Marking Codes for Common Mistakes

CS 205 (Spring 2000)

Comments

1. (a) Comment briefly on what your program does.
   (b) Comment on what each function does.
   See *Writing C++ Programs using g++* or your class notes for examples.
2. (a) Comment the 'logical sections' of your code. Be brief but precise.
   (b) Do not restate what the source code says but rather explain what it is for.
3. Comment on what your variables represent.

Readability

4. Use meaningful variable and function names.
5. Avoid 'magic numbers'. Use meaningfully named `const` variables instead.
6. (a) Indent consistently for readability.
   (b) Seperate sections of your program with blank lines.
7. Do not unnecessarily repeat code in a program. Implement repeated tasks as functions or loops.
8. Use modularity: `main()` should mostly just call the functions that are the sub-modules of the program.

Clear coding

9. Do not include unnecessary code. Do not initialize unnecessarily — it's confusing.
10. Use appropriate conditional structures (e.g., `if` or `if/else` or `switch`).
11. (a) Use the appropriate loop structure (e.g. `while` instead of `do/while` or `for`).
    Use a `while` loop when you want to test the condition before you enter the loop the first time.
    Use a `do/while` loop when you want to execute the loop at least once.
    (b) Use a `for` loop only when you know how many times you want to repeat the body of the loop.
    (c) Don't use `break` to jump out of loops.
12. Use parameters instead of global variables to pass information to (and from) functions.
13. Do not use unnecessary parameters.

Output

14. Prompts to the user should be explicit, and helpful to the user.
15. Format your output for readability.
16. (a) Submit all your output.
    (b) Test runs should be executed in the order presented in the assignment.

Other

17. Check your spelling.
18. `main()` always returns an `int`.
    (a) Unless you are passing arguments on the command line, use `int main(void)`