

Software Specification

Casing Styles

Three styles of casing are used in our software.

- **Pascal Casing** is where the first character in each context change is capitalized starting with the first letter. Context changes occur at word and abbreviation boundaries.
 - i.e. `class TcpListener`
- **Camel Casing** is where the first character in each context change is capitalized *except for the first character*. Context changes occur at word and abbreviation boundaries.
 - i.e. `int localVariable`
- **Bold Casing** is where every letter used is in uppercase and underscores are used at word and abbreviation boundaries.
 - i.e. `#define CAMERA_ENABLED`

Naming

1. Local scope variables and function parameter names shall use Camel casing.
 - a. `int index`
 - b. `void SetSpeed(double speed)`
2. Class names shall use Pascal casing.
 - a. `class Shooter`
3. Interface names shall use Pascal casing preceded by the letter *I*.
 - a. `class ISerializable`
4. Structure names shall use Pascal casing.
5. Function names shall use Pascal casing.
6. File names shall use Pascal casing.
7. Class scope constant names shall use Bold casing.
 - a. `const double PID_P`
8. Typedefs shall use Pascal casing.
9. Enum values shall use Bold casing.
10. Namespaces shall use Pascal casing.
 - a. `namespace CowLib::Server::Utilites`
11. Template variable names shall use T for the template argument if singularly used, but must use Pascal casing preceded by a T_ if there is more than one template variable.
 - a. `template class<T_omato, T_omahto> LetsCallTheWholeThingOff`

- b. `template class<T> OnlyYou`
- 12. Class member variables shall use Pascal casing preceded by a `m_`.
 - a. `Arm* m_Arm`
- 13. Global variables shall be prefixed with the name of the file in which the variable is defined. This will make the variable's definition easier to locate.
 - a. `CodingStandard_GlobalVar = 0; // defined in CodingStandard.cpp`

File Rules

- 1. The file name shall be the same as the name of the *class* defined in the file.
 - a. `Shooter.h` contains the declarations for class *Shooter* and `Shooter.cpp` contains the definitions.
- 2. All header files must begin with

```
#ifndef __Filename_h__
#define __Filename_h__
```

And terminate with

```
#endif
```

Formatting

- 1. [Allman style](#) shall be used.
- 2. Tabs will be used rather than spaces for indentation.
- 3. Any line of text may not exceed 80 characters in width.

Misc

- 1. Magic numbers shall not be used. Constants, enums, and `#defines` shall be used instead.
- 2. Functions shall not exceed 200 lines of code in size (comments not included).
- 3. The verb defining the function shall be placed at the beginning of the function.
 - a. `double GetPosition()` is correct.
 - b. `double PositionGet()` is **not** correct.