**Guidance of JAVA Coding Conventions**

**PARTI Naming**

1. Package

com.XXXX.XXXX

1. Classes/Interfaces
2. concrete class:
   1. *singular noun*
   2. *first letter with each internal words capitalized*
   3. *consistent with table names in DB*
   4. *simple & descriptive*

*Example:*

**public class Department{}**

**public class Account extends Class {}**

1. controller class:
   1. *singular noun*
   2. *first letter with each internal words capitalized*
   3. *simple & descriptive*

*Example:*

**public class DepartmentController{}**

**public class AccountController implements Controller {}**

1. Methods
2. constructor
3. Getters & setters
   1. *Start with get/set + member variable name*
   2. *first letter lowercased, with the first letter of each internal words capitalized*

*Example:*

**public String getDept(){return dept;}**

**public void setName(String name) { cname = name; }**

1. Other methods
   1. *verb ( followed by noun)*
   2. *first letter lowercased, with the first letter of each internal words capitalized*
   3. *simple & make sense*

*Example:*

**public void createNewCourse(int cid, string cname, …){**

**Course course = new Course( , , , ,);**

**…**

**}**

**public Calendar checkTimetable(int fid){**

**return timeTable;**

**}**

1. Variables
   1. *first letter lowercased, with the first letter of each internal words capitalized*
   2. *simple & make sense*
   3. *temporary variables: i, j, k, m, and n for integers; c, d, and e for characters.*
   4. *member variables & local variables: avoid using single letter*

*Example:*

**int i;**

**char c;**

**Account acc;**

1. Constants
   1. *uppercase with words separated by underscores*
   2. *static final*

*Example:*

**static final int MIN\_WIDTH = 4;**

**PARTII Variable Declarations**

1. One declaration per line with trailer comments

**int level; // indentation level**

**int[] array; // array declaration**

1. Initialization of local variables

**JAVA does not require initializing local variables immediately after its declaration; however, it is preferred to have the local variables initialized immediately.**

1. Class, interface and methods declaration

**class object extends Object {**

**private int ivar1;**

**private int ivar2;**

**public constructor (int i, int j) {**

**ivar1 = i;**

**ivar2 = j;**

**}**

**public int emptyMethod() {**

**if(condition)**

**{**

**…**

**}**

**}**

**...**

**}**

**PART III Comments**

*1. Doc comments(Beginning comments of a class)*

**/\***

**\* class name**

**\* version info**

**\* author name**

**\*/**

*2. single line comments*

**if (condition) {**

**/\* Handle the condition. \*/ OR**

**// Handle the condition.**

**If(condition)**

**return false;**

**…**

**}**

*3. block comments*

**if (condition) {**

**/\* This is an**

**\*example of**

**\*block comments**

**\*/**

**If(condition)**

**return false;**

**…**

**}**

*4. trailing comments*

**if (a == 2) {**

**return true; // special case**

**} else {**

**return newCourse(cid); /\* works only for odd a \*/**

**}**

*5. comment out*

**if (foo > 1) {**

**// Do a double-flip.**

**...**

**}**

**//if (bar > 1) {**

**// // Do a triple-flip.**

**// ...**

**//}**

**//else {**

**// return false;**

**//}**

**OR:**

**/\*The following section is for test purpose only**

**if (foo > 1) {**

**// Do a double-flip.**

**...**

**}**

**\*/**

*6. Methods description*

**/\*\***

**\*The Example method provides ...**

**\*@param1**

**\*@param2**

**\*@returns**

**\*/**

**public String Example(int \*param1, int param2)**

**{ ...**

**return str;**

**}**