MOVING AND ANIMATING IMAGES ACROSS THE SCREEN

To move an image across the screen, you will need to use the **Location.X** and **Location.Y** properties of a PictureBox object, which sets the coordinates of the upper-left corner of the PictureBox relative to the upper-left corner of the Form.

So, for example, if we wanted to modify the **Peter Walking** program so that Peter walks across the screen, we would do the following:

The first thing we will need to do is expand our form so that it's wider. Expand the form so that it's 800 pixels wide x 250 pixels high.

Next, we will need to modify the code in the **Timer** control so that every 100 milliseconds, Peter's x-position is going to increase by 5 pixels:

```
Private Sub tmrTimer_Tick(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles tmrTimer.Tick

    If counter < peter.Length - 1 Then
        counter += 1
    Else
        counter = 0
    End If

    picPeter.Image = peter(counter)

    picPeter.Location = New Point(picPeter.Location.X + 5,
        picPeter.Location.Y)

End Sub
```

When you run the program and click the GO button, you'll see Peter walking across the screen. The problem, however, is that he continues to walk off the screen and we never see him again.

Let's modify the program so that when Peter gets to the far-right of the screen, we will reposition him back to the left-side of the screen.

```vbnet
Private Sub tmrTimer_Tick(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles tmrTimer.Tick

    If counter < peter.Length - 1 Then
        counter += 1
    Else
        counter = 0
    End If

    picPeter.Image = peter(counter)

    If picPeter.Location.X + picPeter.Width > Me.Width + 100 Then
        picPeter.Location = New Point(0 - picPeter.Width,
            picPeter.Location.Y)
    Else
        picPeter.Location = New Point(picPeter.Location.X + 5,
            picPeter.Location.Y)
    End If

End Sub
```

Repositions Peter if he goes off the screen (annotation bracketing the second If block)

In order to reposition Peter, we need to first check if the x-position of the picture box + the width of the picture box is greater than the width of the form. (I've added an extra 100 pixels so that Peter does not get repositioned as soon as he reaches the far-right of the screen, but rather has an opportunity to go off the screen before being re-positioned). Once Peter goes off the screen, the location of the picture box gets set to 0 – the width of the picture box so that Peter gradually (instead of immediately) reappears back on the screen. The y-position of the picture box does not change.

You'll notice that In order to set the position of an object you need to create a new **Point** object which requires two values: one representing the x-position and the other representing the y-position.

In the case where Peter is still on the screen, we simply increase Peter's x-position by 5 pixels (this is the code contained in the **Else** statement).

Now run the program to see the result and then re-save the program in your UNIT 5 folder.