Coding Standards

for

JAVA Integration

|  |  |
| --- | --- |
| **Developed By** | **Created Date** |
| Nihar Karle  Prem N  Vinit Doshi | 24/01/2019 |

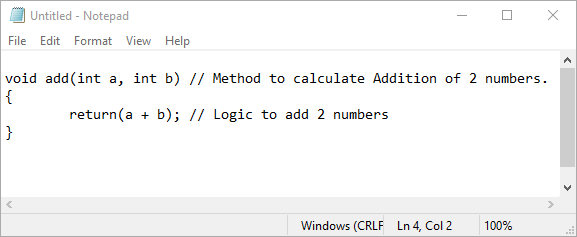
**Index**

|  |  |  |
| --- | --- | --- |
| **No.** | **Topic** | **Page No.** |
| **1** | Things to Follow while coding | 3 |
| **2** | Naming Conventions | 4 |
| **3** | Cyclometric Complexity | 5 |
| **4** | Memory Leakage | 5 |
| **5** | Code Coverage | 5 |
| **6** | Code Repository | 5 |

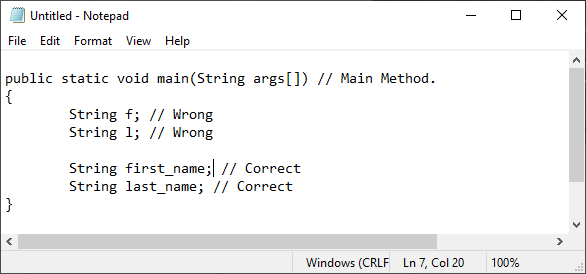
**Coding Standards for JAVA Integration**

1. ***Things to Follow while coding.***

* Comments are Compulsory for every Section.



* Variables should have meaningful names.



* Purpose, Logic and Scope of a method must be commented, documented or looked through and write Start Date and End Date of every method.



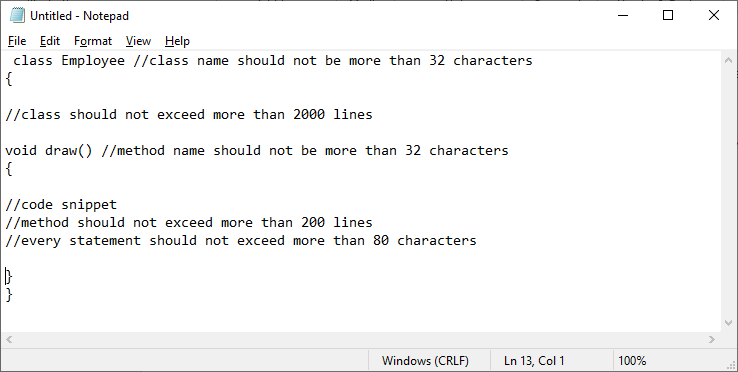
* 2 Approaches of Development are
  + **Business Driven Development [BDD]**

First Code is Written and then Unit Test is done.

* + **Testing Driven Development [TDD]**

First Unit Test case is written and then accordingly code is done.

* **Naming Conventions**
  + Length of the statement should not exceed more than 80 characters.
  + The following are the key rules that must be followed by every identifier:
* The name must not contain any white spaces.
* The name should not start with special characters like & (ampersand), $ (dollar), \_ (underscore).
* **Class**
* It should start with the uppercase letter.
* It should be a noun such as Colour, Button, System, Thread, etc.
* Use appropriate words, instead of acronyms.
* Class names must not be more than 32 characters.
* **Method**
* It should start with lowercase letter.
* It should be a verb such as main(), print(), println().
* If the name contains multiple words, start it with a lowercase letter followed by an uppercase letter such as actionPerformed().
* Method names should not be more than 32 Characters.
* Java follows camel-case syntax for naming the class, interface, method, and variable.
* If the name is combined with two words, the second word will start with uppercase letter always such as actionPerformed(), firstName, ActionEvent, ActionListener, etc.



* **Cyclometric Complexity**
* Cyclomatic complexity is a source code complexity measurement that is being correlated to a few coding errors.
* It is calculated by developing a Control Flow Graph of the code that measures the number of linearly-independent paths through a program module.
* Lower the Program's cyclomatic complexity, lower the risk to modify and easier to understand.
* **Memory Leakage**
* Developer must be careful while using Static keyword. It should not be used everywhere, only use it when the need occurs.
* If a variable is defined static and used for multiple times with different values, then it will not maintain the code stability.
* Exception Handling should be done using Try Catch block.
* **Code coverage**
* Code coverage should be calculated at the end and it must be 80% or more.
* SONAR is the application that is to be used to calculate code coverage.
* Jmeters is a tool/plug in provided by Eclipse IDE for calculating the performance of the code.
* **Code Repository**
* A source-code repository is a file archive and web hosting facility where a large amount of source code, for software or for web pages, is kept, either publicly or privately.
* Usage of Code Repository is must.
* Code repository is done in GIT/SVN.