# Assignment 3 for CPTN230

**Name:** Create and Document a Class Part 2 – Dynamic Memory, Class “static” Members

**References:**

* Text book
* The source files from previous assignments
* The documentation you created for previous assignments
* A3 Sample files
* Internet Help Sites
* Anything else except other people, this an individual assignment

### Assignment Overview:

The goal of this assignment is to have you create a class and demonstrate various techniques of accessing that class. You will only need to define and create one class and an application to use it. You will be asked to separate the application source file from the class source file. For the class you will be asked to create a separate header file from the class source (body) file. This is the exact same packing of files that was used in Assignments 1 and 2.

You get to define what your class is and does. Have some fun with it but between the class and application you will need to demonstrate the following techniques and comply with any restrictions.

* The application will dynamically create multiple instances (objects) of your class
* The application will create enough pointers to track all the objects. How is up to you
* The application will use a function to access the objects via pointers
* The application may not use any global variables or objects
* The class will include the use of a static data member and static member function

Further, in Assignment 2 you were asked to create an application design document. You will do this again for this assignment. The document will have the exact same sections as the one from Assignment 2. You will create the documentation first and then create and test the code.

As always, the requirements for this assignment are exact and no deviation from the requirements or substitution of requirements is allowed

The basic steps needed to complete this assignment are listed below. The “Assignment Description” takes each of these steps and expands on them as does the “Strenuously Recommended Approach” section.

1. Download any needed files from Blackboard.
2. Create the base documentation for the assignment.
3. In stages, create and test the code for the assignment.
4. Post the results to the Blackboard via “View/Complete Assignment” link.

### Assignment Description:

1. Download any needed files from Blackboard.

There are several files.

* This file
* CPTN230A3application\_bettle.pdf
* CPTN230A3class\_bettle.pdf
* CPTN230A3classh\_bettle.pdf
* CPTN230A3output\_bettle.pdf

The .pdf files are copies of demonstration code that will be used in class.

You might want to print out a hard copy for ease of use. Your choice.

1. Create the base documentation for the assignment.

This is, as always, a Microsoft Word 2007 document. It will contain sections used in the same way the document for Assignment 2 was used. See “Assignment Deliverables” for the document outline. Hopefully you kept a copy of the outline document from Assignment 2 to get it started.

The goal here is to get most of the document written so it will serve as a map for your application. You will back fill it as you complete actual code to add or fix details missed on the initial pass.

Do not start to write code until the initial documentation pass is completed. I may even ask to see your documentation before the assignment is due to provide design comments and help.

1. In stages, create and test the code for the assignment.

Suggestions on how to do this is covered in detail in the “Strenuously Recommended Approach” section of this document. But I do want to mention that as you start coding there is a good chance you will have to make minor modifications to the design document. This is expected.

1. Post the results to the Blackboard via “View/Complete Assignment” link.

There are 4 files to be delivered for this assignment. See the next section for details.

### Assignment Deliverables:

There are four files to be delivered with this assignment.

1. A Microsoft Word 2007 Document containing your write up and findings. The name of this file will be **CPTN230\_A3\_Design\_*lastname*.docx** where *lastname* is your actual last name. The format of the document is as follows. Do not deviate.

The Word document must contain the following clearly separated sections.

* Title page
  + Document name
  + Author
  + Creation Date
  + Course Number
* Table of Contents
* Introduction
* A one paragraph description of what the application does
* A description of the program variables and objects
  + Pseudo Code
    - A detailed description of all source code and header files including
    - The purpose of each non blank line
* Conclusions about the application
* References documented using APA format

1. **CPTN230\_A3\_application\_*lastname*.cpp** – The application source file
2. **CPTN230\_A3\_class\_*lastname*.h** – Your class header file
3. **CPTN230\_A3\_class\_*lastname*.cpp** – Your class source file

### Strenuously Recommended Approach:

Even though the word “recommended” is in this section header, consider it as mandatory.

1. Understand the problem.

See Assignment 2 for details.

1. Identify and gather resources.

See Assignment 2 for details.

1. Decide what you want your class and application to do for this assignment.

See Assignment 2 for details.

Big Hint! For this assignment I used the class I used in Assignment 2. It saved a lot of time.

1. Create the base design document outline.

See Assignment 2 for details.

Big Hint!! Reuse the actual design document (with the new name) from Assignment 2. Replace the old content with new or updated information. A fair amount of the content from Assignment 2 is still valid for this assignment.

1. Add the document introduction.

See Assignment 2 for details.

1. Add the application overview to the design document.

See Assignment 2 for details.

1. Add the class header description to the design document.

See Assignment 2 for details.

1. Add the class source code design to the design document.

See Assignment 2 for details.

1. Add the application source code design to the design document.

See Assignment 2 for details.

1. Add the application variables and objects descriptions to the design document.

See Assignment 2 for details.

1. Create an “empty” application file and test it.

See Assignment 2 for details.

Big Hint!! Reuse the application source file (with the new name) from Assignment 2. Comment out most of the code. You can turn them back on as you adjust and use them.

1. Create the class header file and add it to the project/solution.

See Assignment 2 for details.

Big Hint!! Reuse the class header file (with the new name) from Assignment 2. Just add the items you need.

1. Create an “empty” class source file and add it to the project/solution.

See Assignment 2 for details.

Big Hint!! Reuse the class source file (with the new name) from Assignment 2. Just add the new items when you need to add them.

1. In a repetitive procedure add one small feature to the application and test it.

See Assignment 2 for details.

1. Document your application conclusions.

See Assignment 2 for details.

1. Build the references page from the in-line citations.

See Assignment 2 for details.

1. Proofread your Word document.

See Assignment 2 for details.

1. Rebuild the TOC.

See Assignment 2 for details.

1. Post the Word document to Blackboard via the “View/Complete Assignment” link.

See Assignment 2 for details.

As always time management is important. Even if you have not completed the assignment you must submit what you have on the day the assignment is due. The due date will be announced in class. Partial credit is given for partially completed assignments as long as the completed parts are correct. No late submissions will be accepted. In industry when your manager assigns a task and wants something to be done, they expect it on time. It is part of the “salary continuation plan.” I still love that phrase.

1. Down load and test the deliverable.

See Assignment 2 for details.