Program 1

The Purpose of the application is to calculate the interest of different currencies. You type in the numbers you want to multiply and it gives you the result by multiplying them together. This program was mainly made to assist calculations when converting currencies. This program could also be just used to multiply things together; the results would be the same.

In this program they used a technique called Data declaration so they can simply write there code more efficiently without get everything muddled up, here's an example of the data declaration:

```
Dim amount, rate, result As Decimal
```

You can see that it's dimensioning the Amount rate and result as a variable type Decimal

It is then used elsewhere in the code, this makes the data become easily accessible,

```
result = amount * rate
```

Program 2

The purpose of this program is to find out how big of a bonus you made from your sales total. You type in your sales total and it comes up with how much bonus you achieved, this program was mainly made to assist businesses or a salesmen on working out the amount of their bonus.

The code, behind the scenes have used IF statement, here's an example of the Syntax for an IF statement

If *example > 9* Then

code

EndIf

You can see that this is a simple structure which is used in the code to determine what the program should do, what this code is basically applying is that *If the statement is greater than 9 than* use this code, then End the program. You could also make the program do another action or another statement by simply say ELSE *something else ENDIF*

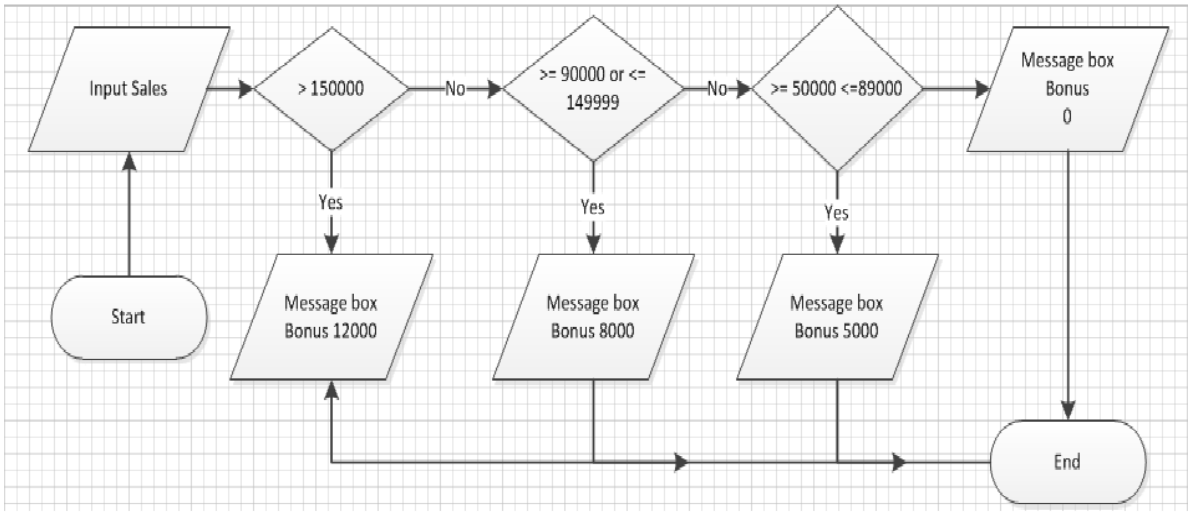
Visual Basics is Object orientated which means you have to think about it visually and think about the objects more than the procedures. Visual Basics is also event driven which means you code to make events for objects, an example would be making a button and coding the event (what happens) in the background. Another characteristic would be the form designer. On the form designer you can drag and drop certain objects such as buttons, resize the actual form and make different various changes on the form designer with the toolbox.

The code used in Program two is efficient because they used event handling correctly. Firstly they made the objects such as the buttons then they clicked on them to add events to the objects. Secondly, they used IF Statements to find the Bonus achieved which was the most efficient way to do it because they needed a Structure where it would access the code it needs whenever the statement was true which in this case would be greater than 150000, it would than pop-up with a message box depending on if the statement was true or not.

The code, in program two, is as fast as it could be because they are following the structure of an if statement and there's no other way of calculating the bonus rather than doing it this way. In terms of performance I would say that it's pretty fast even though there's a lot of behind the scenes code.

Program two is very reliable because there's no way of the program crashing (unless you have to many tasks running and even then it's unlikely) It's been coded in a way where no matter what it will come up with a bonus even if you type nothing, it will still come up with a message box because they've ended the code on

```
Else
MsgBox("Bonus Achieved: 0")
End If
```



There are a couple of improvements that I would make to this program. The first improvement I would make to the program is I would make it so that if they submitted nothing it would come up with a popup saying that no text and would stay in the same state it was originally in, rather than getting a popup saying that "Bonus Achieved: 0" because in a way it can be very misleading especially if they didn't know that they didn't type anything.

The second improvement I would make is make it so it has a clear button so you don’t have to delete what you’ve put in before, this would make it easier especially for a business to work out not only there Bonuses but also other workers.

There are a couple weaknesses to this program, it functions properly and does what its suppose to but it’s not very user friendly. Some people might find it hard to use just because of the lack of functionality, there’s no Clear button so it makes deleting what you’ve wrote before time consuming. Another weakness would be the form takes up to much of the screen. It doesn’t need to cover that much especially if the application was made to work out the total Bonus. Another weakness would be the misleading popup saying that the bonus achieved was 0 at the end of the if statement.

