C++ Code Style Document

Purpose: Ensure consistency and readability in code across all projects.

1. Identifiers:

- Class Names: Use `PascalCase`. Example: `MyClassName`.
- Method Names: Use `camelCase`. Example: `myMethodName`.
- Variable Names: Use `camelCase`. Example: `myVariable`.
- Constants: Use `UPPER_CASE` with underscores. Example: `CONSTANT_VALUE`.
- Enums and Enumerators: Use `PascalCase` for enums and `UPPER_CASE` for enumerators.

2. Indenting:

- Use spaces instead of tabs.
- Indent Size: 4 spaces per level.

3. Whitespace:

- Place spaces around operators and after commas.
- No space after a function name and its opening parenthesis.
- No space between a control statement (`if`, `for`, `switch`, etc.) and its opening parenthesis.

4. Bracing Style:

- Use the K&R style (1TBS variant)

```
if (condition) {
 // code
} else {
 // code
}
```

- Braces are not required for single-line control statements but are recommended for clarity.

5. Comment Requirements:

- File Header Comments: Include a brief description of the file's purpose at the beginning of each file.
- Function Comments: Above each function, describe its purpose, parameters, and return value.
- Inline comments should be used to explain complex logic or important decisions.
- Avoid obvious comments.

6. File and Code Organization:

- One class per file, where feasible. File name should match the class name.
- Group related functions and place public members before private ones in classes.

7. Code Review:

- Each source file must be checked by a different person than the author for style guide compliance.
- Reviewer's name to be added to the source file as a comment at the top.

A code style document. This must be ready by the lab session on the 15th. This style document should cover what your identifiers look like (variables, class names, methods), indenting, whitespace, and comment requirements. This should fit on a single page. Make your code follow your style document. You can look at something like the Google C++ style guide, but it covers a lot of C++ feature usage. I am more interested in the visual appearance of your code. So something like http://www.ivanism.com/Articles/CodingStandards.html (Links to an external site.) but even shorter. Besides having coders follow the style guide, each source file should be checked by a different person than the ones who wrote it for style guide violations. For final submission, add the reviewer's name to the source code file they reviewed. Each member of the team must have at least one file they check.