1.1P: Preparing for OOP – Answer Sheet

1. Explain the following terminal instructions:
   1. cd:

* Stands for "Change Directory."
* Used to navigate between different directories in a file system.
  1. ls:
* Stands for "List."
* Used to list the files and directories in the current working directory.
* It displays a simple list of names.
  1. pwd:
* Stands for "Print Working Directory."
* Used to display the full path of the current working directory.
* It is helpful to know where user is in the file system hierarchy.

1. 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 | Double, float |
| The average age of a group of people | Double, 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 | Address |
| Integer | ID |
| Float | Score |
| Boolean | Whether a program run or not |

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 |  |  | 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 | Integer |
| (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 a variable means informing the programming environment of its existence, whereas initialising a variable means providing it a starting value.

For example:

Declaring: int x;

Initialising: int x=10;

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 that is used in a function or procedure to receive input values when the function is called.

Example using C programming:  
*A screenshot of a computer

Description automatically generated*

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 is the region of the program where a variable or identifier is accessible and can be used.

There are two types of scope:

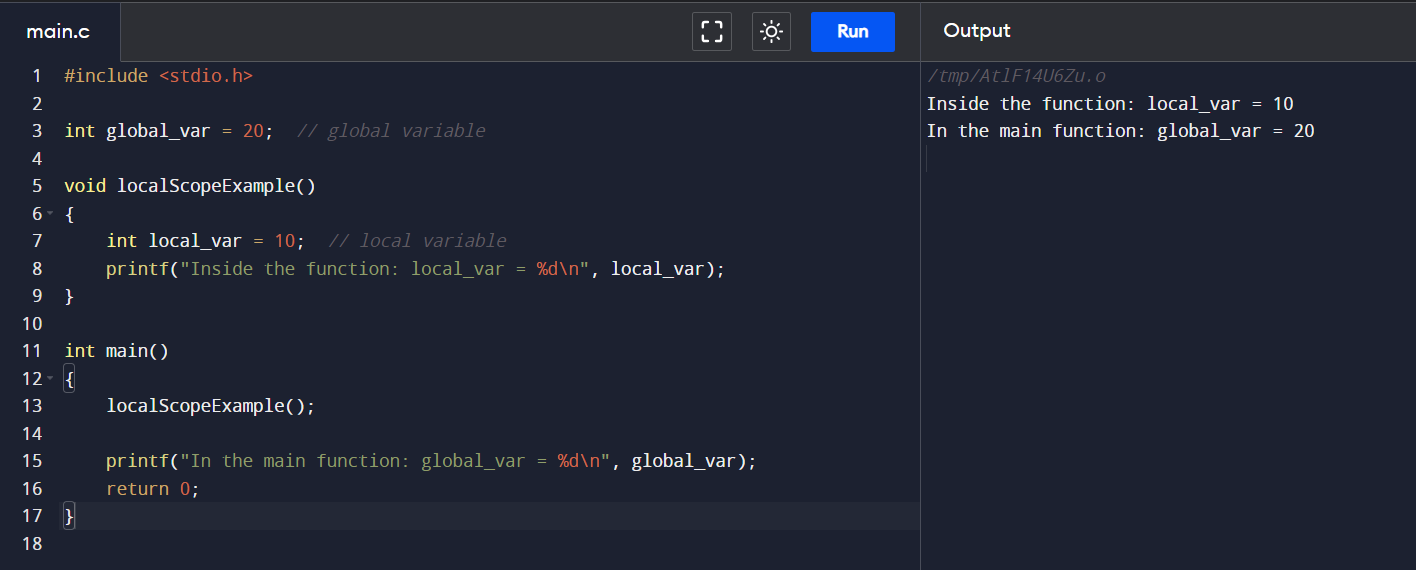
* Local Scope:

The area within a function where a variable is defined is referred to as local scope, also known as function scope. Variables declared in a local scope are only accessible within the function that declared them. They exist only while the function is running and are destroyed when the function is finished.

* Global Scope:

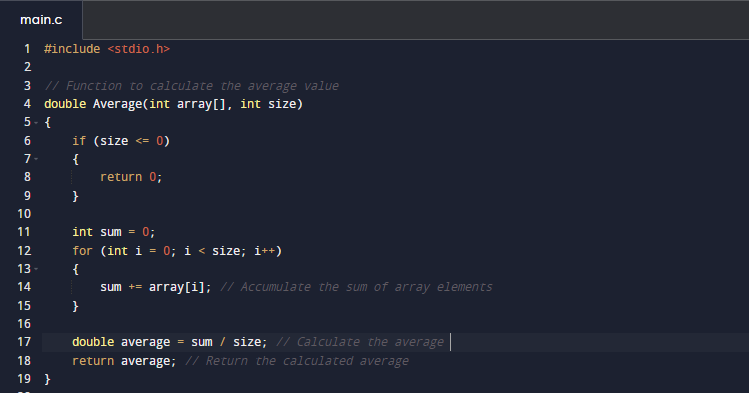
Outside of any function, global scope refers to the main body of the program. Variables declared in the program's global scope are accessible throughout. They exist for as long as the program is active.

Example using C programming:



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.

C Programming:

**

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

A computer screen shot of a program code

Description automatically generated

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.

*<insert a screenshot of your code here>  
A blue screen with white text

Description automatically generated  
<insert a screenshot of your whole program running here>*

A screenshot of a computer program

Description automatically generated A screenshot of a computer program

Description automatically generated