University of North Alabama

CS 455

Software Engineering Project

Spring 2015

Standards

**General**

GitHub shall be used for source code revision control.

All documentation shall be created in Microsoft Word and stored with GitHub.

**Documentation**

The default font for text shall be Times New Roman size 12 (Excluding the Specification Document).

Source code placed in documents shall be in Consolas size 10.

**Coding**

**General**

A tab shall equal 4 spaces.

No line shall be longer than 80 characters.

All variables shall be declared before used.

All function/methods will begin with a capital letter.

All variables will be begin with a lowercase letter.

All globals, constants, and defined names shall be in all caps. Multiple words shall be separated with an underscore.

int CONTANT\_VAR = 1;

**Layout**

Write only one statement per line.

Write only one declaration per line.

If continuation lines are not indented automatically, indent them one tab stop (four spaces).

There will be one blank line before a method definition.

Trailing braces shall be used everywhere (if, else, functions, structures, typedefs, class definitions, etc.)

if (true) {

}

The else statement shall start on the same line as the last closing brace.

if ( x ) {

} else {

}

Parenthesized expressions will not be padded with spaces.

if (x) {

}

And

x = y \* (z + 5);

Not

if ( x ) {

}

And

x = y \* ( z + 5 );

There will be exactly one space between operators.

x = y + z \* a;

not

x=y+z\*a;

**Commenting**

Place the comment on a separate line, not at the end of a line of code.

Begin comment text with an uppercase letter.

End comment text with a period.

Insert one space between the comment delimiter (//) and the comment text, as shown in the following example.

// This a sample comment.

The use of multi-line comments is allowed.

/\*

\* Multi-line comments

\* can be used.

\*/

**C#**

All class and method names will use PascalCasing.

All class and method names will begin with a capital letter and each successive word starts with and upper case letter.

public int MethodName()

All variable names will use camelCasing.

All variable names will begin with a lowercase letter and each successive word starts with an upper case letter.

int varName;

File names shall use PascalCasing.

FileName.cs

Every class shall be in a separate source file. The file name shall be the name of the class.

Ordering of class variables and methods shall be as follows:

1. public variables
2. protected variables
3. private variables
4. public methods
5. protected methods
6. private methods

All methods will have a header of the following format:

/\*

\* Method: Name

\* Parameters: number and type of parameters

\*

\* Description: Describe the purpose of the method.

\*

\*/

All classes will have a header of the following format:

/\*

\* ClassName.cs

\*

\* Description:

\*

\*/

Use predefined type names instead of system type names.

// Correct  
string firstName;  
int lastIndex;  
bool isSaved;

// Avoid  
String firstName;  
Int32 lastIndex;  
Boolean isSaved;