**Welcome to the RISK wiki!**

**Coding conventions:**

1. **Naming Conventions for:**

Classes: Class name should be a word starting with an uppercase letter. If it contains multiple words than every inner word should start with an uppercase.

Example: String, StringBuffer

Global Variables: When choosing a name for your variables, use full words instead of cryptic abbreviations. Doing so will make the code easier to read and understand. In many cases, it will also make your code self-documenting. Also, use camel case when variable name contains two words.

Example: speed instead of “s” currentSpeed instead of “cs”

Local Variables: Use camel case for local variables but begin the first letter in lowercase.

Example: currentSpeed

Static Variables: Here the convention changes slightly, capitalizing every letter and separating subsequent words with the underscore character. By convention, the underscore character is never used elsewhere.

Example: NUM\_GEARS = 6

1. **Other Conventions:**

Parentheses: Follow the method of placing parentheses mentioned in Oracle's documentation. Open brace "{" appears at the end of the same line as the declaration statement. Closing brace "}" starts a line by itself indented to match its corresponding opening statement, except when it is a null statement the "}" should appear immediately after the "{".

Example: class Sample extends Object {

int noOfPlayers;

Sample(int i) {

noOfPlayers = i;

}

int emptyMethod() {}

if(true){

return true;

}

else{

return false;

}

}

Comments: Use // to explain variable

Example: intervalMillis; // interval in milliseconds

Block comments: /\*\* \*/ for method description

Example: /\*

\* Here is a block comment.

\*/

Javadoc comments: As per mentioned in Oracle documentation.

Example: /\*\*

\* Here is a block comment.

\*/

The difference between Block comments and Javadoc comments is asterisk(\*). In Javadoc, first line contains two asterisk i.e. /\*\*.

1. **Testing Conventions:**

For testing, follow the rules defined below:

Testing Class will always have a suffix of Test for example if you are testing class AccountServiceImpl then its corresponding test class would be (AccountServiceImplTest).

Methods always end with "test" for example: If we have to test createNewAccount() method then define the following thing in your test class. @Test public void createNewAccountTest() Note: do include @test annotation before the test method.

Without a single assert class the testing is not relevant. So use either of the available assert classes: assertArrayEquals, assertEquals, assertTrue, assertFalse, assertNull, assertNotNull, assertSame, assertNotSame, assertThat and fail.