CPS1011 – Lab 1

Once you finish this tutorial, remember to commit to Github any code you write and to add Github username nevillegrech to your github repo. Your performance in the tutorials will be a factor in your exam scoring.

1. In this listing
    a. List all symbolic identifiers for variables (variable names),
    b. Distinguish between declaration, lvalue and rvalue usage for each occurrence.
2. Go to `onlinegdb.com` and try it.
3. Modify this listing to also ask for the number of cats, as well as to report the total number of pets.
4. Enhance the `butler()` function in this listing in a way to parametrize the butler's greeting so as to display a name instead of sir, in a similar fashion to what `printf()` does.
   Hints: the parameter type should be as per `printf()`'s first, while the place holder for a phrase is %s instead of %d

5. Write a program with the functions `add(), subtract()` and `multiply()`, and test their usage from `main()`.
6. Set up your tool chain. If you are on Windows set up "Windows subsystem for Linux" (WSL):
    a. Install the C compiler tool chain
        i. On Mac: Clang and cmake
            1. `brew install cmake`
        ii. **(Preferred option)** On Linux: gcc and cmake
            1. `sudo apt-get install cmake`
        iii. On WSL: gcc and cmake (sudo apt-get install cmake)
    b. Install CLion or VS Code
    c. If you're using VS Code:
        i. Install the C/C++ tools
        ii. Install CMake tools
        iii. Install WSL extension

Do make multiple attempts at the setup prior to seeking assistance. Perseverance is a key quality for programmers.

Note: If you're working with VSCode, all compilation, build and debugging is to be carried out through the Cmake tools extension i.e. Go to command palette (Ctrl+Shift+P >CMake: <command>)

d. Open
    i. On Windows: Developer Command Prompt for VS Code
    ii. On Linux/Mac: Terminal
    iii. On Windows to start WSL: Run Powershell and type wsl
    iv. If you're using VS Code - Create a new project using CMake Quick Start
e. If you're using CLion - Create a CMake project by following this tutorial
f. If you're using VSCode, browse the cmake command palette
    i. In VSCode: Ctrl+Shift+P
    ii. Staring with CMake: Configure
    iii. And the CMake taskbar at the bottom of VS Code

7. Set up source control
    a. Create a GitHub account (have it ready by tutorial session)
    b. Install git
    c. First-time git setup
    ```
    $ git config --global user.name "John Doe"
    $ git config --global user.email "johndoe@example.com"
    ```
    d. Create a new blank GitHub repository
    e. Clone repo (git clone <github repo URL>)
    f. Create a file in the repo and add some content
    g. Stage files (git add <files>)
    h. Commit staged files (git commit -m "message")
    i. Keep track of un/staged modifications (git status)
    j. Keep track of commits (git log)
    k. Synch (git pull , git push)
    l. Github quick reference.
8. Make use of the debugger to set up breakpoints at the start of each function in question 4 and inspect arguments and return values. Try also changing the arguments on the fly from the debugger itself.
9. Write a simple inventory program that keeps track of A3, A4, and A5-sized paper stock. All three stocks should be initialized to 1,000 packs. The program lets the user place an order that includes the amount of paper packs per size.

Prior to termination the program displays the order details as well as the updated stock amounts.

- You may assume that the user complies to the following instructions given by the program: orders can only contain up to 1,000 packs per paper size, and the ordered quantities can only be by packs rather than paper sheets (no fractions).