

```
'Dan Morgan - VisuMAX Inc.  
'script makes use of a check digit and Luhn algorithm to verify and validate the  
'accuracy of credit card numbers. Little messy but it does the trick. You can  
'of course add other credit cards too.  
'Use or don't ;-)
```

```
Partial Class _Default  
    Inherits System.Web.UI.Page  
    Dim CardType As String  
  
    Protected Sub btnValidate_Click(ByVal sender As Object, ByVal e As System.EventArgs) ✎  
        Handles btnValidate.Click  
        Dim CCNumber As Long, LenCCNum As Integer, RevCCNumber As String, cardval As ✎  
String, cardint As Integer  
        Dim CCVal As Integer, ccmmod As Integer  
  
        Try  
            lblCardType.Text = "" 'Set var to null  
            lblValid.Text = ""  
            lblCCVal.Text = ""  
  
            CCNumber = 0  
            RevCCNumber = 0  
            LenCCNum = 0  
            CCVal = 0  
  
            CCNumber = txtCardNumber.Text  
            RevCCNumber = ReverseString(CCNumber) 'Call Reverse stringPrep number for ✎  
algorithm by reverseing it  
  
            LenCCNum = Len(txtCardNumber.Text) 'Get length of the number  
  
            CCVal = ParseArrayListString(RevCCNumber, LenCCNum) 'Call Parse to check card ✎  
against algorithm  
  
            ccmmod = CCVal Mod 10 'Check returned value  
  
            If ccmmod = 0 Then 'Card has a valid account number  
  
                cardval = txtCardNumber.Text.Substring(0, 4) 'Determine card type  
                cardint = Val(cardval)  
  
                If cardint > 3999 And cardint < 5000 Then  
                    CardType = "VISA:"  
                ElseIf cardint > 3399 And cardint < 3500 Then  
                    CardType = "AMEX:"  
                ElseIf cardint > 3699 And cardint < 3800 Then  
                    CardType = "AMEX:"  
                ElseIf cardint > 5099 And cardint < 5600 Then  
                    CardType = "MasterCard:"  
                ElseIf cardint = 6011 Then  
                    CardType = "Discovery:"  
                Else  
                    CardType = "Not Valid:" 'Card in not Visa, MC, Disc, or Amex  
                End If  
  
                lblCardType.Text = CardType  
                lblValid.ForeColor = Drawing.Color.Green  
  
                lblValid.Text = "VALID Number"  
  
                ElseIf ccmmod <> 0 Then 'Card does not have a valid account number  
  
                cardval = txtCardNumber.Text.Substring(0, 4) 'Determine card type  
                cardint = Val(cardval)  
  
                If cardint > 3999 And cardint < 5000 Then
```

```

        CardType = "VISA:"
    ElseIf cardint > 3399 And cardint < 3500 Then
        CardType = "AMEX:"
    ElseIf cardint > 3699 And cardint < 3800 Then
        CardType = "AMEX:"
    ElseIf cardint > 5099 And cardint < 5600 Then
        CardType = "MasterCard:"
    ElseIf cardint = 6011 Then
        CardType = "Discovery:"
    Else
        CardType = "Not Valid:" 'Card in not Visa, MC, Disc, or Amex
    End If
    lblValid.ForeColor = Drawing.Color.Red
    lblValid.Text = "INVALID Number"
    lblCardType.Text = CardType

```

```
End If
```

```

Catch ex As Exception
    lblCCVal.Text = "Enter Only Numbers with no spaces!"
    txtCardNumber.Text = ""
End Try

```

```
End Sub
```

```
Function ReverseString(ByVal CCText As String) As String
```

```
    Dim lenText As Long, ChrPos As Long
```

```
    If Len(CCText) = 0 Then Exit Function 'Not data - exit
```

```
    lenText = Len(CCText) 'Get length
    ReverseString = Space(lenText) 'Set string with spaces = to length

```

```

    For ChrPos = lenText To 1 Step -1 'Loop through and peel off last char and set to
front of string
        Mid$(ReverseString, lenText - ChrPos + 1, 1) = Mid$(CCText, ChrPos, 1)
    Next ChrPos

```

```
Return ReverseString 'Pass back the new string
```

```
End Function
```

```
Function ParseArrayListString(ByVal CCNum As String, ByVal LCCNum As Integer) As String
```

```

    Dim i As Integer, j As Integer, chr As Char, chrInner As Char, number As Integer
    Dim flag As Boolean, ccdigits As Integer, CCpart As String, cc As Integer

```

```
Dim CCArrayList As New ArrayList 'Declare an arraylist
```

```
If LCCNum = 0 Then Exit Function 'Length of value passed is 0
```

```
CCArrayList.Capacity = LCCNum 'Sets the number of elements using the cc lenght
passed
```

```
flag = False 'set flag to calculate every other digit for cc varification
```

```

For i = 0 To LCCNum - 1 'Substract 1 - 0 based arraylist
    chr = CCNum.Chars(i) 'strips one char off the string
    If flag = True Then
        number = Val(chr) * 2 'multiply digit to check validation
        If number > 9 Then 'number is multi digit
            CCpart = number.ToString 'convert to string
            ccdigits = Len(CCpart) 'Get length of CCpart
            For j = 0 To ccdigits - 1

```

```
        chrInner = CCpart.Chars(j) 'strip char off string
        number = Val(chrInner) 'convert back to val
        CCArrayList.Add(number) ' increase and add number to array
    Next
    ElseIf number < 10 Then 'number is single digit
        CCArrayList.Add(number) 'writes that char to the array
    End If
    flag = False
    ElseIf flag = False Then
        number = Val(chr) 'convert to val
        CCArrayList.Add(number) 'writes that char to the array
        flag = True
    End If
Next

For i = 0 To CCArrayList.Count - 1 'Loop through the list and calc values
    cc = CCArrayList(i) + cc
Next

Return cc 'Return total

End Function

End Class
```