

Coding Standards

Optimize Prime

Introduction	2
Standards	2
Spacing and blank lines	2
Indentation	2
Conditional Branching	2
Braces	3
Comments	3
Variable Declarations	3
Classes and Methods	3
Line Length	4
Efficiency vs Readability	4
Object-Oriented vs Procedural	4

Introduction

This is a document that enumerates and explains the various coding standards followed by the team Optimize Prime. These standards ensure for readable and consistent code that is essential for group programming.

Standards

Spacing and blank lines

Blank lines should be used to separate code in the same way as it is used in an essay to separate paragraphs. Statements that contribute to the same concept should be seen as a paragraph.

When a control statement like an “if” or “for” is followed by braces, there must be a space between the statement and the braces. These statements should also be preceded and followed by a blank line.

Function and class declarations and implementations should be preceded and followed by a blank line.

A space should follow each comma when parameters that are comma separated.

Variable declarations should be grouped together in the same paragraph.

Indentation

Lines are always indented when an open brace takes place. The lines stay indented until the closing brace is reached or another open brace takes place in which case the line is indented again.

Conditional Branching

An else statement is placed underneath the closing brace of an if statement. If the else is followed by an if, the if should take place on the same line with a space separating the if and the else.

Example:

```
if () {  
}
```

```
else if () {  
}
```

Braces

Open braces are placed next to the declaration and closing braces are placed underneath the declaration.

.

Example:

```
if () {  
}
```

```
for (int i = 0; i < num; i++) {  
}
```

Comments

Class, method and function declarations must be preceded by a description that will assist fellow developers in understanding the declaration. In the case of a method or function declaration, the description should explain the purposes of all parameters, what the function does and what it returns if it has a return statement.

Inline comments should be used where applicable. Usually above a group of statements that perform tasks that contribute to the same concept.

Variable declarations should be preceded by comments.

“//” is used for all comments.

Variable Declarations

Variables are declared in the Camel Case style and the name should explain what the variable is meant for. The names should not be too long.

Example:

```
int myNumber;
```

Classes and Methods

Classes are declared in the Pascal Case style while methods are declared in the Camel Case style and their names should explain what they are meant for.

Example:

```
class BigDog {  
    void barkLoud() {  
    }  
};
```

Line Length

Lines should not exceed the standard width of an A4 page. If a call to a function for example would violate this rule, parameters should be placed on separate lines and braces follow the rules of braces mentioned above.

Example:

```
veryLongFunctionCall(  
    Parameter1,  
    Parameter2  
);
```

Efficiency vs Readability

It is more important that collaborators can understand each others code rather than it being efficient.

Object-Oriented vs Procedural

We follow the principles of Object-Oriented programming.