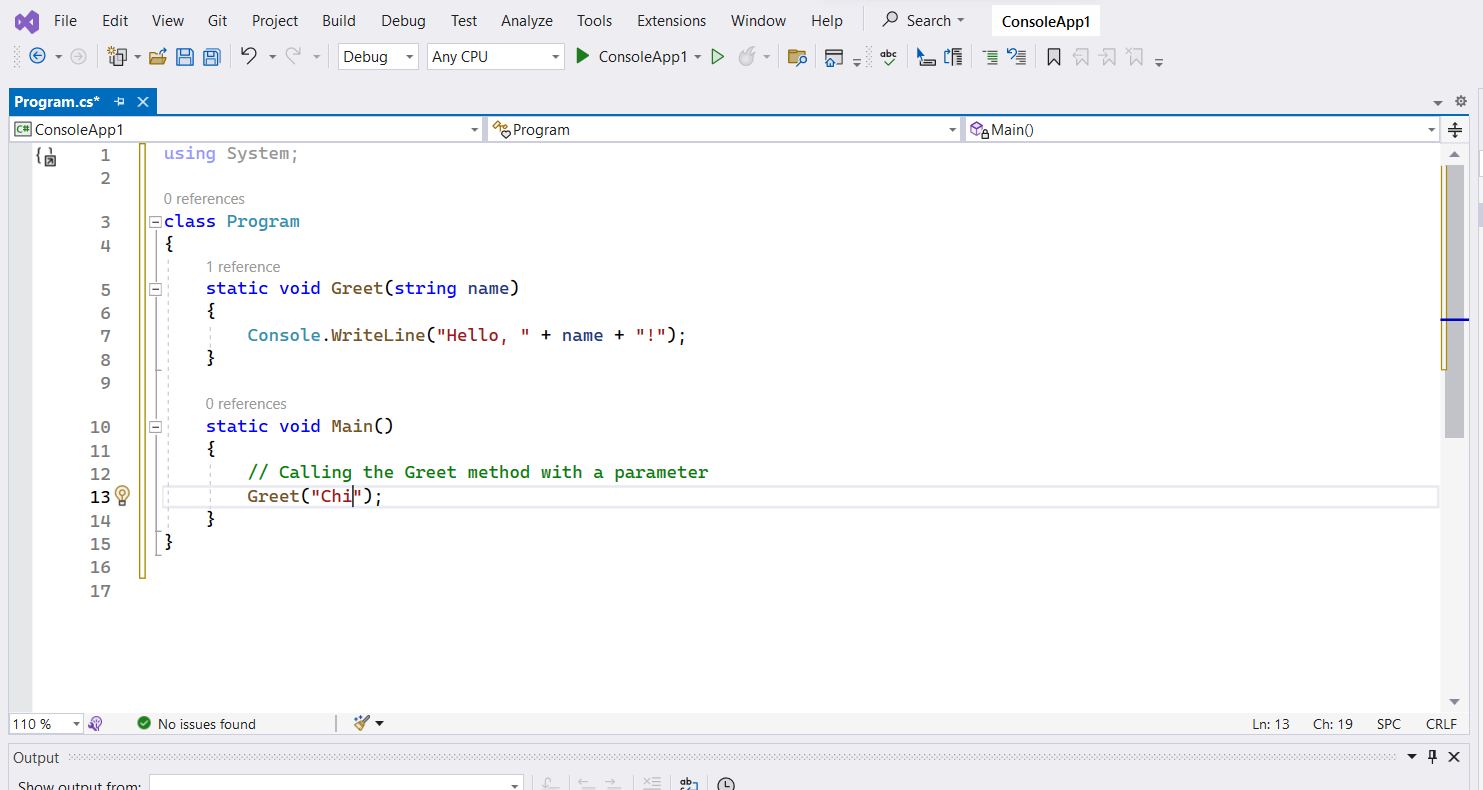
|  |  |  |  |
| --- | --- | --- | --- |
| Expression | Given | Value | Data Type |
| 6 |  | 6 | integer |
| True |  | True | boolean |
| a | a = 2.5 | 2.5 | Float |
| 1 + 2 \* 3 |  | 7 | integer |
| a and False | a = True | False | Boolean |
| a or False | a = True | True | Boolean |
| a + b | a = 1  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 |

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

The difference between the two is declaring is creating space in the computer for a data type, while initialising is assigning an initial value to that storage location.

1. 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.

