

Commenting Standards

Main Program

The main program file should include the name of the person or persons responsible for creating the file along with two dates, the creation date and the date of the last update, a description of the program, a list of user interface options, and any notes that need to be made to the reader of the program. The description should be a short paragraph about what the program does. Notes would include things like descriptions of specialized algorithms being used that would not be apparent from the code, libraries that need to be installed, ...

Author: <Names of all authors>

Creation Date: <Date when program was first created>

Last Update: <Date of last update>

Description: <One paragraph describing the function of the program>

User Interface: <List of all user options and short descriptions of each option>

Notes: <Any notes that a user of the program should know about>

Header Files

Header files are similar to the main in that they should include, in addition to general name and date documentation, a description and any notes that need to be made to the reader of the program. The description should be a short paragraph about what functions the file contains. Most well designed header files contain sets of functions with similar purposes. Notes would include things like descriptions of specialized algorithms being used that would not be apparent from the code, libraries that need to be installed, ...

Author: <Names of all authors>

Creation Date: <Date when program was first created>

Last Update: <Date of last update>

Description: <One paragraph describing the set of functions the file contains>

Notes: <Any notes that a user of the program should know about>

Variables

All variables are to have a short description of what they are used for when the variable is declared. This can be done with a one-line comment beside the variable declaration or just above the declaration. It is more readable if the comment is above the variable.

Functions

Each function must have a description and a list of parameters with short descriptions. If the function returns something then there should be a description of what is being returned. Also, you should

include any notes to the user. For example, pre and post conditions, if the function assumes anything about the input parameters, the output value, or if the function will alter any program state it should be noted.

Description: <One paragraph describing what the function does>

Parameters: <List of parameters and their descriptions>

Return: <What is returned by the function, if anything is returned>

Notes: <Any notes that a user of the program should know about, e.g. pre/post conditions>

Blocks of Code

Short comments before blocks of code describing what that block of code accomplishes are helpful to the reader. The more complicated the block is to read the more necessary the commenting.

Class Structures

As stated in the coding standards, non-templated classes are to have separate specification and implementation files and templated classes are to put all code in the header file with the specification at the top and non-inline code implementation below the specification.

Class Declarations / Specifications

The class declaration header file is like any other header file, it should include, in addition to general name and date documentation, a description and any notes that need to be made to the reader of the program. The description should be a short paragraph about the purpose of the class. Notes would include things like descriptions of specialized algorithms being used that would not be apparent from the code, libraries that need to be installed, ...

Author: <Names of all authors>

Creation Date: <Date when program was first created>

Last Update: <Date of last update>

Description: <One paragraph describing the class>

Notes: <Any notes that a user of the program should know about>

All variables are to have a short description of what they are used for when the variable is declared. This can be done with a one-line comment beside the variable declaration.

Class Implementations

The class implementation is the same as a file of functions. The top should contain names and dates as usual.

Author: <Names of all authors>

Creation Date: <Date when program was first created>

Last Update: <Date of last update>

Each function must have a description and a list of parameters with short descriptions. If the function returns something then there should be a description of what is being returned. Also, you should include any notes to the user. For example, pre and post conditions, if the function assumes anything about the input parameters, the output value, or if the function will alter the state of the class in any way, it should be noted.

Description: <One paragraph describing what the function does>

Parameters: <List of parameters and their descriptions>

Return: <What is returned by the function, if anything is returned>

Notes: <Any notes that a user of the program should know about, e.g. pre/post conditions>