**Use Meaningful Names:**  
- use names that suggest their purpose

**Variable Names**  
- start with a lower-case letter and use upper-case letter as separators  
- do not use underbars (‘\_’)

int myVar = 1;

**Constant Names**  
- use all capital letters and use underbars (‘\_’) as separators

const int MY\_CONST = 1;

**Class Names**  
- start with an upper-case letter and use upper-case letter as separators  
- do not use underbars (‘\_’)

class MyClassName

**Function Names**  
- start with a lower-case letter and use upper-case letter as separators  
- do not use underbars (‘\_’)

int myFunction();

**File Comment Block**  
- every source file must have a comment block at the top containing name of the file and purpose of the file

/\*\*

…

\*/

**Function Comment Block**  
- every function must have a comment block before the function prototype declaration

/\*\*

… or //

\*/  
 int myFunction();

**Curly Braces**  
- place the initial brace on the same line as the keyword and the trailing brace inline on its own line

If (condition) {

…

}

**Whitespace**  
- always layout your source code so that elements that are part of group go together

**No Tab Characters**  
- do not have any tab characters in your code  
- it is difficult to impossible to read source code if your tab settings are different than the authors of the file

**Line Length**  
- limit your line length to about 80-100 characters  
- long lines make it difficult to read your program code, especially when working on smaller screen

**Spacing Around Operators**  
- always put spaces before and after binary operators including << and >>  
- this improve readability of expressions

**Indentation**  
- always indent within curly braces  
- be consistent (3 or **4 spaces**)

**No Global Variables**  
- there is rarely a need to use global variables  
- global variables make a program harder to understand and to maintain

**Atomic Method Functionalities**  
- basically one method for one job  
- this improve of readability of source code

**Committing Single Change**  
- avoid committing many functionalities at once  
- it allows to make easier amendments in the future

**Filename**  
- name your all file staring with upper-case letter and use upper-case letter as separators

**Filename Extension**  
- use **.cpp** extension for your source files  
- use **.hpp** extension for your header files

**GIT**  
- always create a new branch for each new file, then create Pull Request  
- once a pull request is sent, teammates can review the set of changes, discuss potential modifications, and even push follow-up commits if necessary