ITG CODING STANDARDS

Controls Naming

Control name must be clear and understandable, such as:
tbEmployeeFirstName, btnSave, btnCancel, ddlDepartments, hdnEmployeeID... etc.
See the table below:

Control Name	Prefix	Example
AdRotator	ar	arName
Button	btn	btnName
Calendar	clr	clrName
CheckBox	cb	cbName
CheckedListBox	clb	clbName
CompareValidator	cv	cvName
CrystalReportViewer	crv	crvName
DataGrid	dg	dgName
DataGridColumn	dgc	dgcName
DataGridItem	dgi	dgiName
DataList	dl	dlName
DropDownList	ddl	ddlName
FileField	ff	ffName
FlowLayoutPanel	flp	flpName
GridLayoutPanel	glp	glpName
GridView	gv	gvUsers
Hidden	hdn	hdnName
HorizontalRule	hr	hrName
HyperLink	hl	hlName
Image	img	imgName
ImageButton	ibtn	ibtnName
Label	lbl	lblName
Label	lbl	lblName
LinkButton	lbtn	lbtnName
ListBox	lbx	lbxName
ListItem	li	liName
Literal	ltl	ltlName
Multi view	mv	mvName
MultiPage	mp	mpName
Panel	pnl	pnlName
PasswordField	pwf	pwfName
PlaceHolder	ph	phName
RadioButton	rb	rbName
RadioButton	rdb	rdbName
RadioButtonList	rbl	rblName
RangeValidator	rv	rvName
RegularExpressionValidator	rev	revName
Repeater	rep	repName
RepeaterItem	rpi	rpiName
RequiredValidator	rv	rvName

ResetButton	rsb	rsbName
SubmitButton	sbb	sbbName
Table	tbl	tblName
Table	tbl	tblName
TableCell	td	tdName
TableRow	tr	trName
TabStrip	ts	tsName
TextArea	ta	taName
TextBox	tb	tbName
TextBox	tb	tbName
Toolbar	tbr	tbrName
TreeView	tv	tvName
ValidatorSummary	vs	vsName
View	view	viewName

Common Data Types Naming

Variable name must be clear and understandable, such as:
nEmplyeeID, fArticlePrice, bIsValidPrice, structEmployeeInfo...etc.

Data type	Data type prefix	Example
Integer	n	**n**Result
Double	dbl	**dbl**Result
Decimal	dcl	**dcl**Result
Float	f	**f**Result
Boolean	b	**b**Result
Array	[Data type prefix]+Arr	int [] **nArr**Results
Char	c	**c**Result
Byte	by	**by**Result
string	s	**s**Result
Structure	struct	**struct**Person
Void (with methods)	v	**v**SetName()
DataSet	ds	**ds**Clients
DataTable	dt	**dt**Schools
DataRow	dr	**dr**ClientInfo
DataColumn	dc	**dc**Name
DataView	dv	**dv**Districts
DateTime	dti	**dti**CreationDate
Array List	arrl	**arrl**Members
Xml	xml	**xml**Name
• Xml node	xmlnd	**xmlnd**Name
• Xml node list	xmlndl	**xmlndl**Name
• Xml element	xmlel	**xmlel**Name
• Xml document	xmldoc	**xmldoc**Name

Functions

Function name must be clear and understandable and the function prefix must be the returned data type prefix, such as:

Private float fGetArticlePriceTotal(int nArticleID)

Private float[] fArrGetArticlesPrices(int nArticleID)

Private string sGetEmplyeeFullName(int nEmployeeID)

Private bool bIsEmployeeExist(int nEmployeeID) <notice the interrogative name in the Boolean functions>

Private bool bAddEmployees (string[] sArrEmplyeesIDs)

Pages' Names

Page name must contain the main module name then the page functionality for example if you have a page that presents the list of employees then the page name must be (EmployeesList.aspx) and if it updates Employees information it must be (EmployeesUpdateInfo.aspx)

Code Descriptions and Developers Comments

- When adding a new individual class, the header of the file must be like this:

  For ASPX.cs files
```
/* ------------------------------------------------------------------------*/
/* File Name        :GRP_Delete_Everything.cs                              */
/* Created By       :Ismaeel Abu-Tarboush                                  */
/* Creation Date    :55/14/7089 00:00 AM                                   */
/* Comment          :Delete evrything the database.                        */
/* ------------------------------------------------------------------------*/
```

