STYLE GUIDELINES
CS3358 Spring - 2016
Husain Gholoom

Header comments (file documentation block) should be at the top of each file and should contain: File Name, Author, Date, Assignment Number, Course number and section, Instructor, and a brief description of the purpose of the code in the file.

// Roster Number :   Your Roster Number such as 2
//
// Author: Your Name Here
// Due Date: such   as   02/05/2016
// Assignment Number: 1
// Spring - 2016   - CS 3358   -   Your Section Number   such as 1
// Instructor: Husain Gholoom
//
// A representation of a circle properties

#include directives must follow header comments, before the rest of the program.

Comments for variables and functions:
Variable definitions should be commented.
Function prototypes should be commented to describe the purpose of the function.
Function definitions should be commented to describe the parameters, what the function returns (when appropriate) and what it does. See example below. This can go either the .h or the .cpp file, but is not necessary in both.
If the function body contains more than about five statements, there should be comments to describe the various sections of code in the function body.

Variable names:
--must be meaningful
--loop index names can be simple (i, j, k, etc)
--The initial letter should be lowercase, following words should be capitalized, no other caps or punctuation (ie: weightInPounds)

Named constants:
--use for most numeric literals, including array sizes
--all capitals with underscores (ie: TX_STATE_SALES_TAX)
--should occur at top of function, or global (only if necessary)

Line length of source code should be no longer than 80 characters.

Indentation:
  Use 2-4 spaces (but be consistent throughout your program).
  Indent blocks, within blocks, etc.
  Use blank lines to separate sections.

Main Program :

There should be comments to describe the various sections of code in the main program.

Function Prototype example comments:

// short description of what the function does.

Function Definition example comments:

```
//*****************************************************************
// short description of what the function does.
//
// param-1-name description of first parameter (if any)
// param-2-name description of second parameter (if any)
// (remaining params, if any)
// returns description of what function returns (if anything)
//******************************************************************
```

Comments of Classes , templates , Header files , and implementations of classes , templates and header files.

```
//*********************************************************************************************
// short description of what the class /  template / header files  purpose.
//
// Public and private variables and functions
// All Parameter description
// Returns description of what function returns (if anything)
//*********************************************************************************************
```