Computer Science II — CSci 1200
Lab 10
String Class and Operators

Introduction

This lab explores dynamic memory and operators in the context of the `str` class — our own implementation of `std::string`. Get started by downloading the following files:

```
http://www.cs.rpi.edu/academics/courses/fall04/cs2/lab10/str.h
http://www.cs.rpi.edu/academics/courses/fall04/cs2/lab10/str.cpp
http://www.cs.rpi.edu/academics/courses/fall04/cs2/lab10/str_main.cpp
```

Then, turn off all network connections.

Checkpoints

Use the provided main program to test your solutions to the following problems.

1. Write the `erase` member function of the `str` class. This should behave exactly as `erase` does for `std::string`. It accepts as a single argument an iterator pointing to the location in the string that should be erased. The value at this location should be removed, the values above it should be copied down one location, and the size of the string should be reduced by one. An iterator pointing to what was the next entry in the string should be returned. No memory allocation / re-allocation should be done.

2. Write `operator<` for strings. Do this first as a non-member function (not a friend) and then as a member function. The former should work with the contents of `str` through public member functions. Also, it is exactly the same as the `operator<` on `std::string` objects. You only have to show the non-member version actually working, but show the member version commented out. You will need to add declarations to `str.h` and implementation code to `str.cpp`.

3. Rewrite `operator+=( str const& s )` so that it does NOT use `operator+=( char c)`. Instead, it figures out how big to make the final string and re-allocates the space if it is needed. It can only call `new` one time.

As extra practice, write the member function `substr`.