One strength of program two is that the code is very organized and has been laid out correctly so other people can read the code instead of trying to figure out what they have done means and have to find various different pieces of code to work out what happens in the program. Another strength would be that they have used meaningful variables they didn’t just go and call there variable names something like "sqwdswd" because that doesn’t mean anything to anyone and it would quickly get confusing, instead they named them something meaningful like "txtSales" which obviously means more to us as humans than a load of random letters.

```
Imports System.IO
Public Class Form1
    'Dimensions
    Dim streamread As StreamReader
    Dim streamread2 As StreamReader
    Dim Clicked As Boolean = False
    Dim b As String
    Dim c As String = "/currency.txt"
    Dim d As String = "/currency2.txt"
    Dim e As String
    Dim f As String
    Dim Currency As String
    Dim ExchangeRate As String
    Dim answer As Integer
    'New Function called "ReadFile
    Private Sub ReadFile()
        Try
            cmbCurrency.Items.Clear()
            'Opens File C
            streamread = File.OpenText(c)
            While streamread.Peek <> -1
                'Reads line Of File C, Stores text inside B
                b = streamread.ReadLine()
                'Adds text stored inside B to cmbCurrency 1 & 2 (ComboBoxes)
                cmbCurrency.Items.Add(b)
                cmbCurrency2.Items.Add(b)
            End While
            streamread.Close()
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub
    Private Sub ReadFile2()
        Try
            cmbExchangeRate.Items.Clear()
            'Opens File D
            streamread2 = File.OpenText(d)
            While streamread2.Peek <> -1
                'Reads line Of File D, Stores text inside E
                e = streamread2.ReadLine()
                'Adds text stored inside E to cmbExchangeRate (ComboBoxes)
                cmbExchangeRate.Items.Add(e)
            End While
            streamread2.Close()
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        If Clicked = False Then
            cmbAnswer.Enabled = False
            btnConvertAgain.Enabled = False
            cmbCurrency2.Enabled = False
        Else
            cmbAnswer.Enabled = True
            btnConvertAgain.Enabled = True
            cmbCurrency2.Enabled = True
```

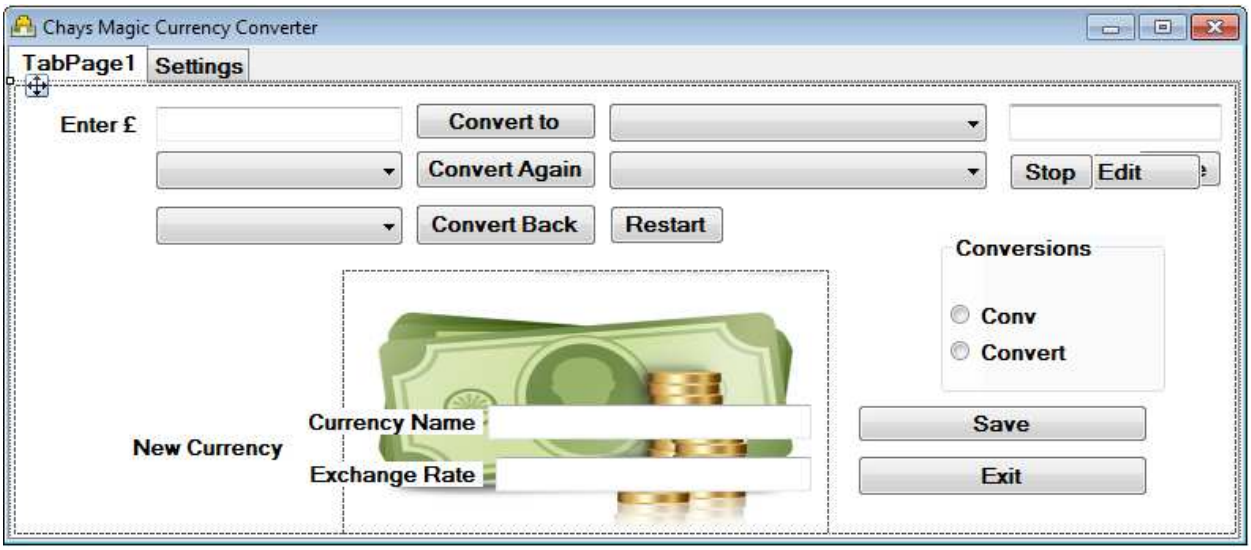
```
        End If
        txtNew.Visible = False
        ' Creates text file if it doesnt exsist on Form Load
        If Not File.Exists(c) Then
            File.Create(c)
        End If
        'Reads currency.txt
        ReadFile()
        ' Creates text file if it doesnt exsist on Form Load
        If Not File.Exists(d) Then
            File.Create(d)
        End If
        'Reads currency2.txt
        ReadFile2()
        'cmbAnswer is selected on "None" because answer not selected
        cmbAnswer.SelectedIndex = 0
        'Convert is selected as default option
        rbnConvert.Checked = True
        'Convert Again button is disabled if answer isnt selected

        'Tab 1 name is Conversions
        Conversions.TabPages(0).Text = "Conversions"
        'Tab 2 name is Settings
        Conversions.TabPages(1).Text = "Settings"
    End Sub
    Private Sub btnConvert_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnConvert.Click
        cmbAnswer.Enabled = True
        btnConvertAgain.Enabled = True
        cmbCurrency2.Enabled = True

        Dim ans As Integer
        'Entered text times the selected exchange rate
        Try
            ans = txtEnter.Text * cmbExchangeRate.SelectedItem
        Catch ex As Exception
            MessageBox.Show("Please enter a number")
            txtEnter.Text = ""
        End Try
        'Displays the answer in a label
        lblDisplay.Text = ans
        'Adds the answer into 2 comboboxs which become assessible
        cmbAnswer.Items.Add(ans)
        cmbAnswer2.Items.Add(ans)
        'Displays the index of the selected item in exchange rate
        lblDisplay.Text &= vbNewLine & "Selected Item " & cmbExchangeRate.SelectedIndex
        'when the button "Convert" is pressed the Answer combobox changes to the exchange
rate beforehand
        cmbAnswer.SelectedIndex = 1
        'Convert again option is disabled when the "Answer" combobox matains untouched
        '  If cmbAnswer.SelectedIndex = 0 Then
        'btnConvertAgain.Enabled = False
        'ElseIf cmbAnswer.SelectedIndex >= 0 Then
        'btnConvertAgain.Enabled = True
        'End If
    End Sub
    Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnAdd.Click
```

