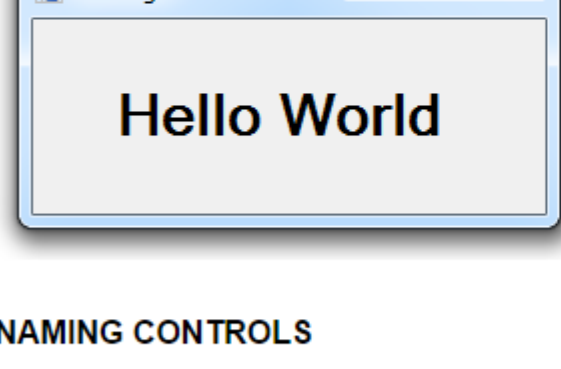


## CONTROLS

- The toolbox in C# contains many **controls**. Controls are graphical widgets that you can place on the form for a person to use. C# contains over 60 basic controls in the Controls Toolbox. **Control objects** display information or get user input. For example, a Label object displays text that cannot be changed by the user. An application interface typically contains many control objects on the Form.

- The Message application has a simple interface with one label that displays Hello World.



## NAMING CONTROLS

- Each control should be named with a prefixed shorthand identifier describing the type of control it is. This is sometimes called Hungarian Notation. This makes it much easier to understand what type of control you are working with when you are looking through the code.

Object	Prefix	Example
Button	btn	btnCancel
Check Box	chk	chkYes
Combo Box	cbo	cboLanguage
Form	frm	frmHello
Group Box	grp	grpToppings
Horizontal Scroll Bar	hsb	hsbVolume
Label	lbl	lblMessage
Link Label	lnk	lnkTSN
List Box	lst	lstCookies
Menu	mnu	mnuFile
NumericUpDown	nud	nudSize
Panel	pnl	pnlTeams
Picture Box	pic	picFLower
Progress Bar	prg	prgStep
Radio Button	rad	radEnglish
Shape	shp	shpSquare
Text Box	txt	txtFirstName
Timer	tmr	tmrAlarm
Vertical Scroll Bar	vsb	vsbSpeed

## THE WINDOWS FORM

The form is the most important graphical element in C#. All of the programs that we will write will be based on forms. Every time we create a new project, the IDE will create a basic form, which we will modify to fit our program's needs.

The code for each form is made up of two main files: the source file for all of the code that you will write and the source file that defines the graphics that will exist on your form.

The source file for your program code typically starts out as "Form1.cs". It is good practice to change this file name to the name that you have given your form.

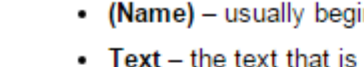
The source file for the form graphics is called "<Form Name>.Designer.cs". This file will hold information about the buttons, menus, colors and shapes on your form.

Some of the more common properties of the form are:

- (Name)** – usually begins with the prefix "frm" (which stands for "form").
- Text** – the text that is shown at the top of the form; it helps to clarify the content of the form.
- BackColor** – sets the background color of your form

The form can have code added to it. If you double-click on the form, you enter an area where code can be activated when the form loads up.

## BUTTONS



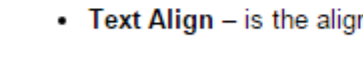
Buttons exist in just about every Windows program. Most actions that can be taken in a program are activated when the user clicks on a button.

- (Name)** – usually begins with the prefix "btn".
- Text** – the text that is displayed on the button.


### \*\*\* (Name) property \*\*\*

A proper object name begins with the appropriate prefix and then describes the purpose of the object. For example, a button that exits a program could be named btnExit.

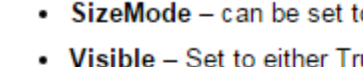
## LABELS



Labels hold the primary text that appears on a form.


- (Name)** – usually begins with the prefix "lbl".
- Text** – the text that is displayed in the label.
- Font** – contains the  button where the font name, style and size can be selected.
- AutoSize** – can be set to True or False. When True, the Label object is automatically sized to fit the text assigned in the Text property.
- Text Align** – is the alignment of the text in the label.