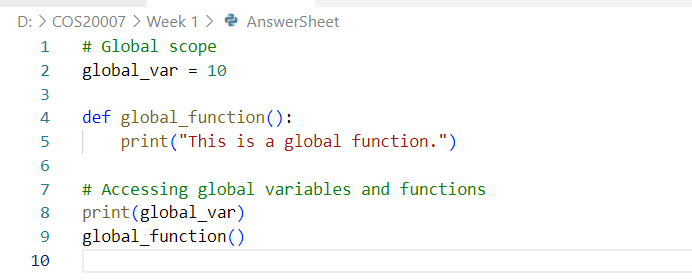
A parameter is a variable or value that is passed into a procedure or function. It allows the procedure or function to accept input values and perform



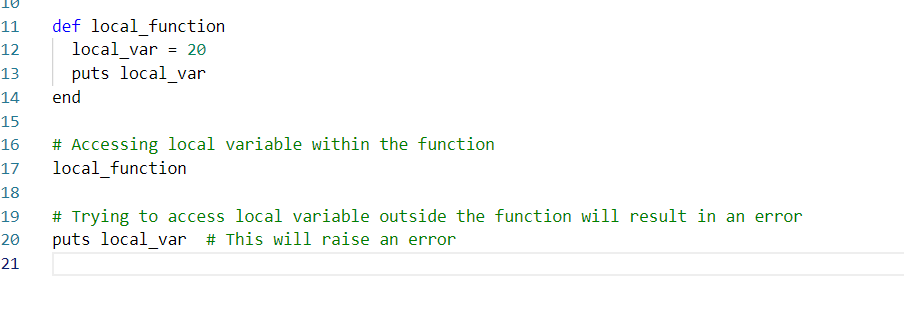
1. Using an example, describe the term **scope** as it is used in procedural programming (not in business or project management). Make sure you explain the different kinds of scope.

Scope, in the context of procedural programming, refers to the visibility and accessibility of variables, functions, and other program elements within a specific portion of code

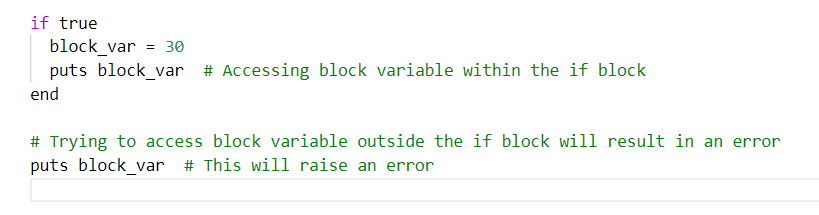
Global scope:



Local scope:



