**Code guidance**

Contents

[1. Purpose: 2](#_Toc463532734)

[2. Namespace: 2](#_Toc463532735)

[3. Make Class name: 2](#_Toc463532736)

[4. Make parameter name: 2](#_Toc463532737)

[5. Make Method name: use verb or verb phrase to make name for method. 2](#_Toc463532738)

[6. Use Comment: 2](#_Toc463532739)

[7. Make Control name: 3](#_Toc463532740)

[8. Formats Code: 4](#_Toc463532741)

[9. Arrange Code: 4](#_Toc463532742)

[10. White Spaces: 4](#_Toc463532743)

[11. Declare variable: 4](#_Toc463532744)

[12. Declare Class and method: 4](#_Toc463532745)

[13. Return: 4](#_Toc463532746)

[14. If-else statements: 5](#_Toc463532747)

[**15.** For statements**:** 5](#_Toc463532748)

[16. While statements: 5](#_Toc463532749)

[17. Switch statements: 5](#_Toc463532750)

[18. Try-Catch statements: 6](#_Toc463532751)

[**19.** Repository: 6](#_Toc463532752)

[20 Comment Repository: 6](#_Toc463532753)

# Purpose:

Show some convention when coding, this rule help team member save huge time in the process of software development and maintenance both during the product. Help student become familiar with teamwork.

# Namespace:

Don’t create 2 namespace have same name or near same name.

Don’t create a method with identical parameters: **void MyFunction(string a, string A)**

Don’t create a method have same name or Different Typefaces: void calculate()/void Calculate()

Each class mustn’t be over 2000 KOL, reasonable delivery code to have a clear structure.

Use Private, Protected, Internal, Public is suitable to security your source code.

# Make Class name:

Use a noun or noun phrase to make name for class.

# Make parameter name:

Parameter name must be descripted fully and it include name and data types.

Type GetType(string typeName)

string Format(string format, object[] args)

# Make Method name: use verb or verb phrase to make name for method.

Exp: RemoveAll()

# Use Comment:

Explain your code to other member by comment, which must be short and clearly.

Avoid to abuse comment in your code.

Use **//** or **/\*…\*/** to comment.

# Make Control name:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Lbl** | **Label** | | **cd** | **ColorDialog** |
| **Llbl** | **LinkLabel** | | **pd** | **PrintDialog** |
| **But** | **Button** | | **ppd** | **PrintPreviewDialog** |
| **Txt** | **Textbox** | | **ppc** | **PrintPreviewControl** |
| **Mnu** | **MainMenu** | | **err** | **ErrorProvider** |
| **Chk** | **CheckBox** | | **pdoc** | **PrintDocument** |
| **Rdo** | **RadioButton** | | **psd** | **PageSetupDialog** |
| **Grp** | **GroupBox** | | **crv** | **CrystalReportViewer** |
| **Pic** | **PictureBox** | | **pd** | **PrintDialog** |
| **Grd** | **Grid** | | **fsw** | **FileSystemWatcher** |
| **Lst** | **ListBox** | | **log** | **EventLog** |
| **Cbo** | **ComboBox** | | **dire** | **DirectoryEntry** |
| **Lstv** | **ListView** | | **dirs** | **DirectorySearcher** |
| **Tre** | **TreeView** | | **msq** | **MessageQueue** |
| **Tab** | **TabControl** | | **pco** | **PerformanceCounter** |
| **Dtm** | **DateTimePicker** | | **pro** | **Process** |
| **Mon** | **MonthCalendar** | | **ser** | **ServiceController** |
| **Sbr** | **ScrollBar** | | **rpt** | **ReportDocument** |
| **Tmr** | **Timer** | | **ds** | **DataSet** |
| **Spl** | **Splitter** | | **olea** | **OleDbDataAdapter** |
| **Dud** | **DomainUpDown** | | **olec** | **OleDbConnection** |
| **nud** | **NumericUpDown** | | **oled** | **OleDbCommand** |
| **Trk** | **TrackBar** | | **sqla** | **SqlDbDataAdapter** |
| **Pro** | **ProgressBar** | | **sqlc** | **SqlDbConnection** |
| **Rtxt** | **RichTextBox** | | **sqld** | **SqlDbCommand** |
| **Img** | **ImageList** | | **dvw** | **DataView** |
| **Hlp** | **HelpProvider** | | **bar** | **StatusBar** |
| **Tip** | **ToolTip** | | **nico** | **NotifyIcon** |
| **Cmnu** | **ContextMenu** | | **ofd** | **OpenFileDialog** |
| **Tbr** | **ToolBar** | | **sfd** | **SaveFileDialog** |
| **Frm** | **Form** | | **fd** | **FontDialog** |
|  |  | |
|  | |
|  | |

# Formats Code:

When an expression can’t add more word in a line, we must down the line by “,”

**Exp: longMethodCall(expr1, expr2,expr3,  
expr4, expr5, expr6);**

# Arrange **Code:**

**when your finish a method or function, you need arrange your code to fix bug easily.**

# **White Spaces:**

**you should delete some white spaces, which is unnecessary to save your KOL. In other hand you should add white spaces after some characters such as:**

for (int i = 0; i< 10; i++)

don’t use: ~~for(inti=0;i<10;i++)~~

# **Declare variable:**

**A lot of person declare variable as: int a, b; //What is 'a'? What does 'b' stand for?**

**It is must be int size; Boolean checkdata;**

# **Declare Class and method:**

**Don’t use White spaces between method name and parameter.**

**“{“ must down the next line.**

**“}” must be same column with “{“.**

# **Return:**

**Don’t use “(“ coveroutsite.**

# **If-else statements:**

if (condition)

{

   DoSomething();

   ...

}

else if (condition)

{

   DoSomethingOther();

   ...

}

else

{

   DoSomethingOtherAgain();

   ...

}

# **For statements:**

**for (inti = 0; i< 5; ++i)**

**{**

**...**

**}**

# **While statements:**

while (condition)

{

       ...

}

# **Switch statements:**

Switch (condition)

{

       case A:

       {

              ...

              break;

       }

       case B:

       {

              ...

              break;

       }

       default:

       {

# **Try-Catch statements:**

try

{

       ...

}

catch (Exception)

{

}

# **Repository:**

**Get and commit your source code to repository follow a mantra: Update -> Coding -> Update -> Commit. Do it follow this mantra your source code have no problem.**

# **20 Comment Repository:**

**When you commit your source code, you need comment what thing you have already fixed or updated and leave your name.**

**Exp: Updated Function chat online – Tai Nguyen**