COM S 213 – Fall 2002

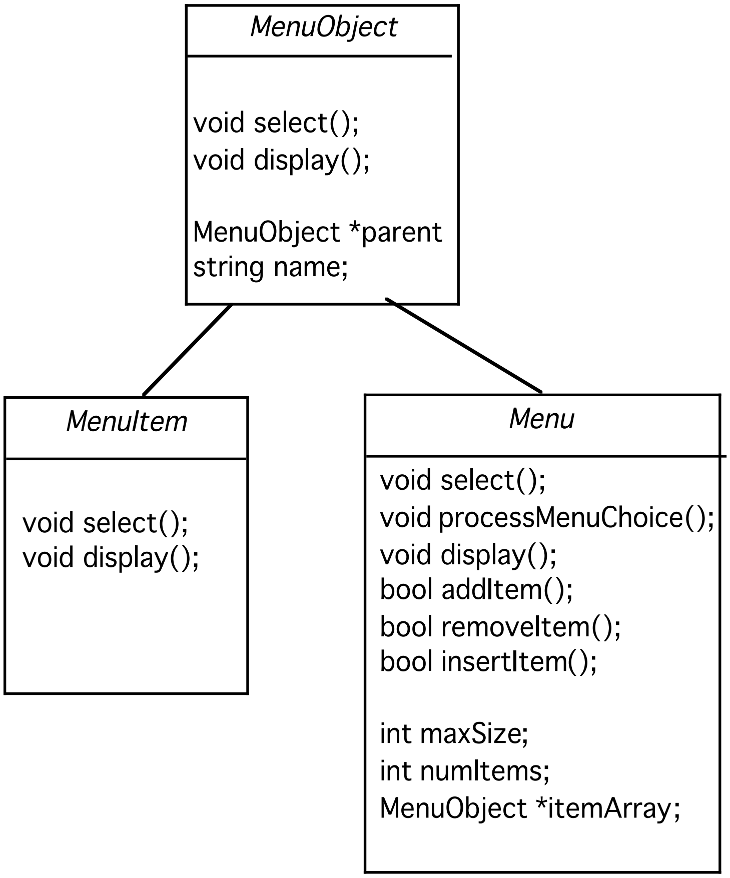
ASSIGNMENT #4: **More Menus**

DATE GIVEN: **10/3/02**

DATE DUE: **10/10/02**

PURPOSE:
To test your understanding of "polymorphism" (also known as utilizing virtual member functions)

ASSIGNMENT:
Consider a simple set of three classes that provide the ability to construct and navigate a text based menu system. Each menu can have as many choices as you'd like, including other menus. In order to implement such a system, you've chosen to utilize inheritance and virtual member functions—and you've drawn a modest diagram of your classes below:



Here, the `MenuObject` class is a base class and `MenuItem` and `Menu` derive from it. Notice inside the `Menu` class there is a dynamic array of `MenuObject` types (the base class). This allows the `Menu` class to have one array of all of its items while still allowing some variance in what type each item is (`MenuItem` or a nested `Menu`).

In this diagram I have given you some ideas for member function and variable names, but I haven't provided any parameter lists (if necessary) with the member functions. Your assignment is to implement this structure with each of the member functions/variables having the following meaning:

CLASS	**MEMBER NAME**	**DESCRIPTION**
MenuObject	Select()	When a MenuObject is selected some action needs to be taken. The specific action is implemented in the overridden version of this member function in each derived class. This member function should *always* be overridden.
MenuObject	Display()	At some point a menu object will need to be displayed. Exactly how that is done is determined in the overridden version of this member function in each derived

		class. This member function should *always* be overridden.
MenuObject	Parent	A MenuObject pointer to the “parent” of this current MenuObject. If this is the “top most” MenuObject this field will be NULL.
MenuObject	Name	The name of this menu object. This is what will be presented to the user when displaying all MenuObjects in a particular Menu.
MenuItem	Select()	This is the one of two member functions for a MenuItem. All it needs to do is print something out which says “Hey, you just chose…” and complete that sentence with the name of this MenuItem taken from the base class. After a MenuItem has been “selected” it will call select() for its parent to keep the menu process “going”.
MenuItem	Display()	For now, all we do when displaying a MenuItem is print out it’s name. You probably want to avoid printing newlines or anything like that, simply send the name string, taken from the base class, to cout.
Menu	Display ()	Displays the name of this menu object with a trailing asterisk to indicate that this is a Menu (which has more choices that will be displayed if selected).
Menu	Select()	When a Menu is selected, it’s contents must be displayed and the user must be given the choice of selecting one of the items, going back “up” one level (if appropriate) or quitting the program. Choice should be presented to the user in the form of a number that represents the item followed by a call to the corresponding MenuObject’s Display() member function.
Menu	ProcessMenuChoice()	This member function prompts the user for a menu choice (any menuobject in the menu or “up one level” or “quit”) and then either quits or calls “select” on the appropriate MenuObject, or quits.
Menu	AddItem()	Adds a new MenuOject into the current menu, at the end.
Menu	RemoveItem()	Removes the specified MenuItem from the

		Menu.
Menu	InsertItem	Inserts a new MenuObject into the Menu at the specified location. Any existing MenuItems at that location or further down are moved down 1 space to accommodate the insert.
Menu	MaxSize	The maximum number of MenuItems allowed in this Menu. There should probably be a constructor that takes this number as an argument.
Menu	ItemArray	A dynamic array of MenuItems. This array is allocated when the Menu object is constructed. If a “setMaxSize()” setter is implemented, the array may need to be reallocated if more storage space is necessary.
Menu	NumItems	The total number of MenuItems in the current Menu (as a result of either add, remove or insert calls).

SUGGESTIONS

OK, that’s a lot of information. This assignment isn’t nearly as daunting as it seems. I’ve just outlined the classes and member functions specifically to steer you in the direction of inheritance and virtual functions.

Draw this out on paper if you need to first to understand how it will be implemented. Add constructors, getters and setters where appropriate. Make some judgment calls on whether or not member variables/functions should be public, private or protected.

As always, feel free to send email with questions…