DevEnvironmentSetup

July 16, 2021

1 C++ Dev Environment

- the following tools are recommended for this course
  1. Visual Studio (VS) Code editor
     - light weight cross-platform editor for many programming languages; has rich extensions
  2. git client for version control
     - Note: VS Code provides GUI-based git
  3. g++ compiler
- follow the instructions from https://github.com/rambasnet/DevEnvSetup to setup Jupyter Notebook on various platforms

1.1 Using g++ compiler on Windows WSL, Mac and Linux

- the steps provided here assumes that you're using the recommended C++ dev environment above
- open a Terminal program
- be familiar with the terminal and some basic bash commands
- change current working directory to where the right folder where the .cpp file is
  - use `ls` commad to see all the contents of the directory
  - use `cd <dir_name>` command to change directory to the given dir_name
- make sure the current working directory is where your `.cpp` file is
  - use `pwd` command on a `*nix` terminal to know the current working directory
- compile using g++
- run the executable
- the following sequence of commands are worth remembering
  - can use these commands on repl.it cloud-based IDE as well

```
$ cd projectFolder # change working directory to the project folder
$ pwd # print current working directory
$ ls # list contents of current directory
$ g++ -std=c++17 -o outputProgram inputFile.cpp  # compile inputFile.cpp to outputPrgram
$ ./outputProgram  # run output program
```

1.2 Using Make program

- a great way to compile, build, run, test and deploy C/C++ program
- create a file named `Makefile` inside the project folder

  - see a quick tutorial on Makefile https://makefiletutorial.com/
  - see makefile_demos for various Makefile examples
- use Makefile template provided in makefile_demos/Makefile_template
- run the following commands from inside the project folder on a Terminal

```
$ cd projectFolder # change current working director - folder with c++ file(s)
$ make # build program
$ ls # see the name of your executable in the current directory
$ ./programName # run the program by it's name
$ make clean # run clean rule; usually deletes all object/exe files
```

[ ]: