

<pre> if condition Then statement If condition Then statement statement End If </pre>	<pre> If mark >= 80 Then Label1.Caption = "Honour Roll" If mark >= 80 Then Label1.Caption = "Honour Roll" Label2.Caption = "Good Work" End If </pre>
<p>If-Else Case</p>	<p>Sample</p>
<pre> If condition Then statement statement Else statement End If </pre>	<pre> If mark >= 80 Then Label1.Caption = "Honour Roll" Label2.Caption = "Good Work" Else Label1.Caption = "Not Honour Roll" Label2.Caption = "Try Harder" End If </pre>
<p>If-Elseif Case</p>	<p>Sample</p>
<pre> If condition Then statement statement Elseif condition Then statement Else statement End If </pre>	<pre> If mark >= 80 Then Label1.Caption = "Honour Roll" Label2.Caption = "Good Work" Elseif mark >= 70 Then Label1.Caption = "Not Honour Roll" Label2.Caption = "Nice Try" Else Label1.Caption = "Not Honour Roll" Label2.Caption = "Try Harder" End If </pre>

The Form Load Sub

Every program we have written up until now has depended upon the user to initiate some event by clicking on a button or other control.

When a VB project starts up there are a number of events that occur even before the user has a chance to click on any controls.

One of these events is a form load event.

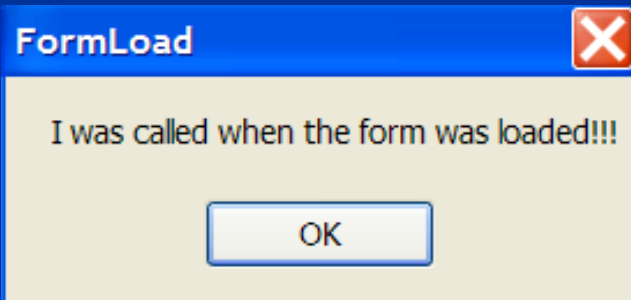
By double-clicking on the form at design time we can insert code that will run as soon as the program starts up.

Variables can be declared and assigned values that will have a scope that includes all controls within the form.

Any code that is placed in the FormLoad Sub will run without user interaction as soon as the program is run.

Form Load Code

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal  
    MsgBox("I was called when the form was loaded!!!")  
End Sub
```



This Message Box pops up even before the form of the program becomes visible.

The form doesn't get drawn until after the user clicks OK

String Compare Using =

```
Private Sub Form1_Load(ByVal sender As
    Dim password As String = "secret"
    Dim input As String

    input = InputBox("Enter Password")

    If (password = input) Then
        MsgBox("Welcome")
    Else
        MsgBox("Incorrect")
        Me.Close()
    End If
End Sub
```

Value of the variable 'password' is compared with value of variable 'input'. If two String variables are equal then the operation evaluates as true.

This code closes the form and effectively shuts down the program. The 'Me' keyword is VB's short cut for a form to refer to itself. Form1.Close() would have done the same thing.

String Compare Using .equals()

```
Private Sub Form1_Load(ByVal sender As  
    Dim password As String = "secret"  
    Dim input As String  
  
    input = InputBox("Enter Password")  
  
    If (password.Equals(input)) Then  
        MsgBox("Welcome")  
    Else  
        MsgBox("Incorrect")  
        Me.Close()  
    End If  
End Sub
```

Value of the variable 'password' is compared with value of variable 'input'. If two String variables are equal then the operation evaluates as true.

This code closes the form and effectively shuts down the program. The 'Me' keyword is VB's short cut for a form to refer to itself. Form1.Close() would have done the same thing.

Using the <> Not Equal Operator

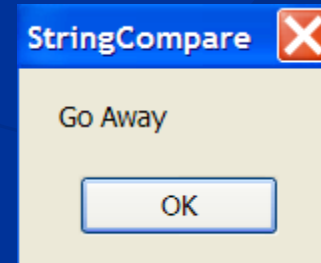
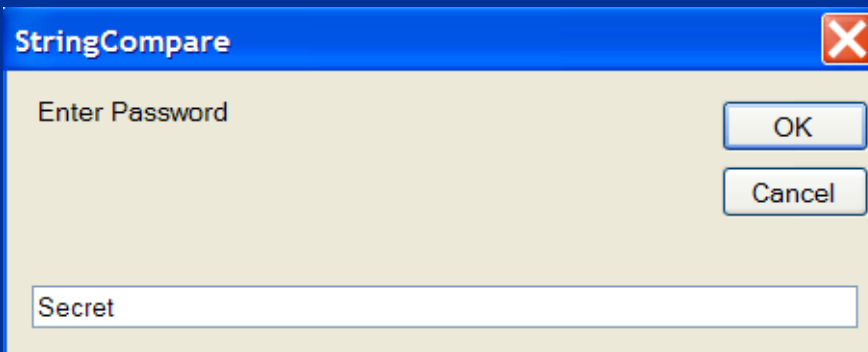
The process can be reversed so that as long as input does not equal password then the message will read "Go Away".

```
Dim password As String = "secret"  
Dim input As String  
  
input = InputBox("Enter Password")  
  
If (password <> input) Then  
    MsgBox("Go Away")  
Else  
    MsgBox("Welcome")  
End If
```

Using toUpper and toLower Functions

In instances where the user's input of capital and lower case might interfere with the smooth operation of a program it is possible to take the input and convert it to all lower case or upper case.

Consider the following example where the password is "secret" but the user capitalizes the first letter of the input



How rude!

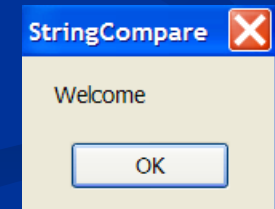
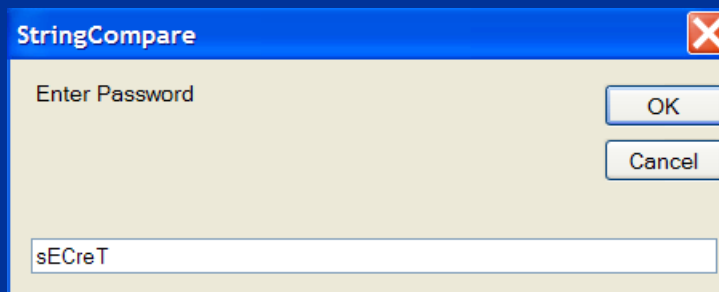
Convert to Upper Case

```
Dim password As String = "SECRET"  
Dim input As String  
  
input = InputBox("Enter Password")  
input = input.ToUpper  
  
If (password <> input) Then  
    MsgBox("Go Away")  
Else  
    MsgBox("Welcome")  
End If
```

Code the String literal in all capital letters

input gets assigned the previous value of input but capitalized.

Now, no matter what case the user uses the input will be converted to upper case and accepted. We say the password is **not** case-sensitive.



The toLower Function

Modify the previous program so that the input is converted to lower case and the password input is again case-insensitive.

Nested If Statements

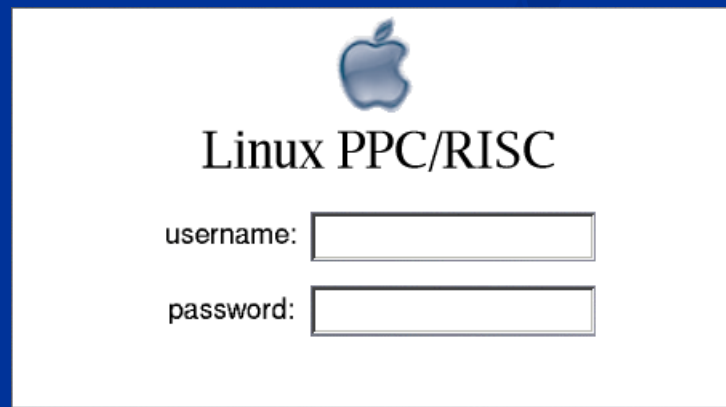
The sad and one-sided romantic affair between
Angelina Jolie and

Login and Password

A login sequence often involves the use of a username and a password. In order to gain access both must be correct.

First the system will test if a username is valid then it will test to see if the correct password associated with that username has been entered.

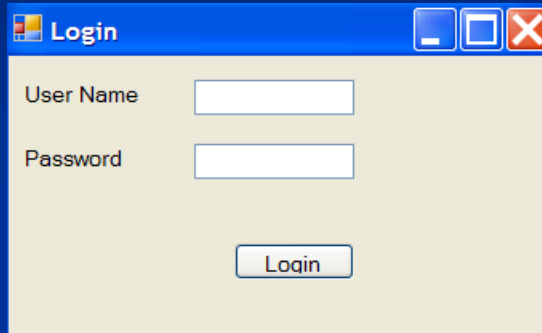
This two step process can be represented in code using a structure called a nested if statement



A screenshot of a login screen for Linux PPC/RISC. At the top center is the Apple logo. Below it, the text "Linux PPC/RISC" is displayed. Underneath, there are two input fields: "username:" followed by a rectangular box, and "password:" followed by another rectangular box.

Nested If Statements

Assume that you have a login 'homer' and a password 'secret'. The program tests to see if the correct username has been entered.



```
Dim userNameAttempt As String
Dim passwordAttempt As String
Dim userName As String = "homer"
Dim password As String = "secret"

userNameAttempt = txtUserName.Text
passwordAttempt = txtPassword.Text
```

The First Test

“....and now my young apprentice, you must pass the first test.”

Think of a 'nested if' statement as a 2 part test. Only if you can get by the first part, do you get to attempt the second.



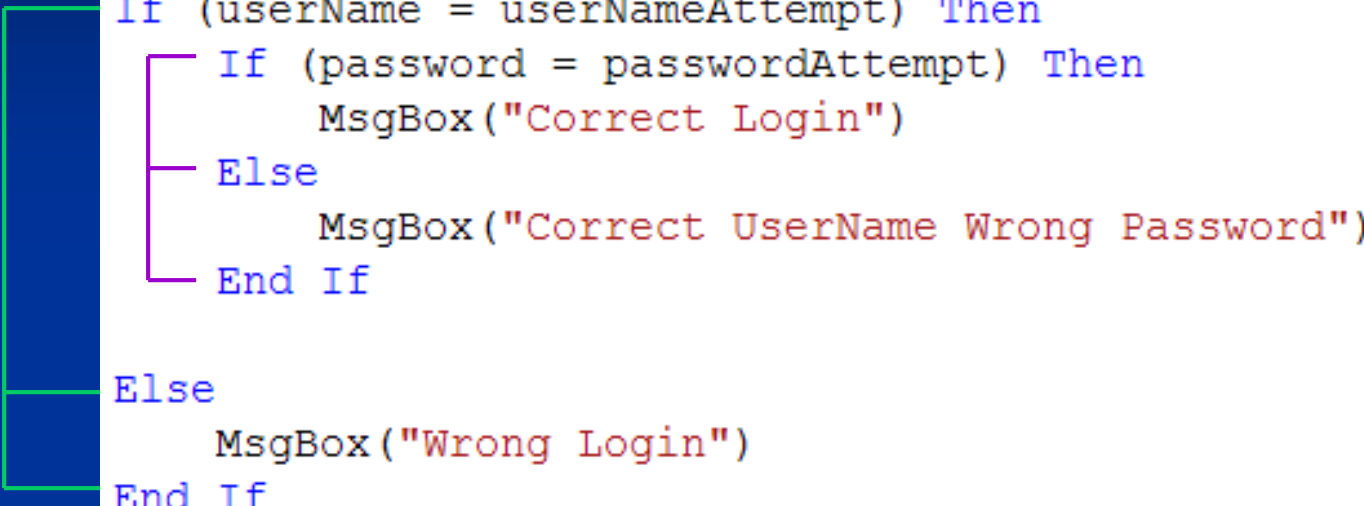
```
If (userName = userNameAttempt) Then  
Else  
End If
```

If this part evaluates to true you go here.

Otherwise you end up here

The Second Test

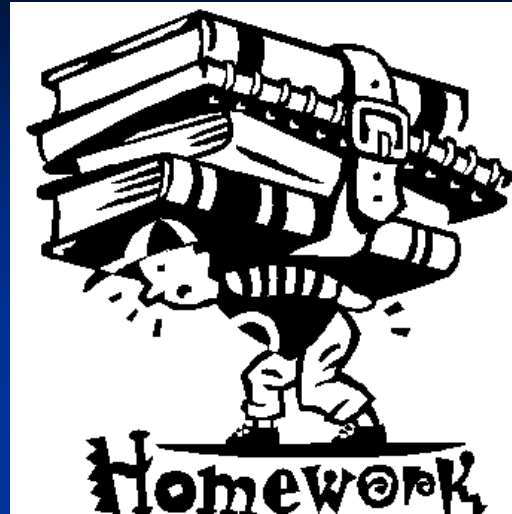
```
If (userName = userNameAttempt) Then
  If (password = passwordAttempt) Then
    MsgBox("Correct Login")
  Else
    MsgBox("Correct UserName Wrong Password")
  End If
Else
  MsgBox("Wrong Login")
End If
```

A diagram illustrating nested if statements. The code is displayed on a white background. The outer if statement is highlighted with green lines, and the inner if statement is highlighted with purple lines. The code is as follows:

```
If (userName = userNameAttempt) Then
  If (password = passwordAttempt) Then
    MsgBox("Correct Login")
  Else
    MsgBox("Correct UserName Wrong Password")
  End If
Else
  MsgBox("Wrong Login")
End If
```

The green lines indicate the elements of the outer if statement while the purple lines show the elements of the inner or nested if statement.

Homework



Modify the previous program so that all information is submitted using Input Boxes instead of Text Boxes.

Also make the username and password case-insensitive .