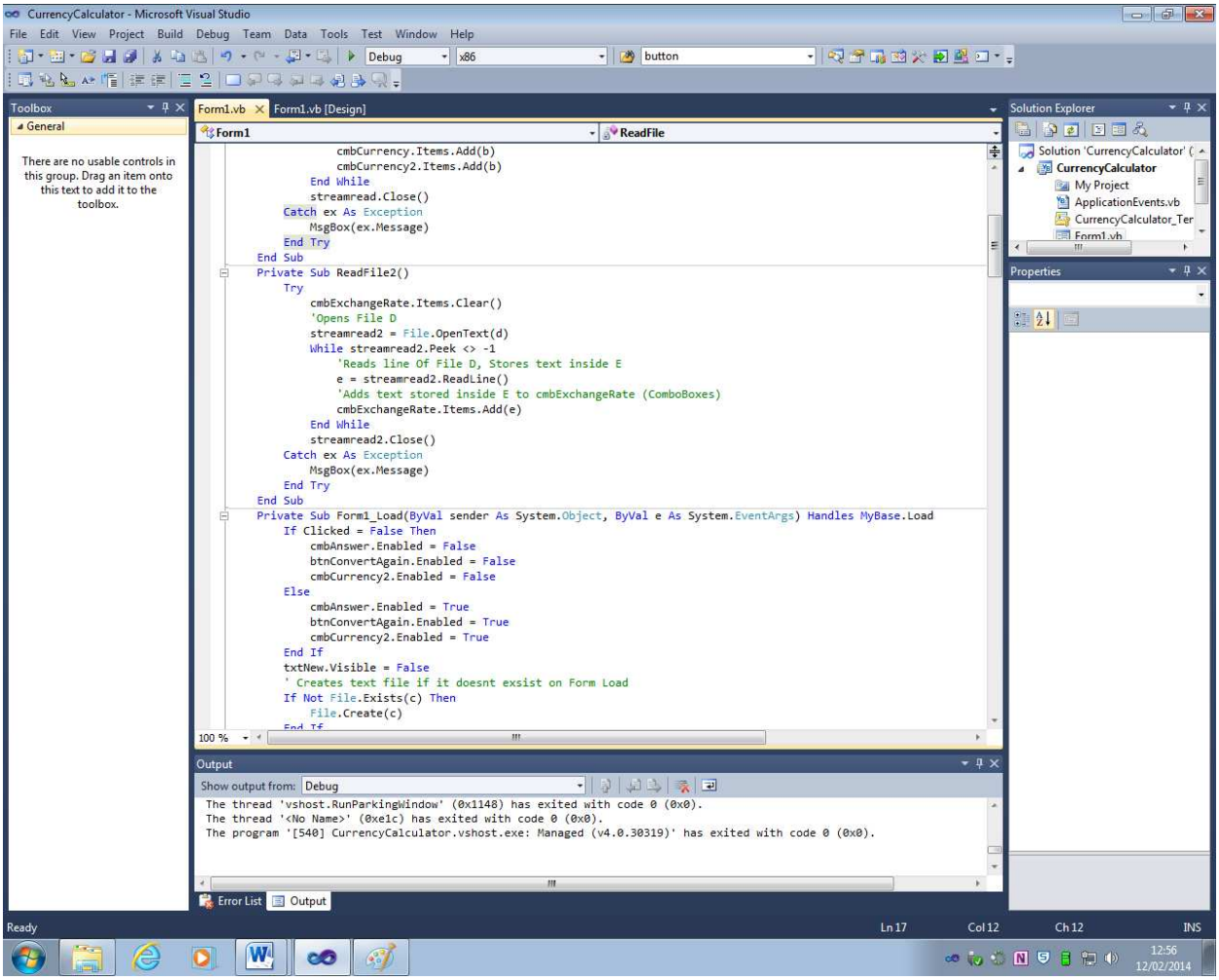
```
        'makes controls visable and unvisble when new currency is added
        If btnAdd.Text = "Add New Currency" Then
            lblNewCurrency.Visible = True
            lblExchangeRate.Visible = True
            lblCurrencyName.Visible = True
            txtNameCur.Visible = True
            txtExchangeRa.Visible = True
            btnAdd.Visible = False
            btnFinish.Visible = True
            btnExit.Visible = True
            picMoney.Visible = False
        End If
    End Sub
    Private Sub btnFinish_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFinish.Click
        'hinds controls and shows controls when the user exits "adding new currency
        lblNewCurrency.Visible = False
        lblExchangeRate.Visible = False
        lblCurrencyName.Visible = False
        txtNameCur.Visible = False
        txtExchangeRa.Visible = False
        btnFinish.Visible = False
        btnAdd.Visible = True
        btnExit.Visible = False
        picMoney.Visible = True
    End Sub
    Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnExit.Click
        'saves currency and hides/unhides controls needed
        Dim Currency As String
        Dim ExchangeRate As String
        Currency = txtNameCur.Text
        ExchangeRate = txtExchangeRa.Text
        lblNewCurrency.Visible = False
        lblExchangeRate.Visible = False
        lblCurrencyName.Visible = False
        txtNameCur.Visible = False
        txtExchangeRa.Visible = False
        btnFinish.Visible = False
        btnAdd.Visible = True
        btnExit.Visible = False
        'Adds currency name to currency.txt
        File.AppendAllText(c, Currency & vbCrLf)
        'Adds ExchangeRate to currency2.txt
        File.AppendAllText(d, ExchangeRate & vbCrLf)
        'Clears textBoxes
        txtExchangeRa.Text = ""
        txtNameCur.Text = ""
        'Tells the user all the information has been saved
        MsgBox("All Saved", MsgBoxStyle.Information, "Saved")
        'Restarts program so the new exchange rate and currency name is saved together in
comboboxes
        Application.Restart()
    End Sub
    Private Sub btnStop_Click(ByVal sender As Object, ByVal e As EventArgs) Handles
btnStop.Click
        'Stops editing button
```

```
        cmbExchangeRate.Enabled = False
        txtNew.Visible = False
        btnEditBox.Visible = True
        btnEndEdit.Visible = False
        btnStop.Visible = False
    End Sub

    Private Sub btnEditBox_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEditBox.Click
        txtNew.Text = cmbExchangeRate.SelectedItem
        txtNew.Visible = True
        'Allows the User to change the exchange vaule inside the combobox
        cmbExchangeRate.Enabled = True
        'Allows editing / ends editing
        btnEditBox.Visible = False
        btnEndEdit.Visible = True
        btnStop.Visible = True
    End Sub
    Private Sub btnEndEdit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEndEdit.Click
        'Edits exsiting Currency
        Dim thefile As String = "C:\Users\Chay\Desktop/currency2.txt"
        Dim lines() As String =
System.IO.File.ReadAllLines("C:\Users\Chay\Desktop/currency2.txt")
        Dim q As String = cmbCurrency.SelectedIndex
        txtNew.Visible = False
        'stops it from being enabled so you cant type in combobox
        cmbExchangeRate.Enabled = False
        Try
            lines(q) = txtNew.Text
        Catch ex As Exception
            MessageBox.Show("Something went wrong")
        End Try
        System.IO.File.WriteAllLines(thefile, lines)
        btnEditBox.Visible = True
        btnEndEdit.Visible = False
        Application.Restart()
    End Sub
    Private Sub btnConvertAgain_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnConvertAgain.Click
        Dim ans As Integer
        'Answer times exchange rate to convert again
        ans = cmbAnswer.SelectedItem * cmbExchangeRate.SelectedItem
        'displays answer when been after second conversion
        lblDisplay.Text = ans
        'displays what item on combobox selected
        lblDisplay.Text &= vbNewLine & "Selected Item " & cmbExchangeRate.SelectedIndex
        'Adds items to Answer box
        cmbAnswer.Items.Add(ans)
        'Adds items to Answer box 2
        cmbAnswer2.Items.Add(ans)
    End Sub
    Private Sub btnConvertBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnConvertBack.Click
        'the answer divided by the exchange rate is sotred in answer
        answer = cmbAnswer2.SelectedItem / cmbExchangeRate.SelectedItem
        'displays answer
        lblDisplay.Text = answer
```

```
        'adds items to comboboxes for more possible conversion
        cmbAnswer.Items.Add(answer)
        cmbAnswer2.Items.Add(answer)
    End Sub
    Private Sub btnRestart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRestart.Click
        'clears items from answer boxes
        cmbAnswer2.Items.Clear()
        cmbAnswer.Items.Clear()
        'restart applications
        Application.Restart()
        End
    End Sub
    Private Sub rbnConvert_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
        'If the convert button clicked then soem controls will be visible
        If Clicked = False Then
            cmbAnswer.Enabled = False
            btnConvertAgain.Enabled = False
            cmbCurrency2.Enabled = False
        Else
            cmbAnswer.Enabled = True
            btnConvertAgain.Enabled = True
            cmbCurrency2.Enabled = True
        End If
        lblEnter.Enabled = True
        txtEnter.Enabled = True
        btnConvert.Enabled = True
        cmbCurrency.Enabled = True
        cmbExchangeRate.Enabled = False
        btnEditBox.Enabled = True
        ' btnConvertBack.Enabled = False
        ' cmbAnswer2.Enabled = False
    End Sub
    Private Sub rbnConvertBack_CheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs)
        'hides and shows controls when user selects convert back
        If Clicked = False Then
            cmbAnswer.Enabled = False
            btnConvertAgain.Enabled = False
            cmbCurrency2.Enabled = False
        Else
            cmbAnswer.Enabled = True
            btnConvertAgain.Enabled = True
            cmbCurrency2.Enabled = True
        End If
        lblEnter.Enabled = False
        txtEnter.Enabled = False
        btnConvert.Enabled = False
        cmbCurrency.Enabled = False
        cmbExchangeRate.Enabled = True
        cmbAnswer.Enabled = False
        btnConvertAgain.Enabled = False
        cmbAnswer2.Enabled = True
        cmbCurrency2.Enabled = False
        btnEditBox.Enabled = False
        btnConvertBack.Enabled = True
    End Sub
```

```
    Private Sub cmbCurrency_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles cmbCurrency.SelectedIndexChanged
        'make them have the same selected index
        cmbExchangeRate.SelectedIndex = cmbCurrency.SelectedIndex
    End Sub

    Private Sub cmbCurrency2_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles cmbCurrency2.SelectedIndexChanged
        cmbExchangeRate.SelectedIndex = cmbCurrency2.SelectedIndex
    End Sub
    Private Sub cmbExchangeRate_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles cmbExchangeRate.SelectedIndexChanged
        cmbCurrency.SelectedIndex = cmbExchangeRate.SelectedIndex
    End Sub


End Class
```




