**Coding Conventions**

Advanced Programming Practices Fall 2018

Team 20

## Introduction

This document provides a clear overview about the coding standards and conventions are followed for developing RISK game in java. It is strongly recommended that everyone follows this document while writing and modifying the codes.

## Coding Standards

Following standards are maintained while coding for RISK game:

### Indentation and spacing:

Indentation should be applied throughout code to emphasize

1. New blocks
2. Conditional statements
3. Control statements
4. Exception Handling mechanisms

In order to apply indentation, programmer should use four spaces from the beginning of a line. For example:

if(age < 10){

System.out.println(“child”);

}

In order to emphasize the operators and for increasing the readability of code, one space should be used between operators and operands. For example:

int age = 13;

### Line break:

If any statement is too long to contain in a single line, then for better visibility, it should be written in two lines. Either follow “break after comma” or “break after operator” principle.

Public void (int veryBigNameOfTheFirstVariable,

double refundAmountFromTheCashBackPurchase)

{

// body

}

or,

System.out.println(“This is a very large document. ” + “It may contain very useful things. ”

* “So read it carefully. ”);

### Inline comments:

Inline comments should be used after finishing a statement to clarify the purpose of that statement. For example,

int age = 32; // indicates age of the student

### Size of class, subroutines, functions and methods :

It is recommended to keep the size of class, subroutines, functions and methods reasonable. If any method or subroutine goes more than 150 lines then programmer should try to divide that into multiple smaller class or subroutines.

### Variable names:

Variable names should be short and meaningful that convey to a casual observer its intent of its use. Variables should be initialized before first use.

### Use of braces:

Braces are used to delimit the bodies of conditional statements, control structures, blocks of scope. Either of the following styles are encouraged to follow:

for (int i=0; i<5; i++){

// something to do

}

for (int i=0; i<5; i++)

{

// something to do

}

Braces should be used when there is even only one statement in the control block.

### Number of statements per line:

Only one statement should be there in a particular line. It creates ambiguity when two statements are written in the same line of code.

### Use of parentheses

Parentheses should be used whenever there is even a small chance of ambiguity. Even if some operator precedence does not require parentheses specifically, programmers should use them often to make their code more readable.

### Use of curly braces

When writing if,else,while and everything that involves curly braces, they should be in the next line and body is followed by the next line after the braces with default indentation Eg.

if(condition)

{

body

}

### Use of blank lines

Blank lines can be added for separating new code section (i.e, between function and method definitions)

### Naming convention:

Use well-knows camelcase format (capitalize first letter of new word) in case of large names (i.e, dateOfBirth, lastLoginTime).

This convention can be followed in any kind of naming like file naming, class naming etc.