**Software Engineering I**

Course Number: CS 07321 1

Instructor: Dr. Adrian Rusu

**Fall 2014**

**Team #1**

**December 3rd, 2014**

**Robo-Ops Competition**

**Coding Conventions**

**NASA/Department of Mechanical Engineering**



**Team Members:**

**Daniel Bittner**

**Edward Carter**

**Jun He**

**Matthew Ng**

**Dakota Pollitt**

**Zhentao Zhong**

Implementation

Our team chose to implement our software in Java due to it’s extensive communications libraries, as well as our comfort with the language.

Coding Conventions

**Braces, Parentheses, and Indentation**

Braces will begin on the same line as the structure that called them while ending braces will line up with this structure further down the page.

All indentations will be four (4) spaces. All sub-structures will be indented at their beginning and ended with a closing brace at the same level of indentation.

Line breaks with the following condition:

-- Breaks after semicolons

-- Breaks after method headings follow with open braces

-- Breaks after close braces

**Comments**

Our code has been thoroughly documented using Javadoc. Each method header is preceded by an explanation of the method’s functionality, what parameters it requires, and the resulting output. Inside each method we provide further explanation of the code. Variables and blocks of code are preceded by one or two line comments detailing the operations of slightly more complex portions of our code. Examples are provided below:

**Javadoc Comments:**

/\*\*

\* This is the explanation of the method below

\* @param parameterName This is the name and explanation of the parameter this method requires

\* @return This is the explanation of what this method returns

\*/

public void methodName (parameterType parameterName) {

...

}

**Inner Comments:**

//This is an example comment

/\* This style of comment will be used for comments

using two or more lines \*/

**Declarations and Statements**

Declarations and statements have been limited to one line each.

int a = 1;

String name = “Motor”;

**Control Structures**

Structures will use curly braces even if the statement is one line.

if(condition) {

//code;

}

Dependent structures(if/else, try/catch) will contain the following segment on the same line as the ending brace of the previous structure.

if(condition) {

//code;

} else if(condition) {

//code;

} else {

//code;

}

try {

//code;

} catch(exception) {

//code;

}

switch(variable\_to\_test){

case value:

expression;

break;

case value:

expression;

break;

**.**

**.**

**.**

}

while(condition) {

statement;

}

for(initialization; condition; update) {

statement;

}

**White Space**

Blank lines are added after the end of each method, also between a block or group of code.

Blank spaces are used to separate keywords, commas, variable names, and logical operators.

**Naming Conventions**

**Type Rules Example**

Class name Class names should be meaningful, and first letter of each class ArmControllers;

word should be capitalized. Abbreviations should not be used.

Method name Method names should be short and meaningful, and the first getRotation()

letter of after the first word should be capitalized.

Variable name Variable names should be meaningful. int speed;