CurrencyCalculator - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Tools Test Window Help
Debug
x86
button
Toolbox
General
There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.
Form1.vb
Form1.vb [Design]
Form1
ReadFile
cmbCurrency.Items.Add(b)
cmbCurrency2.Items.Add(b)
End While
streamread.Close()
Catch ex As Exception
MsgBox(ex.Message)
End Try
End Sub
Private Sub ReadFile2()
Try
cmbExchangeRate.Items.Clear()
'Opens File D
streamread2 = File.OpenText(d)
While streamread2.Peek <> -1
'Reads line Of File D, Stores text inside E
e = streamread2.ReadLine()
'Adds text stored inside E to cmbExchangeRate (ComboBoxes)
cmbExchangeRate.Items.Add(e)
End While
streamread2.Close()
Catch ex As Exception
MsgBox(ex.Message)
End Try
End Sub
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
If Clicked = False Then
cmbAnswer.Enabled = False
btnConvertAgain.Enabled = False
cmbCurrency2.Enabled = False
Else
cmbAnswer.Enabled = True
btnConvertAgain.Enabled = True
cmbCurrency2.Enabled = True
End If
txtNew.Visible = False
' Creates text file if it doesnt exsist on Form Load
If Not File.Exists(c) Then
File.Create(c)
End If
100 %
Output
Show output from: Debug
The thread 'vshost.RunParkingWindow' (0x1148) has exited with code 0 (0x0).
The thread '<No Name>' (0xe1c) has exited with code 0 (0x0).
The program '[540] CurrencyCalculator.vshost.exe: Managed (v4.0.30319)' has exited with code 0 (0x0).
Error List
Output
Solution Explorer
Solution 'CurrencyCalculator'
CurrencyCalculator
My Project
ApplicationEvents.vb
CurrencyCalculator_Ter
Form1.vb
Properties
Ready
Ln 17
Col 12
Ch 12
INS
12:56
12/02/2014

