# Basic Software

By Friend

# Pre-workshop

- https://www.students.cs.ubc.ca/~cs-221/2019W2/resources/shortcourse/
  Watch Sunday video and follow along (it's 2 hrs so allocate your time well)
- https://www.youtube.com/watch?v=HkdAHXoRtos
  Git with command line is more complicated than Github Desktop (what I asked you to use), but you can do more with it
  - You won't need to use this if you're not going into software, but it's good to be aware of them

# If there’s a word you don’t understand in the slide, let me know and I’ll add it here

-

# Note

- While it’s important to be able to have an IDE on your laptop and compile and run your program, I won’t teach it since there’s a lot of OS/machine specifics that’s different for you all. We’ll use online GDB for today.
- Same with reasoning with Git. You should download Git Bash and try it out on your own time. Try putting the code you wrote today on your own repo.

# Agenda

- We’ll create a sample system to review some cpp topics you saw in the video
- We’ll do some LeetCode questions to practice problem solving

# Part 1: C++ practice

# Sample problem: team membership

- UBC Rapid has members, each with their own info (name, contact info, position, subteam, year joined, etc.)
- It would be good to be able to quickly tell:
  - How many members are on the team?
  - Who are the execs? Who are the captains?
  - What are the members’ contact info?
  - Who are on each subteam?

# Sample problem: team membership

- Whenever you see a problem, break it down into steps
- Sounds like we need team class, member class, and a few functions
- There are many ways to do this; this is just one of them

# Get started

- Go here https://www.onlinegdb.com/online_c++_compiler to start writing
- There's a cpp file in the repo called "starter.cpp" that when run, should print stuff that's specified
- Use that file to guide you on what functions/classes to write
- Use online references to find syntax and data structures

# Hint

Sample steps:

1. Make member class with each field being each personal info associated with member, and a constructor that takes those fields
2. Make team class with a list of members to store all members, and functions to read that list and return whatever is asked for

https://appdividend.com/2019/06/12/cpp-list-tutorial-with-example-list-in-c-standard-template-library-stl/ <- Helpful

# Part 2: Problem solving

# Fizz Buzz

- Leetcode is a popular platform for coding interview questions
- Fizz Buzz is a fairly well known and simple coding question
- Try this out https://leetcode.com/problems/fizz-buzz/
- Most easy problems have solutions available

# Two Sum

- https://leetcode.com/problems/two-sum/

# Part 3: Other info

# Quick concepts related to Marlin

- Think of Marlin code like Arduino code
- Board runs setup() when it starts, then runs loop() till it’s shut down
- Instead of defining everything in a single cpp file, you define stuff in header file (.h) and then #include that header file into your cpp code
  - https://docs.microsoft.com/en-us/cpp/cpp/header-files-cpp?view=vs-2019

# Some further readings

- Get an IDE, and compile and run your code
    - If you use VS Code: https://code.visualstudio.com/docs/languages/cpp
- Look at these references https://www.students.cs.ubc.ca/~cs-221/2019W2/info/syllabus/#c-reference
- Second part to the pre-workshop video tutorial covers more advanced object-oriented programming concepts
- CPSC110 (intro to programming) is free on edx https://edge.edx.org/courses/course-v1:UBC+CPSC110+2020S/course/
- https://github.com/MarlinFirmware/Marlin
    - Try cloning Marlin and skimming through it
    - https://github.com/MarlinFirmware/Marlin/blob/2.0.x/Marlin/src/MarlinCore.cpp
      This cpp file is a good entry point to start reading
        - Ctrl-F “setup()” and “loop()”