## PICTURE BOX



Images can make an application more interesting or improve usability. An image is added to an application by placing it in a picture box.

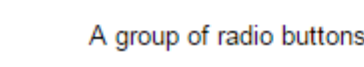
- (Name)** – usually begins with the prefix "pic".

- Image** – contains the  button that is clicked to display the Select Resource dialog box. In this dialog box, images can be added to the resource folder, it then becomes available for any picture box on the form. Right-clicking Image and then selecting Reset removes the picture from the picture box.
- SizeMode** – can be set to either: Normal, StretchImage, AutoSize, CenterImage, or Zoom.
- Visible** – Set to either True or False. Visible is often used at run time to display or hide an image.
- Size** – is the picture box size in pixels.

### Changing an image at run time

```
picAnimal.Image = Properties.Resources.dog;
```

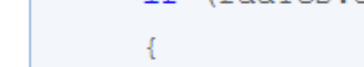
## GROUP BOX



A group box is used to group related radio buttons. A Group Box must be added to a form before adding Radio Button objects.

- (Name)** – usually begins with the prefix "grp".
- Text** – is the text displayed at the top of the group box.

## RADIO BUTTONS



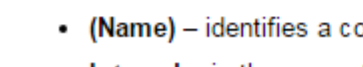
A group of radio buttons is often used in an application to enable the user to choose from a set of options. Only one radio button in a set can be selected at a time.

- (Name)** - usually begins with the prefix "rad".
- Text** – the text that is displayed next to the button.
- Checked** – can be set to either True or False to display the radio button as selected or not selected, respectively.

Writing an if statement that determines whether a radio button is checked or not is easy.

```
if (radYes.Checked == true);  
{  
    lblMessage.text = "You said YES!";  
}  
  
else if (radNo.Checked == true);  
{  
    lblMessage.text = "You said NO!";  
}
```

## TIMER



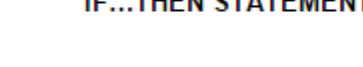
A timer is used to execute code at specific intervals.

- (Name)** – identifies a control for the programmer. Timer object names should begin with "tmr".
- Interval** – is the amount of time that passes before the event procedure is executed. Interval is specified in milliseconds between 0 and 64,767 where 1,000 milliseconds equals 1 second.
- Enabled** – setting enabled to True allows the Tick event to occur at the end of each interval, while setting it to false stops the event from occurring.

### TOGGLE A TIMER THROUGH CODE

```
tmrExample.Enabled = !tmrExample.Enabled;
```

## CHECKBOX



Check boxes allow the user to select options. Unlike radio buttons, more than one check box can be selected at a time.

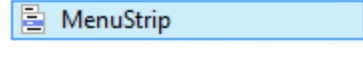
- (Name)** – identifies a control for the programmer. CheckBox object names should begin with "chk".
- (Checked)** – can be set to either True or False to display the check box with or without a check, respectively.

### IF...THEN STATEMENT

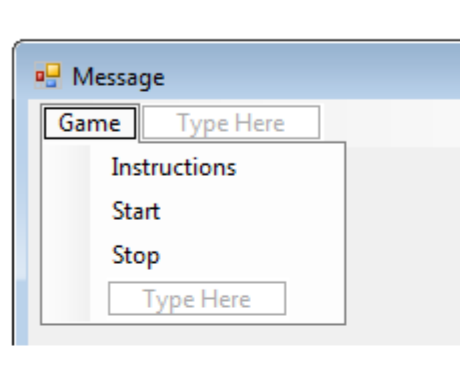
- An If...Then statement can be used in a program to determine if a check box is selected or cleared.
- For example, the following statement displays a message that depends on the state of the check box:

```
if (chkLunch.Checked == true);  
{  
    MessageBox.Show("Don't forget your bottled water!");  
}  
  
else  
{  
    MessageBox.Show("Take lunch money!");  
}
```

## MENUSTRIP



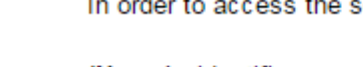
- A windows application typically includes menus that contain commands.



