Programming Methodology I COEN-243 Section N NA

SIXTH TUTORIAL

October 26, 2022

LAST TUTORIAL

- FOR Loops (complete)
- WHILE and DO…WHILE loops (complete)
- NESTED LOOPS (complete)
- Infinite loops (complete)
- ARRAYS (ongoing)
- STRINGS (incomplete)

Arrays

- A collection of multiple variables pointed by a single variable
  - You can think of it as a single variable storing multiple other variables with different values
- Declaration: *variable_type array*_name[*array_size*].
- Initialization: *variable_type array*_name[*array_size*] = {variables separated by ‘,’}
- EXAMPLES:
  - int int_array[5] = {1,2,3,4,5}; char char_array[5] = {‘A’, ‘r’, ‘r’, ‘a’, ‘y’};
- Remember that the index starts from ‘0’ to ‘n -1’ for an array of size ‘n’
  - For instance, the value of int_array[0] is 1 and int_array[4] is 5.
  - Accessing indexes bigger or equal to n, e.g., 5 or bigger in the above arrays, is ‘out_of_index’ error
-

2D or Multidimensional Arrays

Similar to an array storing variables, arrays can also store other arrays

- A 2D array is an array that stores multiple other arrays
  - Can be thought of as a big block of memory contain smaller blocks of memory space
- Declaration: *variable_type array*_name[*rows_size*][columns_size]:
  - Initialization 1: *variable_type array*_name [*rows_size*][columns_size] = {variables separated by ‘,’}
  - Initialization 2: *variable_type array*_name [*rows_size*][columns_size] = {{values}, …, {values}}
- Thus, it can be said higher dimensional arrays are also possible

String Variable Type & Library

- A custom variable that contains a collection of characters specified by double quotation mark (“ ”)
- It can be thought of as an array of characters next to each other
- Declaration + initialization: *string* string_name = “text”;
- Similar to arrays, the index can also be used to access specific characters.
- Methods and functions: https://cplusplus.com/reference/string/

Functions

- Functions are chunks of code that run only when it is called
- Highly utilized for reducing repetitive code and reusing that code
- Defining a function in C++:
  - The return type of the function: void, int, bool, char, double…
  - Function name: my_function, (any non-generic name can be used)…
  - Arguments/Parameters: similar to ‘x’ in the function ‘f(x)’
- E.G.: bool is_prime(int num), int square_num(int num), ….
- Pre-defined C++ Function: **int main(){}**

Functions (cont'd)

- User-defined functions are not executed unless called in the main function (directly or indirectly)
- To call a function: function name + (arguments in brackets or brackets only if no args)
- EXAMPLE: is_prime(2); square(10)…
- Since some functions represent a value, they can be treated similar to variables
- **Prototypes (declaration then definition) help with code optimization and organization**

References

- https://www.w3schools.com/cpp/
- https://en.cppreference.com/w/c/language/array_initialization
- https://cplusplus.com/reference/string/
- https://www.geeksforgeeks.org/multidimensional-arrays-c-cpp/
- **https://github.com/TheBarzani/COEN243_Fall2022**

THANK YOU ☺