PA 1: Environment Setup

Due Date

- Assignment due on 8 April by class before midnight
- Submit all files and projects to perforce
    - Create a directory called: PA1 in your student directory
    - /student/<yourname>/PA1/…

Goals

- Setup your environment correctly
    - Visual Studio's Developers studio for C++ and C#
    - Version control
        - Perforce
    - Communication
        - Piazza Class Forums
- Write a simple C++ project
- Write a simple C# project

Assignments

1. Piazza Class Forum
    a. Join Piazza
        i. Sign-on
    b. Reply to a forum topic in the class forum
        i. Need to start a thread… (sample thread) (piazza link)
    c. Use this Piazza to ask any questions you have about assignments or material in the class
2. Perforce server
    a. Follow the instructions from the class wiki
        i. Setup workspace
        ii. Download reference material
    b. Add some sample files and play around in your student directory.
        i. Add files, check out, submit, delete, add files into different directories
    c. Ask questions
        i. Post to Class forum for class questions
3. Setup Microsoft Visual Studio environment
    a. Install Microsoft Visual Studio 2013 Professional
        i. Install everything or minimally for
            - Visual C++
            - Visual C#
        ii. Install location to download the professional version
            - Download Visual Studio (piazza link)
            - C# and C++ install
        iii. Microsoft Visual Studio 2015 is not used in this class.

4. Create a ***C++ project*** and solution
    - **a. Create the Doubly Linked List program in C++**
        - i. Win32 Console Application
        - ii. Document the all code
        - **iii.** Code should be warning free
    - **b. Program**
        - **i.** Nodes should be dynamically allocated.
            - Nodes should created dynamically
        - **ii.** Should be able to add / delete nodes
            - Add or delete any node anywhere on the tree
        - **iii.** Find specific nodes
            - Walk through every node, starting at the head and find a specific node
        - **iv.** Sort nodes
            - You should be able to sort your linked list according to its data
        - v. Print nodes
            - Print the contents of each node
            - Use printf() to print data
    - c. **Test program**
        - i. **Data – See the Morse code chart**
            - Create a data structure
                a. That contains a character plus an integer, for example:
                char string: "A"
                int: 12
            - The character strings are the alphabet
                a. The data replace dots with 1, and dashes with 2.
            - For example:
                a. Letter ***P*** which is {• — — •}
                i. The string is “P”
                ii. The data is 1221
                b. Letter ***K*** which is {— • —}
                i. The string is “K”
                ii. The data is 212
        - **ii. Test 1: Insert data to a List**
            - Insert the data to create alphabetical order one at the time A-Z
                a. In order insertions:
                i. 1st insert
                1. “A” as a string and 12 as it’s data
                ii. 2nd insert
                1. “B” as a string and 2111 as it’s data
                iii. Goal is to have them in A-Z order
                1. Hint: insert to the end of the list

b. For all letters A-Z (26 of them)
    i. Insert 1 at a time
    ii. If you are a geek (which is a good thing, btw)
        1. Insert to the front of the list in reverse order
        2. Then the linked list is in order… (brilliant)

- Print the whole list (both string and integer data) by walking the linked lists from head to tail
    a. Should be in alphabetical order
        i. 1st node A, 2nd node B, ….

iii. Test 2: Find and delete nodes of a List

- Find specific strings and delete them
    a. Using the list created in Test 1
- Find one at a time a character at the time, then delete that node
    a. Order {F, R, B, Z, A, M, G, R, C, Q, Y, C, N}
        i. In this order!
        ii. F is first, R is second…
    b. Do each search and delete one at a time
        i. Find the character
        ii. Delete them from the list
        iii. If you can't find the specific node, then do nothing
- Print the entire list by walking the linked lists

iv. Test 3: Sort the data

- Sort the list by its MORSE data
    a. Using the list modified in Test 2
- Sort the list by its data
    a. Lowest number to highest number
    b. For example, 'K' is 212, 'S' is 111
        i. 'S' would be lower in value than 'K'
        ii. So 'S' would be before 'K' in the list
- Print the entire list

d. In your main

i. You should have:

- The 3 tests being called individually
    a. Test1()
    b. Test2()
    c. Test3()
- No code in main file
    a. Only includes and these 3 functions

   e. No arrays or built-in containers
      i. No STL, Vectors, Lists, or Arrays allowed
      ii. Need to use DOUBLE linked lists for insert/sort/delete.
   f. You need to submit a complete C++ project
      i. Solution, project and source files
      ii. Do not submit anything that is auto generated
      iii. start a forum thread (PLEASE)
5. Create a ***C# project*** and solution
   a. Create C# Console Application
   b. Repeat problem Morse code problem , but now for C# project
      i. YES - do the same
      ii. Should be a simple port
      iii. Do all 3 tests again in C#
   c. Use `System.Console.WriteLine()` to print data
6. Make sure you do good Perforce descriptions on submissions
   a. That was easy!

Validation

Simple check list to make sure that everything is checked in correctly

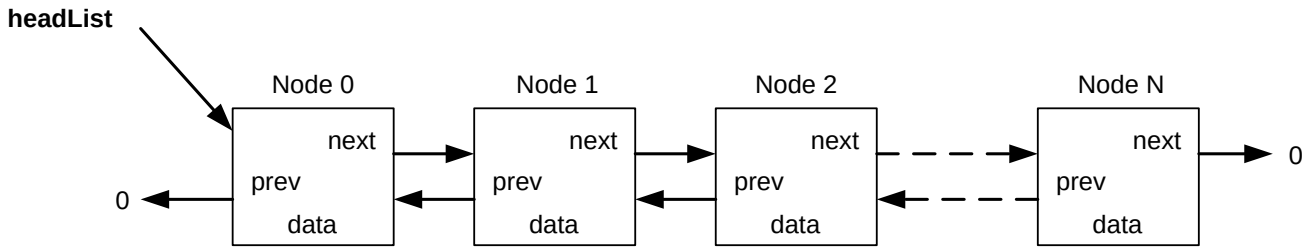
- Did you login and participate in:
  - Piazza?
- Did you submit PA1 into perforce?
  - C++ project
  - C# project
- Did you write good submission comments to perforce?

Hints

Most assignments will have hints in a section like this.

- When lost please post on the forums
  - We can help each other out.
  - Don't get intimidated, we can you get through this material together.
- You will get it.
  - Enjoy – have fun!

Doubly Linked List



Remember there are edge conditions

- No extra terminating (dummy) nodes

Deletion:

4 states that need testing:

- Deleting the First Node
- Deleting the Last Node
- Deleting the Node in the Middle
- Deleting the Only node

Addition:

- Adding to the front
- Adding to the back
- Inserting after a specific node
- Inserting before a specific node

Sorting

- Reshuffling nodes to the correct order
- Need to detach and reinsert node to any location

International Morse Code

1. A dash is equal to three dots.
2. The space between parts of the same letter is equal to one dot.
3. The space between two letters is equal to three dots.
4. The space between two words is equal to seven dots.