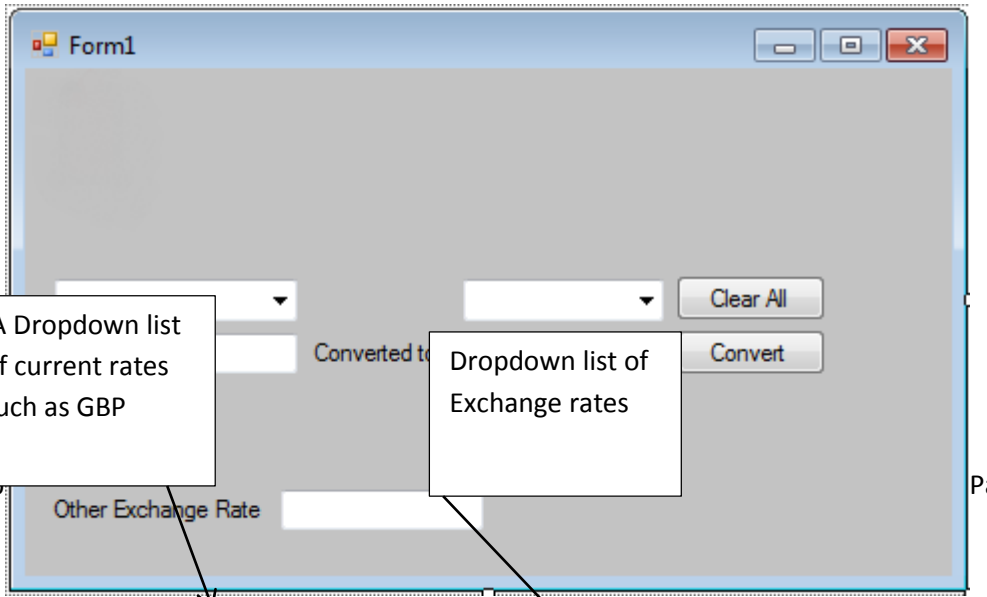
Assignment Planning

The purpose of the program is to help the local travel agent calculate currency exchange for its customers.

The program needs to be able to enter an amount in another currency such as Euro; they need to be able to calculate the equivalent in British pounds. The program also needs to be able to allow the user to alter the exchange rates used to match the current rates.

The business could be experiencing many issues such as small mistakes that make big issues in the company, people having bad maths and having no idea if the calculations are right or wrong because they can't check because they have no currency converter to check.

Inputs	Process	Outputs
Amount in British pounds	Pounds x Exchange Rate	**Amount in foreign currency**
Amount in other currency **(example: Euro)**	Pounds / Exchange Rate	The equivalent in British pounds



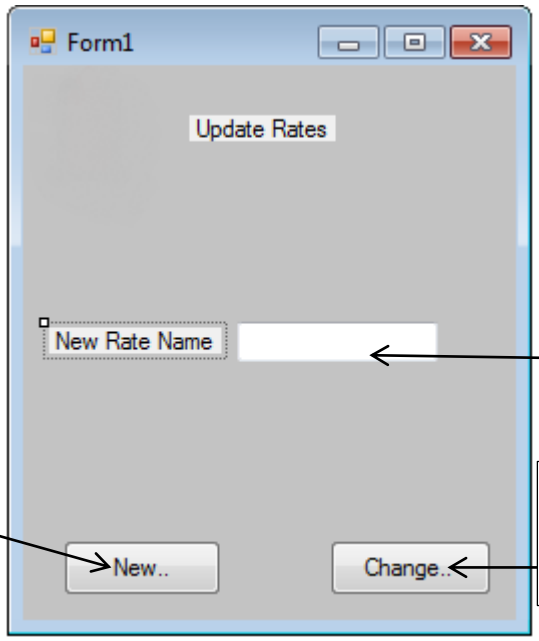
This will be the button which shows the new currency

This is where the amount of pounds will be entered

This is where amount will be shown in a different currency

Additional box to type any other exchange rates.

