

What does the program need to do? How should the software perform?

- **Convert British Pound Sterling to a foreign currency.**

The program meets this requirement, as it is completely capable of converting British Pound Sterling into four different foreign currencies, including US Dollars and the Euro.

The example below shows how the calculation is done in the code.

```
amount = CDec(txtAmount.Text)
rate = CDec(txtRate.Text)
due = amount * rate
lblAmountDue.Text = due
```

- **Convert foreign currency back into British Pound Sterling.**

The program almost meets this requirement, as it is capable of converting another currency into Pound Sterling, however you have to know and manually enter the exchange rate yourself.

This is because all of the radio buttons used are using exchange rates based off of Pound Sterling, therefore the Pound Sterling radio button has an exchange rate of '1' and the result will be the same as the amount entered. The example below shows the code for the Pound Sterling radio button.

```
rdopound.Checked = True Then
rate = 1
txtRate.Text = rate
```

- **Should not freeze/ crash.**

The program meets this requirement, as it does not crash or freeze at all.

The example below is an extract from the User Feedback form (Question 3) showing that the program did not freeze while being used.

How often did the program crash or freeze (1 – 5)? 1: Non-stop / 5: Rarely/not at all

Answer 1 - 5
Answer 2 - 5

- **Should be easy to understand, and easy to use.**

Based on user feedback, the program is easy to use but could be easier. However, the code behind the program is hard to understand, due to a lack of comments explaining the processes used in it.

The example below shows the answer to Question 4 on the User Feedback

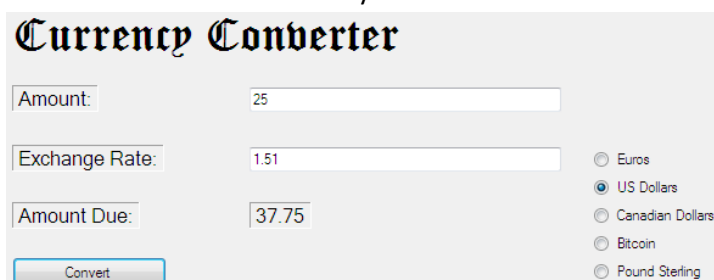
Questionnaire, the examples of code above show the lack of comments...

On a scale of 1-10, how easy was the program to use/understand? 1: Difficult / 10:Very easy

Answer 1 - 9
Answer 2 - 8

- **Display the results.**

The program meets this requirement, as provided the data entered is valid, the program shows the correct result every time.



The screenshot shows a web-based 'Currency Converter' interface. It features a title 'Currency Converter' in a stylized font. Below the title, there are three input fields: 'Amount:' with the value '25', 'Exchange Rate:' with the value '1.51', and 'Amount Due:' with the value '37.75'. To the right of these fields is a vertical list of radio buttons for selecting the target currency: 'Euros', 'US Dollars' (which is selected), 'Canadian Dollars', 'Bitcoin', and 'Pound Sterling'. At the bottom left, there is a 'Convert' button.

Test Table:

Test	Type of Test	Expected Outcome	Actual Outcome	Corrective Measures
Check buttons work.	Functional	Buttons work.	Same	N/A
Attempt to type into amount & rate boxes.	Functional	Can type in boxes.	Same	N/A
Is an output displayed?	Functional	An output is displayed.	Same	N/A
Check that calculations are made correctly.	Logical	Calculations are made correctly.	Same	N/A
Does an error message show when incorrect data is entered?	Logical	An error shows for any incorrect data entered.	Same	N/A
What happens if no data is entered?	Logical	An error shows, counting no data as incorrect data.	Same	N/A

Code Screenshots:

```

Public Class CurrencyConverter
    Dim amount, rate, due As Decimal
    Private Sub CnvrtBtn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CnvrtBtn.Click
        Try
            amount = CDec(txtAmount.Text)
            rate = CDec(txtRate.Text)
            due = amount * rate
            lblAmountDue.Text = due
        Catch
            MsgBox("Invalid data.")
        End Try
    End Sub
End Sub

Private Sub rdoeuro_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles rdoeuro.CheckedChanged
    If rdoeuro.Checked = True Then
        rate = 1.28
        txtRate.Text = rate
    End If
End Sub
  
```