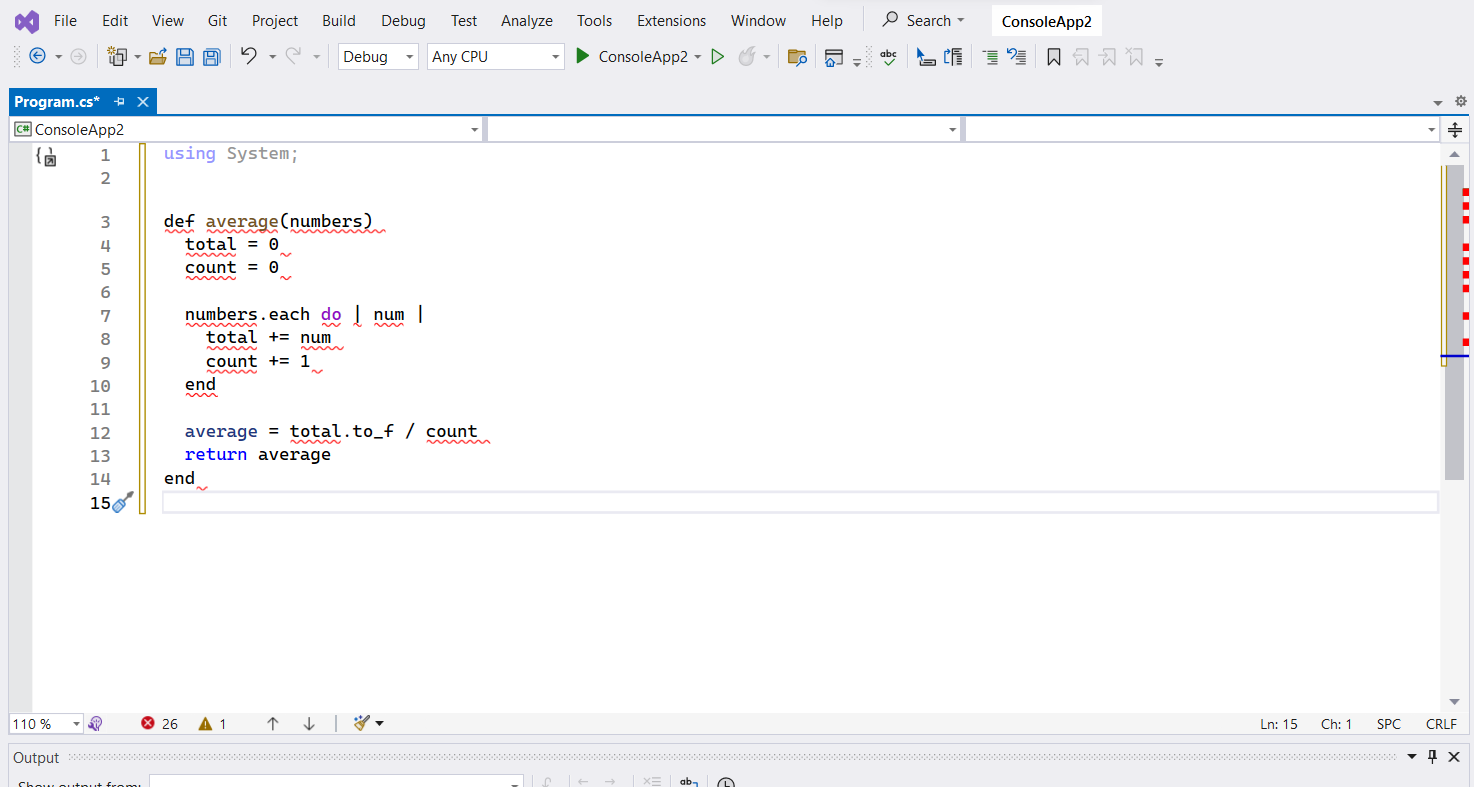
Block scope



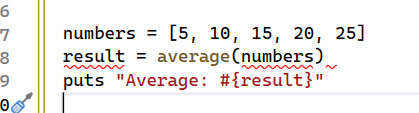
Global variables in Ruby are those that begin with a dollar sign ($), and they can be accessed from anywhere in the program. A method or block's declared variables have a local scope and are only used within that method or block. The specific construct, such as an if statement or a loop, determines the block scope in Ruby, and variables declared within those constructions are only available within them.

1. 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.

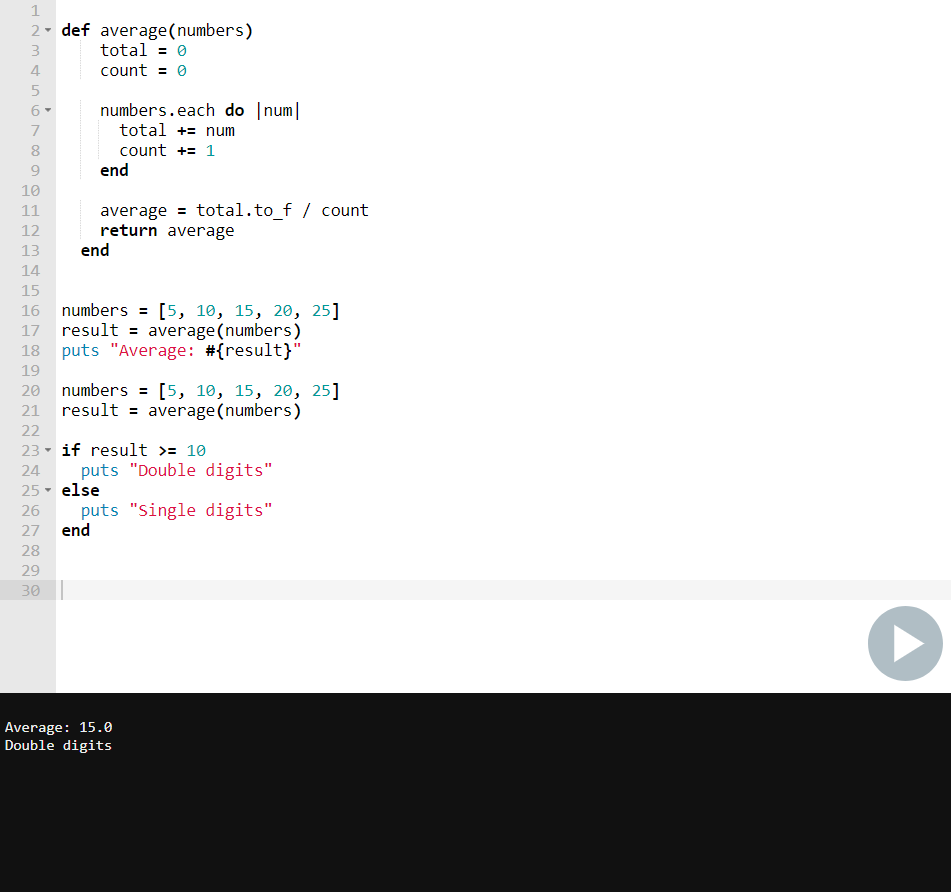
Function called average that accepts an array of integers and returns the average:

**

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

Code to call the average function and print out the result:  


1. 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.

**