Additional form to change exchange rates

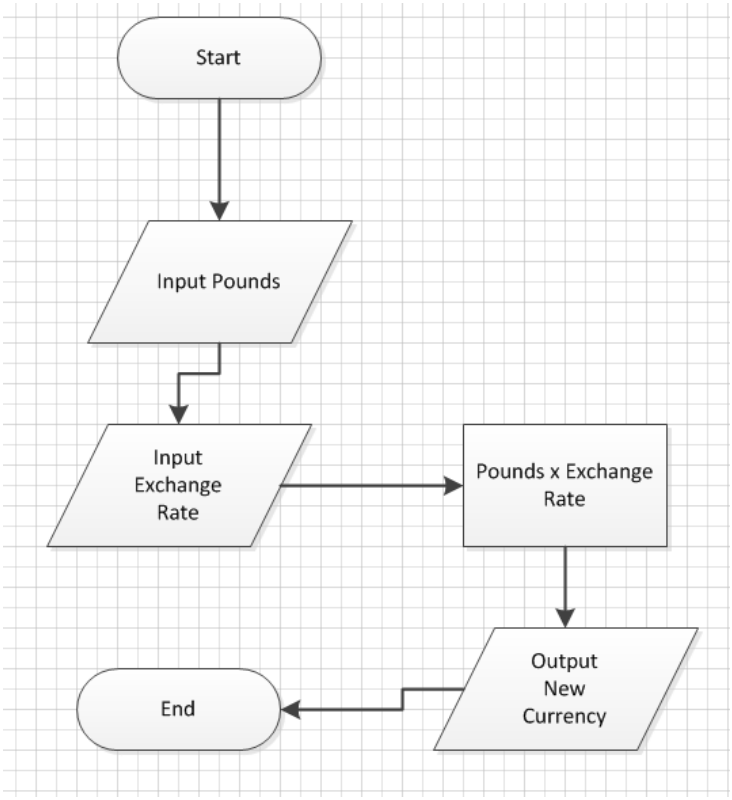


Text box for rate name

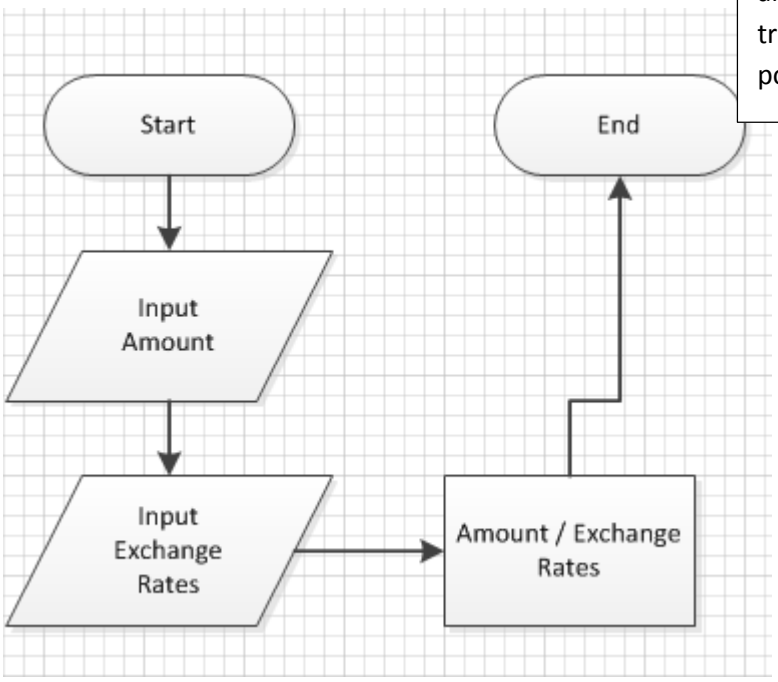
Add any additional rates

Change default Rates

Flowchart of the British pounds converting to the new currency.



Flowchart of the amount being transferred back into pounds



When creating my program I am going to encounter many restrictions on input/output fields this includes the text box that will be used to type the amount in pounds into. If

the user types in “123q” on their keyboard it will crash the program this is because the conventional field hasn’t been written in a way where it does something about this problem. Say I wanted to type in an exchange rate, as you already know the exchange rate can only be numerical, and I made it so that it was “aa” if I then multiplied this by the amount in pounds the program will crash this is because it is not possible to multiply aa by amount in pounds. If the program crashes there are many techniques which can be used to prevent errors. To prevent the program from crashing it’s suitable to use structured exception handling which can used to stop the program from crashing and to stop errors. In my program I will definitely use this structure to prevent crashes in my program.

Here’s an example of structured exception handling which deals with restrictions on input fields:

```
Try

// code that might go wrong would go here

Catch ex  as Exception

// if something goes wrong this is where you say how to stop the program from
Crashing or how you fix the error

Finally

// this is where your able to use cleanup code

End Try
```

In my program I will use arrays to store data. An array is a set of values which are logically related with each other in a certain way, an example of an array could be how many people are lined up at lunch time. When using arrays you can get different values with the same name buts what’s different about the arrays is they have an index number which can tell them apart. The array can help me from when the user inputs data to the final output (which in my case is the exchange rate). I will use the array to help me store my information in a combo box. For my program I am going to store the information about added currencies in a text file and use stream readers to gather and store information.

I have many ideas for my program, the first idea I had for my program was allow the user to be able to convert twice instead of once. I would do this by making the answer for the first

conversion appear in another combo box so the user can then again make another conversion even with a different currency. I would use a combo box because the user might have more than one first conversion. This would help the local travel agent because it would save them from having to write down and use a calculator when converting currencies.

The second idea I have for my program would be a settings tab so the travel agent could go into the settings tab and change some of the programs features, such as theme and what is shown on the form. This would be very nice for the travel agent since they will be able to customise the program to their preferences and remove unwanted parts that they don't use on the program. I would put all my program controls in a tab box and create a separate tab for the settings; I would then have radio buttons which change what's enabled on the form.

The last idea I have for my program would be a built in notepad. A built in notepad could help the travel agent a lot since they could write down anything, if the travel agent had loads of currency rates and wanted to add them but didn't have enough time since they had to leave for a meeting they could just paste them down in the built in notepad so next time they launch the program all the currency rates will be there ready to be inputted through the program. I would do this by adding a rich text document to my program on a separate tab and add a toolbar so they can save files. I would also had a quick save button so all the user would have to do would be to click the button before they closed the program so next time it opened the information would be there.

.**Justification of the designs**

I have chosen the design because it's simpler for the user and meets all the requirements. The user will be able to convert from British pounds to any currency. They will be able to do this because I will allow them to select which currency they want to convert to in dropdown list, type in the amount in British pounds into a text box and convert them by allowing the user to click a button. The user will also be able to convert into British pounds from a foreign currency. This is because once the user has the foreign amount they can then click a button which will convert it back using the exchange rate in a combo box which for example would be Euro if you're converting to pounds from Euro. The travel agent will be able to add new currency's to the program and update any currencies on the list. This is because I will allow the user to select the currencies they want to change in the combo box and edit them straight from the form. There could possibly be some restrictions on allowing the user to edit the currency straight from the combo box so I might need to have the currency editing on a different tab and make it so that the currencies are edited in a different location rather than from the actual combo box itself.

I mainly rejected my alternative design because it was too complicated for a simple currency calculator also there would be many time constraints when trying to get my code to work. If I decided to implement a text editor into my code then it would take up a lot of time and I would spend most my time making a text editor rather than meeting the requirements.

