Lecture 1 8/25/2020

# Introduction

1. Welcome to the Zoom
   1. Please keep your camera and mic on at all times because the Professor wants to keeps this as close at an actual classroom as possible.
2. What will this Class cover?
   1. Data structures, containers, sorting memory for the data
   2. Objectives:
3. Prerequisite
   1. Need to know how to program in C++
   2. Will learn: data structures (vector, linked list, stack, queue, binary search tree, graphs)
      1. Also, Big O notation, worst case scenario, sorting
   3. There is a strong correlation between engineers who know their materials well and those who get paid a lot

# Lecture

1. Arrays
   1. How do we look for what we want in a sorted array?
      1. (linear search) We go through each item one by one
      2. **(binary search) Is the item smaller/bigger than what I’m looking at?**
         1. **Smaller: left side, we don’t need to look at the other side of the array**
         2. **Bigger: Right side**
         3. **So we choose this method in terms of saving time and memory**
      3. Either way, the array has to be SORTED
2. Schedule
   1. 15 weeks of lectures
      1. 30 classes
      2. Can only miss 3 of them
   2. Multiple quizzes
   3. Projects
   4. 2 midterms ()
   5. 1 Final
3. Tom’s Survival Guide
   1. It mentions Tuffix, hopefully it means that they upgraded it
   2. We are structured along major container types
4. Resources
   1. CPPReference.com is the closest thing to a free universal manual for C++
   2. Review them when you get the chance, otherwise they are references
5. Course Strategy Overview
   1. Explore roughly 8 different data structures
   2. From two different viewpoints
      1. Client (user)
      2. Designer (implementor)
   3. Knowledge checkpoints and homework due dates aligned with major topics
      1. The projects help us understand container’s behavior and usage
      2. We will look at examples of code, quizzes, and exams will focus on these implementation
         1. Yes, there will be code writing involved for both projects and tests
6. Exploring from different views
   1. Client
      1. Why this not this?
      2. How do I use?
         1. Code, interface, behavior (front-end stuff)
   2. Designer
      1. How is data structures implemented?
         1. Code, algorithms (back-end stuff)
      2. Implementation options?
7. Projects
   1. They are usually due before class on the Monday before
   2. For specific dates, see the syllabus
   3. Start on the section, due before the next section
8. Exams
   1. A part of the exam that is taken on the computer (multiple choice)
   2. Essay questions
   3. Programming Questions
   4. Drawings (sketch things)
9. Programming Controls
   1. Needs C++17, g++ version 9, clang++ version 9
   2. Please upgrade ALL of them or, better yet, download Tuffix and set up the VM for it
10. Build.sh
    1. Used to
       1. Find all source files in the current and sub directory
       2. Used to required compile options
       3. Compile your work with both g++ and clang++
       4. Grant it permission to execute
          1. chmod +x Build.sh
       5. So apparently they don’t like   
          using namespace std; // :/