

**SOEN 6441: Advanced Programming Practice**

**Fall 2018**

**Project – Risk Game (Build 1)**

Submitted To: **Dr. Joey Paquet**

Submitted By: **Team 8**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name** | **Student ID** |
| 1 | Hetal Harishkumar Jariwala | 40039879 |
| 2 | Sadgi Sadgi |  |
| 3 | Mandeep Kaur | 40059801 |
| 4 | Jasraj Bedi | 40046931 |
| 5 | Binay Kumar |  |

**CODING CONVENTIONS**

1. **Code Layout**
2. **Indentation**

* We use indentation of 4 columns.
* Body of the function is indented with respect to its header.
* Body of loop and conditional statements are indented with respect to their first line.

1. **Position of curly brace**

* An open curly brace is appended to the statement preceding it, hence minimizing the code length.

1. **Line spacing**

* Blank lines are used to separate code components and sections, are added between:
* Public, protected and private sections of the class declaration.
* Major sections of long and complicated functions.
* Class declarations in file.
* Function and method definitions.

1. **Space pad operators and equals**

* int add = a + b + c; // good
* int add=a+b+c; // bad

1. **Absence of commented code**

* Unnecessarily commented code specified between “/\*” and “\*/” is deleted to increase the readability.

1. **Naming Conventions**
2. **Classes**

* Class names are in upper CamelCase, with the first letter of every word capitalized.

1. **Attributes (data members) and Methods (member functions)**

* Both data members and member functions are in lower CamelCase.

1. **Constants**

* For constants we use UPPER\_SNAKE case i.e., upper case letters with underscore between words.

1. **Local variables**

* These are written entirely in lowercase without any underscore character.

1. **Global variables**

* Global variable names are prefixed with the project name.

1. **Commenting Conventions**

* There will be a comment at the beginning of each file, explaining the purpose of this file in the project.
* Each class declaration will have a comment explaining what the class is for.
* Each method or function will have comments explaining what it does and how it works, as well as what is the purpose of its parameters.
* All variables declarations, most importantly class data members, should be appended with a comment describing its role, unless its name makes it obvious.

1. **Writing testable code**

* Junit testing is used to test the code functionality.

1. **References**

* Coding Conventions, slide by Dr. Joey Paquet
* [Code Conventions for the Java Programming Language](http://www.oracle.com/technetwork/java/codeconvtoc-136057.html) , [Google's Java Style Guide](https://google.github.io/styleguide/javaguide.html)