Test Plan

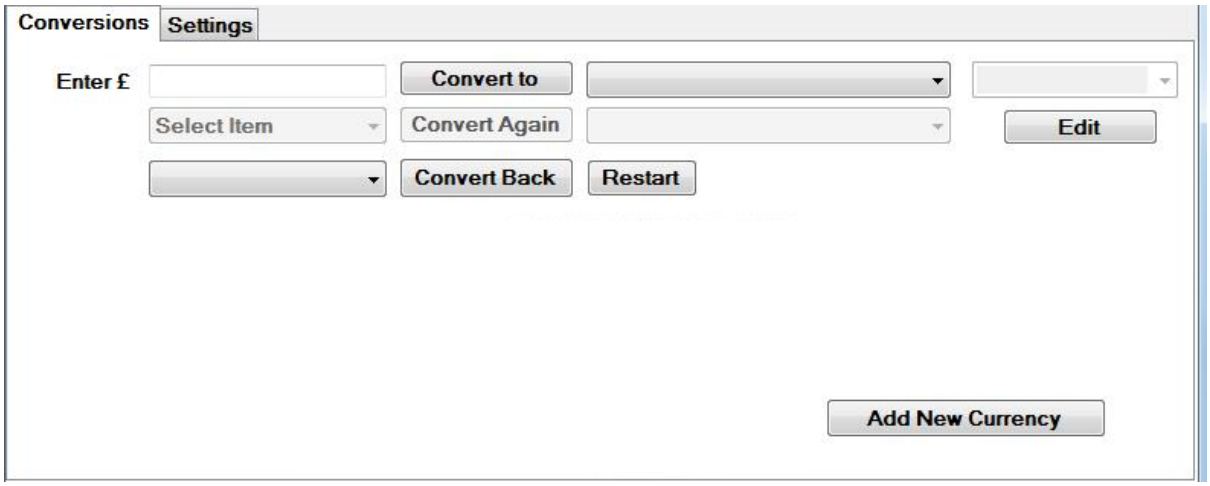
No	Part 1			Part 2		
	Purpose	**Test Data**	**Expected Outcome**	**Actual Outcomes**	**Pass /Fail**	**Comments**
1	Splash Screen with information should appear before application opens	No Test Data	Loading screen should come up before the application is loaded	Loading screen came up correctly because the splash screen was selected to “Form2”	✓	My code was correct, the splash screen was selected to “Form2”.
2	The user clicks on the tabs and they change depending on the tab clicked	No Test Data	The tab should change to the chosen tab with a click	The tabs worked correctly without any coding since it is all form based	✓	Tabs where just added to form
3	The user enters the amount in pounds and selects the exchange rate name, exchange rate is displayed next to the name and the converted currency appears in a label	10	The currency is converted and displayed on in a label	The currency is converted	✓	Make bot combo boxes have the same selected index.
4	Restart button restarts program		Program restarts	The program didn’t restart it just closed	✗	Add an Application. Restart() rather than an End
5	The user clicks the “Add Currency” button and it displays two textboxes, one for the currency	Euro 1.1122	The information should get stored and the application should restart,	The user clicks the button and both text boxes show up successfully, The exit button and save functions also work	✓	I also made the controls for adding currency’s only display when a button is clicked

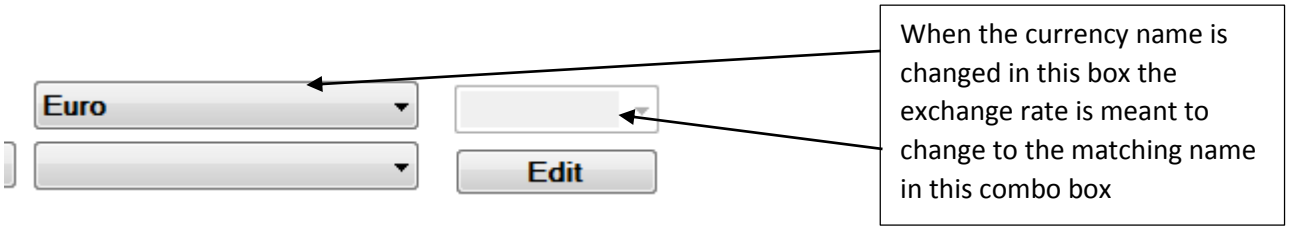
No	Part 1			Part 2		
	Purpose	**Test Data**	**Expected Outcome**	**Actual Outcomes**	**Pass /Fail**	**Comments**
	name and one for					
6	The controls appear for adding currency's and the information gets saved in a text file when "save" button clicked	Chays Currency 123	The new currency should be displayed in the combo box ready for conversions	The new currency is displayed in the combo box all ready for conversions	✓	The application needed a restart for the new information to be saved
7	Make sure that the conversion is correct when converting British pounds to the other currency, example Euro(1.2)	5	The answer of 6 is displayed in a label called lblDisplay	The answer is displayed correctly	✓	
8	Can convert back to British pounds	6 (Euro)	5 (GBP)	5 is displayed which means it was converted back	✓	
9	Are the files created successfully and save in the right location		When adding currency's to the text file, the text should be saved to the text file	The file is created successfully but when editing the currency from the combo box it trys to save on the desktop instead	×	I will need to make reader try to find the file in the same location (as the file)

Refining Quality

I gathered feedback from my friends and found out that the instructions are not clear, as you can see on my form there's nothing telling the user where to go to perform tasks. I've wrote in a label "Enter £" this is not clear enough I should replace it with "Enter Amount in Pounds".

From feedback 1 I got the question "How do I select a currency" because I didn't make it clear enough where to select the currency on my form. To refine the quality I'm going to add labels which give more instructions of what to do on the form.





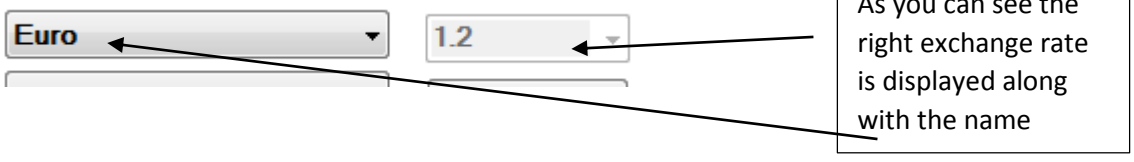
To fix this issue I will have to make it so that the selected indexes are the same as you can see if I manually select the exchange rate it is there but the problem is when I select euro it doesn't show this exchange rate.

To fix this issue I double clicked the first combo box to get this in my coding section

