

**CSC 1310: Assignment 4**

A cartoon of a cucumber

Description automatically generated

It’s time to play with the Standard Template Library lists! The aim of this lab is to give you experience with using the STL list library, and iterators. To make it fun, we’re building a D&D/MMO character selector. We want to make a circular linked list, where the end will wrap around to the beginning.

# Directions

1. Three given files have been provided:
   1. **main.cpp** – I have provided a printMenu function for your convenience.
   2. **threeCharacters.txt** – input file
   3. **iteratorExample**.cpp – example program of creating a STL list and using an iterator to move through it. You will most likely need to use this as a frame of reference while completing this assignment.

# Character.h

Your character should have **three attributes. Name, class, and race.** There should be a set and a get method for all three of these attributes. You should also have an overloaded output ( << ) operator, more details on how to write that within Module 2 slides. Other than that, there isn’t anything spicy within this class.

# Main.cpp

This is where the magic happens. Two things that you **must** include at the top:

#include “character.h”

#include<list>

Character.h will allow our driver to see and interact with our class, and list will enable us to use the STL library.

Recall that the STL library utilizes templates. To let the library know what data type it is working with, we must specify the data type when we create the list. This can be achieved by placing the type in < >.

Next, we need two iterator objects. One to signify the end of the list, and the other to move around it.

Now we need to finish the setup.

* Call your load characters function with the “threeCharacters.txt” to fill it with some data.
* Set your beginning iterator to the beginning. Use the **.begin() function** to do this.
  + <https://cplusplus.com/reference/iterator/begin/>
* Set your ending iterator to the end, and back it up by one. **.end()** sets the iterator to the “**past-the-end” element.** This was a pain to debug when I built this.
  + https://cplusplus.com/reference/iterator/end/

Now, onto the main loop of the program!

Create a while loop/switch statement combo. At the beginning of the loop, print the current character (using the overloaded output operator), and print the menu. Ask the user for input.

Case 1: Next character

* Check if we have hit the end of the list. If we have, you need to reset our traversal iterator to the beginning of the list (it = myList.begin()).
  + Else, increment iterator.

Case 2: Previous character

* Check if we have hit the beginning of the list. If we have, you need to reset our traversal iterator to the end of the list.
  + Else, decrement iterator.

Case 3: Add character

* Add a character to the list using the addCharacter function. After we add a character, we need to **update the end pointer. Reassign it and decrement it once again.**

Case 4: Delete character

* Delete the character! This can be done by using the .erase() function in our list.
* Since our iterator is now pointing somewhere that no longer exists, we need to update it. You can set it to the beginning.

## Functions

You will have two functions in your **main.cpp: loadCharacters, and addCharacters.**

**void** loadCharacters(string fileName, list<Character>**\*** list);

* This function will open a file, create a character, and add the character to our list.
* To read in from a file, I suggest using getline, and specifying ‘#’ as the separator.
* We can add to our list using either list->push\_front(…) or list->push\_back(…). Either will work.
* The reason we are passing the list in as a pointer, is we have already created the list in main. We want to add things to the list in our main function, so by passing it in as a pointer, we can directly work with that list. To pass the list by reference (memory), I would tack an ampersand (&) to the list in the function call.

**void** addCharacter(list<Character>**\*** ll);

* Instead of reading in from a file, this function reads input from the command line. It follows the same logic as the above function.

# Submitting Your Program

There is no lab report for this one, so instead write me a little snippet on how you feel about this assignment. It is part of the 1310 rework I’m trying to cook up. It can just be a .txt file, I’m not looking for anything professional.

Zip up your **main.cpp**, **threeCharacters.txt, and character.h, and write-up files and upload them to Module 5 Assignment Dropbox.**

# Sample Output

User input is highlighted in **yellow**.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CURRENT CHARACTER**

**Character: Grog**

**Class: Barbarian**

**Race: Human**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1. Next Character**

**2. Previous Character**

**3. Add Character**

**4. Delete Current Character**

**5. Exit Program**

**1**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CURRENT CHARACTER**

**Character: Bran**

**Class: Paladin**

**Race: Human**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1. Next Character**

**2. Previous Character**

**3. Add Character**

**4. Delete Current Character**

**5. Exit Program**

**1**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CURRENT CHARACTER**

**Character: Elric**

**Class: Wizard**

**Race: Elf**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1. Next Character**

**2. Previous Character**

**3. Add Character**

**4. Delete Current Character**

**5. Exit Program**

**1**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CURRENT CHARACTER**

**Character: Grog**

**Class: Barbarian**

**Race: Human**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1. Next Character**

**2. Previous Character**

**3. Add Character**

**4. Delete Current Character**

**5. Exit Program**

**4**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CURRENT CHARACTER**

**Character: Bran**

**Class: Paladin**

**Race: Human**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1. Next Character**

**2. Previous Character**

**3. Add Character**

**4. Delete Current Character**

**5. Exit Program**

**1**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CURRENT CHARACTER**

**Character: Elric**

**Class: Wizard**

**Race: Elf**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1. Next Character**

**2. Previous Character**

**3. Add Character**

**4. Delete Current Character**

**5. Exit Program**

**1**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CURRENT CHARACTER**

**Character: Bran**

**Class: Paladin**

**Race: Human**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1. Next Character**

**2. Previous Character**

**3. Add Character**

**4. Delete Current Character**

**5. Exit Program**

**3**

**CHARACTER NAME:**

**brandon the cool guy**

**CHARACTER CLASS:**

**comedian**

**CHARACTER RACE:**

**human**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CURRENT CHARACTER**

**Character: Bran**

**Class: Paladin**

**Race: Human**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1. Next Character**

**2. Previous Character**

**3. Add Character**

**4. Delete Current Character**

**5. Exit Program**

**2**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CURRENT CHARACTER**

**Character: brandon the cool guy**

**Class: comedian**

**Race: human**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**1. Next Character**

**2. Previous Character**

**3. Add Character**

**4. Delete Current Character**

**5. Exit Program**

**5**