**Coding Conventions – Group 7**

**File Organization:**

1. Java source file contains a single public class or interface or enum.
2. Each sections in file is separated by blank lines and Javadoc comments.

**Naming Convention:**

1. **Classes:** Camelcase format with first letter of each word in uppercase

E.g. class Player, class GamePlay

1. **Interfaces:** Follows same naming convention as Classes. E.g. interface ArmyAssociation
2. **Methods:** Camelcase formatwith the first letter in lowercase and first letter of each internal word in uppercase. E.g. getName(), printAll()
3. **Variables:** Variable names are generally short and meaningful; and are written according to the need of program. E.g. int temp, i, j, k, etc.
4. **Constants:** Constant variables are written in all Capitalized letter with words separated by underscore (“\_”). E.g. int CONTINENT\_VALUE
5. **ENUM:** enums are constants, the name of an Enum type’s field is written in uppercase letters. The enum class body can include methods and other fields. E.g,

public enum Day{

MONDAY(“monday”), TUESDAY(“tuesday”), WEDNESDAY(“wednesday”), THURSDAY(“thursday”), FRIDAY(“friday”), SATURDAY(“saturday”), SUNDAY(“sunday”);

Private String day;

Day(String day){ this.day = day;}

Public String getName(){return name;}

}.

**Indentation:**

We have used four spaces (1 tab) as the unit of indentation.

**Declarations:**

1. Declare only one statement per line. E.g.

int count; //line1

int value; // line 2

1. Local variables are initialized at the point of declaration.

**Comments:**

There are two types of comments used for improving the code readability:

1. **Implementation comments:** It is generally inside of a block describe the code. The formats of implementation comments are **/\*.......\*/** and **//.**
2. **Document comments:**  These comments are generally delimited by **/\*\*. . .\*/ .** Javadoc will be used for document comments.
3. **Add comments for long methods**
4. **Remove commented out code unless necessary**

Add comments at the beginning of each:

1. **File:** The comment helps in explaining the purpose of the file in the project.
2. **Class:** It is used to describing the purpose of the class and must be written just before its declaration.
3. **Method or function:** A comment before a method helps in explaining what that method does and how it works, as well as what is the purpose of its parameters and what to expect as return value.
4. **Variable declarations:** A comment on the side of a variable describes its purpose. For a class data member, a comment describes the role of the data member in that class.

**References:**

* **Robert L. Glass: Facts and Fallacies of Software Engineering; Addison Wesley, 2003. ISBN-13: 978-0321117427.**
* **Oracle Corporation. Code Conventions for the Java Programming Language. http://www.oracle.com/technetwork/java/codeconvtoc-136057.html**
* **Google Inc. Google Java Style: https://google.github.io/styleguide/javaguide.html**
* **Joey Paquet, Course notes for COMP6441: Advanced Programming Practices. Concordia University, Fall 2019.**