  For Classes
```
/*-------------------------------------------------------------------------*/
/* Program Name      : Oradb.cs                                            */
/* Designed by       : Saed Omar                                           */
/* Created by        : Saed Omar                                           */
/* Creation date     : dd/MM/yyyy hh:tt AM/PM                              */
/* Version number    : 1.0                                                 */
/* Author comments   : Oracle database connection and data manipulation    */
/*-------------------------------------------------------------------------*/
```

- You must add the following comments template above your method, like this:
```
/// <summary>Description and main purposes for the method</summary>
/// <param name="x">Description for parameter x</param>
/// <param name="y">Description for parameter x</param>
/// <returns>return type</returns>
/// <WrittenBy>username  01-01-2005 3:00 PM</WrittenBy>
```

- When adding variables in the web.config file, You must add your hint like this :
```
<appSettings>
<add key="ConnStr" value="pw=xx;uid=yy;ds=orcl"/><!–To be used for db
connection-->
</appSettings>
```

  P.S.: Adding special keys to the web.config file needs an official approval by the team leader.

- When the developer forced to updated or fix bugs in others code he/she must add hint like this:

  (// username dd-MM-yyyy hh:tt) if the update was in a single line and use regions if the update was a new block of code like this:

  #region Block name (username dd-MM-yyyy hh:tt)

  #endregion

  PS, do not delete or update the wrong statement, just comment it then use the above format.

General Notes

- Do not use ambiguous functions names like nGetValue, sGetName, bCheckThis, sMyFunction...etc
- Do not leave any logical ambiguous statements with out writing your comment above it, like this:

  ```
  // Update the index to be 1 instead of –1,
  // Means that the Action has been updated
  OpNode.GetElementsByTagName("Action")[0].Attributes["ID"].InnerText = "1";
  ```
- Do not use any hard coded things in your code such as:

  Server name (Localhost) , Connection string, Paths (URLs or physical)... etc, such data must be fetch from settings or configuration file.
- Do not leave your code scrabble, your code must hierarchical and well-arranged
- Do not leave a block of code that has a certain goal in the main code area, You must include it in the #region parenthesis, Such as:

  ```
  #region CheckEmployeeStatus
  #endregion
  ```
- Do not declare your variables like this:

  ```
  Int  nEmployeeID = 0;
  String    sEmployeeName=   string.empty;
  Float    fPrice          = 0.0;
  XmlElement xmlel= null;
  ```

  It must be like :

  ```
  Int  nEmployeeID          = 0;
  String sEmployeeName      = string.empty;
  Float fPrice              = 0.0;
  XmlElement xmlel          = null;
  ```

- Concerning Opening objects such as database Connection or files, Do not use:

  ```
  Db.Open() or file.Read()...
  Do Statement 1
  Do Statement 2
  Do Statement .
  Do Statement N
  Db.Close() or file.Close()....
  ```

Use this bellow code template to ensure closing the object connection after being opened:

```
If ( db.Open() )
{
    try
    {
            Do Statement 1
            Do Statement 2
            Do Statement .
            Do Statement .
            Do Statement N
            Db.Close()
    }
    catch
    {
            //Always report any exception happened
            Errorlog.ReportError(Exception)
    }
    finaly
    {
            db.Close()

    }
}
```

PS, Do not keep the catch block empty, always report the error and notify the user.

- Do not use the following if statement format:

    If (bCanAddEmployee == true) or If (bCanAddEmployee == false)

  Its more professional to use:

    If (bCanAddEmployee) and If (!bCanAddEmployee)

- Always be aware of using objects properties directly, You must be sure that the object is not null, like this:

  Example 1:

```
DataSet dsEmployeesData = dsGetEmployeesData( sConnectionString )
If (dsEmployeesData != null )
{
//Your code goes here.
 }
```

  Example 2:

```
XmlDocument xmldocEmployeeInfo = new XmlDocument();
XmldocEmployeeInfo.Load( sFileName );
XmlElement xmlelEmployeeName = XmldocEmployeeInfo.GetElementsByTagNames(
“EmployeeFisrtName” )[0];
If (xmlelEmployeeName.Attributes[“ID”].InnerText == SomeVariable )
{
// Do some things.
}
```

You must validate the Attaribute if it is exist or not, like this:

```
If (xmlelEmployeeName.Attributes["ID"] != null )
If (xmlelEmployeeName.Attributes["ID"].InnerText == SomeVariable )
{
// Do some things.
}
```

- When filling a drop down list, always check if the drop down list had no items or not, if it is had no items, fill the first item with (-- not exist --, -- not found --, etc.).