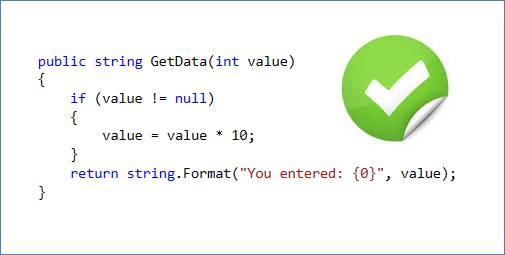
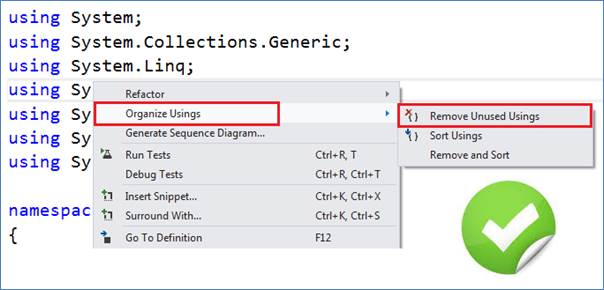
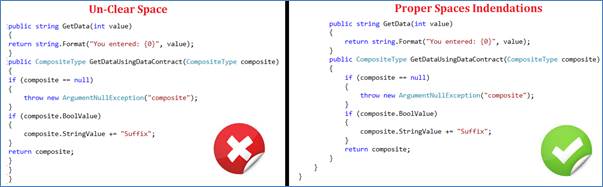
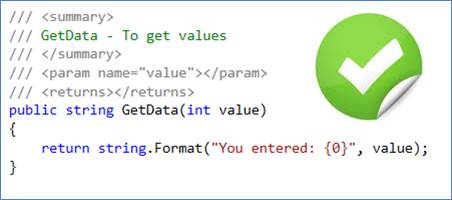
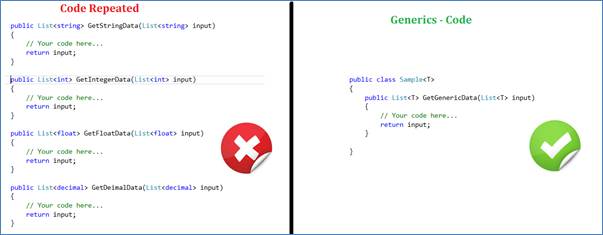
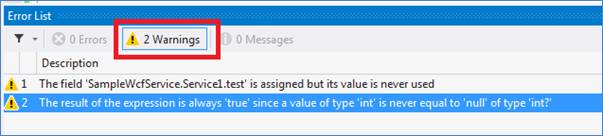
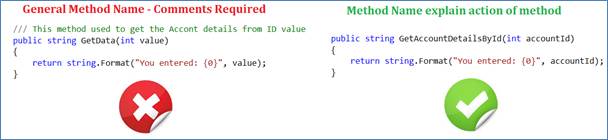
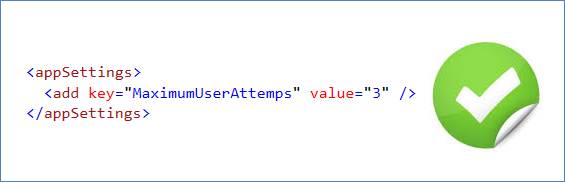
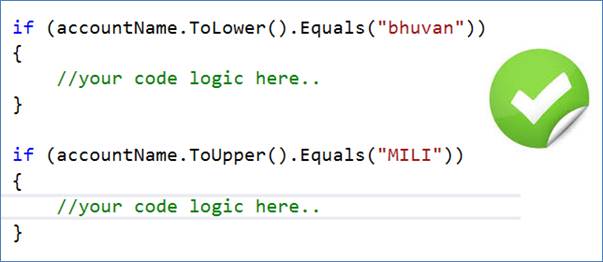
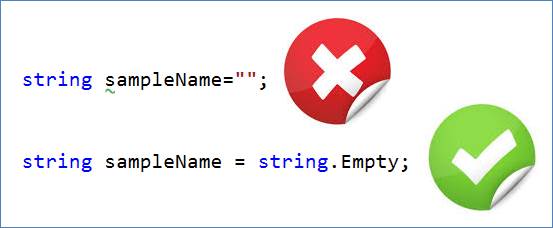
Best Coding Guidelines - Part One

In this article you will get to know about some Best Coding Guidelines in C#.

**Introduction**This article helps you to know about the best coding practices that are used while coding using [C#](http://www.c-sharpcorner.com/technologies/csharp-programming" \t "_blank)language. When we deliver code the created code should follow some standard which will improve optimization, readability, and quicker action on code that also improves the security aspects of code.  
  
**Coding Guidelines**Please find the review guidelines, as follows,  
  
**Standards**This is basically suggested for all developers, code can be revisited or maintained by any developer so many times it's good to follow the basic standards like proper File name, class name and method name.  
  
**Naming Conventions**The Name you have given for all above class, method, variable etc. should have proper naming standards and meaningful information.  
  
**Pascal Casing**

First characters of all words are Upper Case and other characters are lower case.  
  
  
  
**Camel Casing**

First characters of all words, except the first word are Upper Case and other characters are lower case.  
  
  
  
**Check for 'null'**Where ever there is possible for Null occurrence, make an explicit null check, this avoid the execution error.  
  
  
  
**Remove Unused Using’s**Using statement will load all the assemblies, so it’s better to remove unwanted assembly namespace reference in the codes.  
  
  
  
**Indentation and Spacing**It is recommended to use TAB space than a white space and this helps for better code readability. Use single space between two lines of codes and two lines between methods.  
  
  
  
**Comment Lines**It is always good to use the comments lines for each class, methods and some logic which is complex for a reader to understand.  
  
  
  
**Code Reusability**Try to use the available code and use the concepts like Generics to avoid repetitive codes.  
  
  
  
**Project Warnings**Always avoid project warning, which will help hackers to find a path to get into your code and break the applications.  
  
  
  
**Modularity of Code**Avoid writing very long methods. Mostly method should typically have approximately 1~25 lines of code.  
  
If a method has more than 25 lines of code, you must consider re-factoring into separate methods and this is called modularization of code.  
  
As discussed in naming standard, the method name should tell what it does. Do not use misleading names. If the method name is obvious, there is no need of documentation explaining what the method does explicitly.  
  
  
  
**Hard codes**Avoid Hard codes, when a part of value is hard coded in the application, which is not easy to modify the code whenever required; to modify such values again the application should be deployed.  
  
Sometimes the hard coded values may affect the actual logic of application behavior.  
  
  
  
**Config Values**If any of the values are required to be modified frequently or occasionally then it is recommended to have the values stored in the configuration files rather than storing it in any of the application objects and these value modifications do not require the application restart or deployment only the update in config file is fine.  
  
  
  
**String Comparison**When there is any scenario to compare any of the string values, it is always better to convert the string to lower case or upper case and then compare both the string values.  
  
  
  
**String Declaration**It is always suggested to declare the string value with “string.Empty” value than the “” empty values between double quotes.  
  
  
  
**Usage of Enum**We have the special data type called “enum” in C#, it’s strongly recommended to use the enum data type whereever required instead of using the string or integer to indicate discrete values.  
  
  
  
**Note:**We have more guidelines to be discussed, soon expect the next part of coding guidelines.