General Coding and Style Standards

for Software Projects

*Maniac Branch*

Table of Contents

1. **INTRODUCTION**
   1. Purpose
2. **FILE GUIDELINES**
   1. File Naming Guideline
3. **VARIABLES DECLARATION GUIDELINES**
4. **COMMENTS**
5. **CODE LAYOUT**
   1. One statement per line
   2. Indentation Guidelines
6. **NAMING CONVENTIONS**
   1. Select Clear and Meaningful Names
   2. Naming Constants
   3. Variables
   4. Abbreviations
7. **INTRODUCTION**
   1. PURPOSE

The goal of this document is to create uniform coding practices among the Multithread Maniacs. This document will be updated as the team works on more projects together.

1. **FILE GUIDELINES**
   1. File Naming Guideline

A system will be in place for naming of files for all members. The expected naming convention will be described by the project lead at the beginning of each project.

1. **VARIABLE DECLARATION GUIDELINES**

Always assign a value to a variable. Do not allow the compiler to declare a default value.

1. **COMMENTS**

Comment blocks will be the preferred method of commenting our code. This will be done before a block of code.

For example:

*/\**

*\* This is a block of comments*

*\**

*\* This is more comments*

*\**

*\* This is the end of the comment block*

*\*/*

1. **CODE LAYOUT**

We will be using 4 spaces per indent.

Currently the Multithreaded Maniacs will be placing the brace as follows:

private void exampleMethod()

{

Code here

}

* 1. One Statement per line

We would like to maintain clean code therefore; we will only place one statement per line. This will help with readability so that many members can work on our code at one time.

* 1. Indentation Guidelines

Indented items include:

* Class body
* Method body
* Statements in blocks
* Switch, Case, and Break statements

1. **NAMING CONVENTIONS**
   1. Select Clear and Meaningful Names

For all variables we make we will select intent revealing names. This will make code cleaner and reduce the amount of comments we must place in our code which then makes the code more readable.

* 1. Naming Constants

For all constants we will fully capitalize the variable name, so they are easy to detect. For example:

*CONSTANT\_VARIABLE*

Would follow the constant naming convention.

* 1. Variables

For naming of all variables, we will follow normal java naming convention for example:

*thisIsAVariableName*

This standard will help keep everyone on the page.

* 1. Abbreviations

An abbreviation is acceptable if it will be saving the typing of many letters. For example:

*kb* to replace *keyboard* is acceptable

*ky* to replace *key* would not be acceptable