COM S 213 – Spring 2003

ASSIGNMENT #3: **Operator Overloads in the Menu Class**

DATE GIVEN: **2/6/03**

DATE DUE: **2/13/03**

PURPOSE:
This week we learned about operator overloads and constructors. We'll apply these topics to the Menu class we've been working on.

ASSIGNMENT:
While calling member class functions such as `Menu::printMenu()` and `Menu::getInput()` is fine, we can use operator overloads to provide a "cute" way to call these functions and to expand the functionality of our Menu class. For the first part of this assignment, overload operator<< to print the menu out to the stream specified to the left of the operator, and overload operator>> to read menu input from the specified stream into a member variable of the corresponding Menu class which holds input. An example of these two overloads in action might be this…

```
// Code snippet
void Menu::Run()
{
     // Misc code here

     // Print the menu out to cout
     cout << *this;

     // Read in input
     cin >> *this;

     // rest of code here
}
```

Now, I'm nor trying to suggest that this is the best way to implement the solution, but it does give us experience with some operator overloads! Also it would give us the ability to use our menu system with *any* stream, not just the standard input and output streams (the console).

The second part of your assignment is to add a constructor to the Menu class which reads in 4 menu items from a file called "menu.dat". The menu items will be strings which are only one word long and will be present in menu.dat one string per line. So, an example menu.dat file might look like this:

```
New
Open
Close
Print
```

There will always be a fifth item named "Quit". Your job is to read these strings into the class (create as many member variables as are needed) and then use the data read in when printing the menu.

To create menu.dat, take the four items above and place them in a text file, one item per line. Then save the file as "menu.dat" and use with your program. If the file cannot be opened, the member variables you create which hold the four items should be initialized to an empty string.

Once the data is successfully read in, use the read in data when printing your menu!

The rest of the Menu class should behave similarly to what was implemented in Assignment #2. You may need to add a couple of more member functions to the Menu class to successfully implement the overloads of `operator<<` and `operator>>` (to deal with the fact that these overloads cannot access the private data in `Menu` directly.

This assignment is still on the somewhat easy side, but the assignments will begin to get more difficult as we move forward from here.