Using MenuStrips:

- (Name)** – Identifies an object for the programmer.
- MenuStrip names should begin with "mnu".
- MenuStrip items should begin with "mni".
- Text** – Is the menu or command name and is set when text is typed into the MenuStrip box in the Design window.

## SCROLL BARS

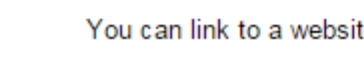


In order to access the scroll bars, you must click on All Windows Forms in the toolbox.

- (Name)** - identifies a control for the programmer. Horizontal object names should begin with "hsb". Vertical object names with "vsb".
- (Minimum)** – when the user drags the scroll box to the far left or top of a scroll bar, the scroll bar returns the value stored in the Minimum property.

- (Maximum)** - when the user drags the scroll box to the far right or bottom, the scroll bar returns the value stored in the Maximum property.
- (Small Change)** – defines how far the scroll box moves when the user presses the arrow key or clicks the scroll bar arrows.
- (Large Change)** – defines how far the scroll box moves when the user presses the Page Up or Page Down key or clicks the scroll bar area of the scroll bar.
- (Value)** – When the user chooses a value through the scroll bar, the value is stored in the scroll bar's Value property.

## LINK LABEL



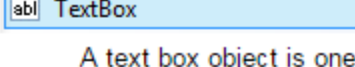
You can link to a website from your project by adding a LinkLabel object.

- (Name)** - identifies a control for the programmer. LinkLabels should begin with "lnk".
- (ActiveLinkColor)** – sets the color of the link when clicked.
- (LinkColor)** – sets the color of the link.
- (Text)** – sets the text the user will click on.
- (VisitedLinkColor)** – sets the color of the link that has been previously visited.

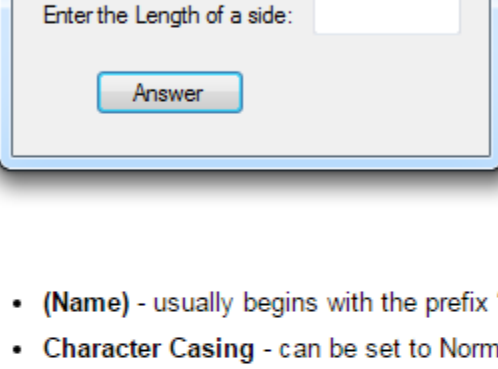
Under the click event, add these two lines of code. The first one activates the visited color and the second gets the computer to open up your browser and navigate to the address provided.

```
lnkTSN.LinkVisited = true;  
System.Diagnostics.Process.Start("http://www.tsn.ca");
```

## TEXT BOX



A text box object is one way to allow users to enter values. For example, the Area of a Square application includes a text box for the user to type a value for the length of the side:



- (Name)** - usually begins with the prefix "txt".
- Character Casing** - can be set to Normal, Upper, or Lower. Upper entries are converted to uppercase. Normal entries appear as typed and Lower entries are converted to lowercase.
- MaxLength** - can be set to numeric value indicating maximum number of characters allowed in the text box.
- Multiline** – setting this property to TRUE will enable text to appear on multiple lines.
- ReadOnly** – deciding whether a user can change the contents. Usually when a lot of text is necessary to display because they can have scroll bars and Labels cannot.
- ScrollBars** – you can add vertical or horizontal scroll bars or both. MULTILINE must be TRUE.
- Text** – is what is displayed in the text box.
- TextAlign** - sets the alignment of text-relative to the text box.

At run time, the TextBox Text property stores whatever characters are currently in the text box. This property can be used in an assignment statement to retrieve the data typed by the user.

```
string firstName = txtName.text;
```