

---

**CS590 Programming Guidelines – Summer 2010**

1. Documentation
  - a. Every source code file has good header comments \_\_\_\_\_
  - b. Every module (function, procedure, class) has good header comments \_\_\_\_\_
  - c. Use self-documenting code (i.e. good variable names) \_\_\_\_\_
  - d. Good comments for identifiers (variables, constants, etc) \_\_\_\_\_
  - e. Judicious use of meaningful comments in the body of the code \_\_\_\_\_
  
2. Readability and formatting
  - a. Good use of consistent indentation (3-4 spaces or a tab) \_\_\_\_\_
  - b. Good use of modularity (small coherent functions, etc) \_\_\_\_\_
  - c. Good use of blank lines around control structures \_\_\_\_\_
  - d. Lines of code do not wrap around when printed \_\_\_\_\_
  - e. Avoid global variables if possible, but not at the expense of adding complexity \_\_\_\_\_
  - f. Good use of constants for important numbers or limits \_\_\_\_\_
  - g. Avoid unnecessary code duplication \_\_\_\_\_
  - h. Avoid obscure and complex code when possible \_\_\_\_\_
  - i. Proper use of whitespace around operators, etc. \_\_\_\_\_
  
3. Program output and testing
  - a. Output is readable, well-formatted, and informative \_\_\_\_\_
  - b. Appropriate test cases completed and the results are documented \_\_\_\_\_
  - c. Testing of key errors messages completed and documented \_\_\_\_\_
  - d. Results are reasonably explained \_\_\_\_\_
  - e. Total program package is organized and readable \_\_\_\_\_
  
4. Functionality
  - a. Program meets all specifications of the assignment \_\_\_\_\_
  - b. Source code is well organized and easy to follow \_\_\_\_\_
  - c. Program is on time \_\_\_\_\_
  
5. Deliverables
  - a. Commented source code, and documented test cases \_\_\_\_\_
  - b. Assemble output into logical format (i.e. program 1, output 1, program 2, output 2) \_\_\_\_\_
  - c. Turn in via hardcopy or email \_\_\_\_\_
  - d. Emailed assignments need to be single attachment (text, pdf, or Word doc) \_\_\_\_\_