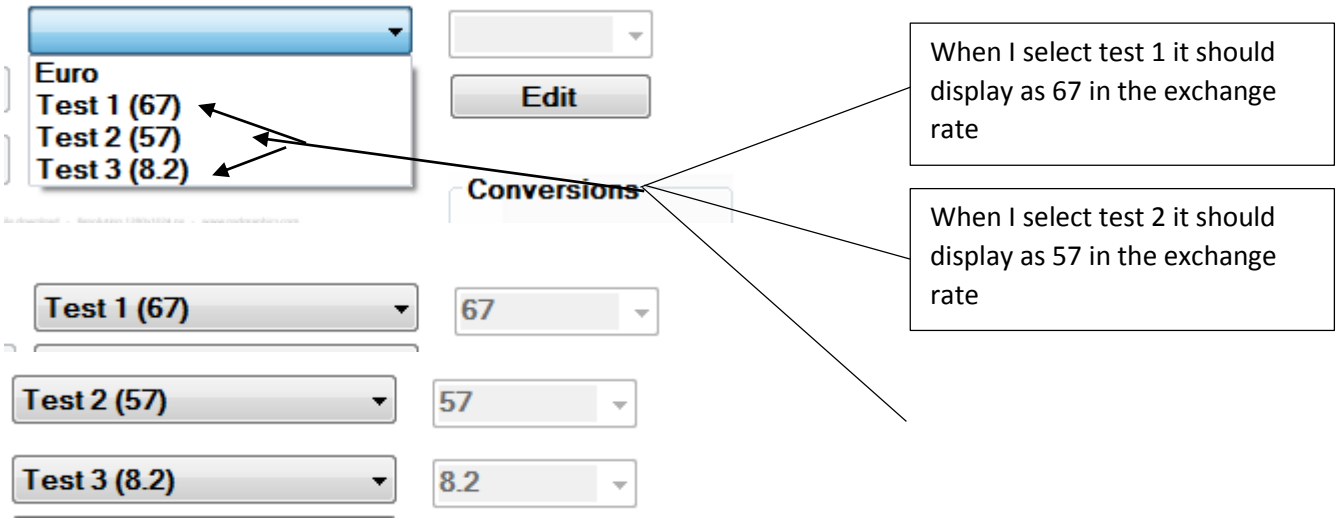
```
    0 references
    Private Sub cnbCurrency_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cnbCurrency.SelectedIndexChanged

    End Sub
End Class
```

Here I'm basically telling the program when the selected index is changed do something, I then added this to my program so when the index of the Currency combo box is changed the exchange rate will also change with it. All the currency names and exchange rates are stored in separate text file but the line numbers are the same. If in the currency name text file "Euros" line number was 5 then the exchange rate would be on line 5 in the exchange rates text file which means when the values are added to the combo boxes the selected indexes will match each currency name and exchange rate automatically. Once I added this code the selected indexes changed together.

```
    End Sub

    0 references
    Private Sub cnbCurrency_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cnbCurrency.SelectedIndexChanged
        cmbExchangeRate.SelectedIndex = cnbCurrency.SelectedIndex
    End Sub
End Class
```

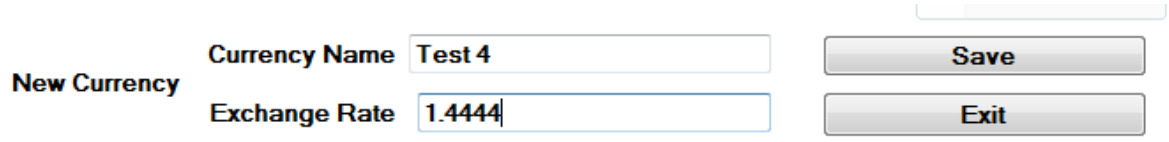


After it displayed the right exchange rate I made 3 tests in my combo box each told me what the number should be in the exchange rate combo box

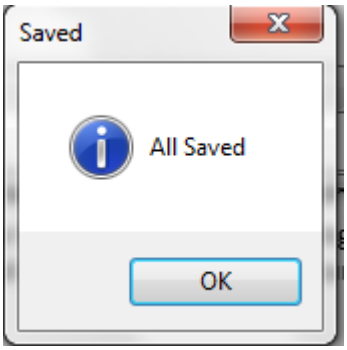


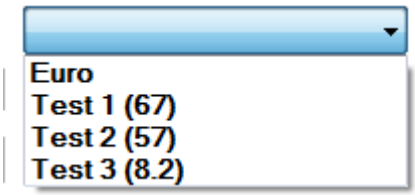
All three tests worked correctly

The need improvement I had to make was the saving of the new currency, when saving the currency it would all write to the text file but it wouldn't show up in the combo box. This I s because the new information hasn't been read from the text file. To fix this I'm going to have to find a solution. I noticed when I exit the application and open it back up the new currency is in the combo box.



I set up a new currency and clicked save and as soon as I looked at the combo box it hasn't been updated






```
'Restarts program so the
Application.Restart()
d Sub
```

I added an application restart to my code so when the application restarts it will display the new currency in the combo box.

The user interface

1. **Are the instructions clear and the interface labelled correctly?**

Feedback 1: the instructions are not clear enough. where do I select the currency? where do I enter the amount in pounds. Not labelled correctly

Feedback 2: Instructions not clear

2. **Can the user enter an amount in pounds and convert to a currency?**

Feedback 1: yes the user can enter the amount in pounds and convert.

Feedback 2: yes

3. **Is the Spelling and grammar correct?**

Feedback 1: The spelling and grammer is correct

Feedback 2: yes

4. **Are the font colours, size and the colour and size of the button appropriate?**

Seedback 1: Yes very appropriate.

Seedback2: yes

5. **Are the buttons and other functions visible and enable appropriately?**

Seedback 1: yes very appropriate, very good work Don on visible controls

Seedback 2: yes

6. **Are the Instructions and Error messages are clear?**

Seedback1: Instructions still not clear. Error messages are clear

seedback 2: Instructions not clear

7. Do you have any more comments?

Feedback 1

very plain user interface

may be need some pictures

Feedback 2

well done

Ascii Code Table

Dec	Char	Dec2	Char2	Dec3	Char3	Dec4	Char4	Dec5	Char5
32	*blank*	51	3	70	F	89	Y	108	l
33	!	52	4	71	G	90	Z	109	m
34	"	53	5	72	H	91	[	110	n
35	#	54	6	73	I	92	\	111	o
36	$	55	7	74	J	93	]	112	p
37	%	56	8	75	K	94	^	113	q
38	&	57	9	76	L	95	_	114	r
39	'	58	:	77	M	96	`	115	s
40	(	59	;	78	N	97	a	116	t
41	)	60	<	79	O	98	b	117	u
42	*	61	=	80	P	99	c	118	v
43	+	62	>	81	Q	100	d	119	w
44	,	63	?	82	R	101	e	120	x
45	-	64	@	83	S	102	f	121	y
46	.	65	A	84	T	103	g	122	z
47	/	66	B	85	U	104	h	123	{
48	0	67	C	86	V	105	i	124	|
49	1	68	D	87	W	106	j	125	}
50	2	69	E	88	X	107	k	126	~