

Postage Calculator

1. The task

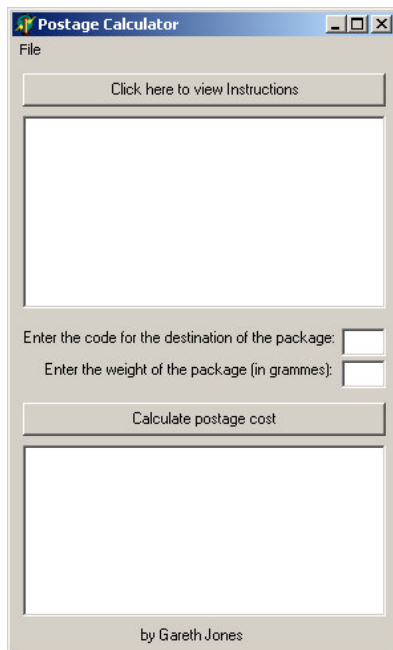
Design and implement a program for employees of a travel agency who send out documents to the UK and EU only; the program will calculate the cost of postage.

Each envelope of documents weighs between 50 and 100 grams. Urgent UK postings will be sent first class (27p up to 60gm, 41p up to 100gm). The non-urgent ones will be sent second class (19p up to 60gm, 33p up to 100gm). Those going to the EU (excluding UK) will be sent at small packet rate (60p up to 60gm, 90p up to 100gm).

Pressing the "Instructions" button should show simple instructions in the memo box, such as a shortened form of the rest of this paragraph. The user of the program will then enter a letter in one edit box to indicate the type of postage service which she/he requires to use (first class, second class or Europe) and the weight of the item in the second box. Pressing the "Calculate postage" button invokes the calculation.

Incorporate checks to ensure the weight entered is neither negative nor over 100 grams.

Implement your system with suitable code using at least one case statement and at least one if ... then statement. The program is terminated when "CLOSE " is pressed.

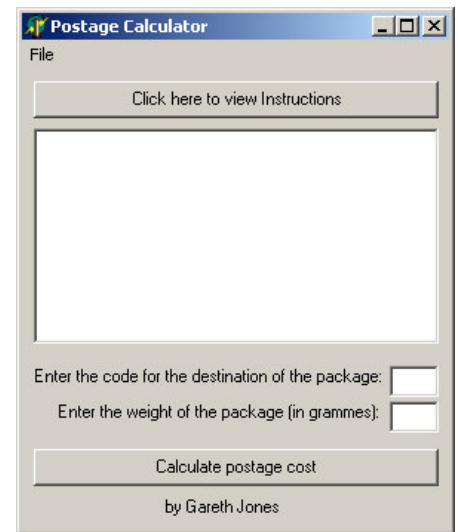


2. Design and Development

Initial design:

The user clicks a button to display the instructions. The instructions direct the user to enter a code into the destination box, either E, F or S depending on the users postage choice. The user then enters the weight of the package in grammes and clicks the "Calculate postage cost" button. The cost is displayed in the memo field beneath. To exit the program the user chooses "Close" from the "File" menu.

This first user interface is shown to the left. The buttons and features of the program are arranged in a chronological order, the order in which the user will want to use them starting at the top.

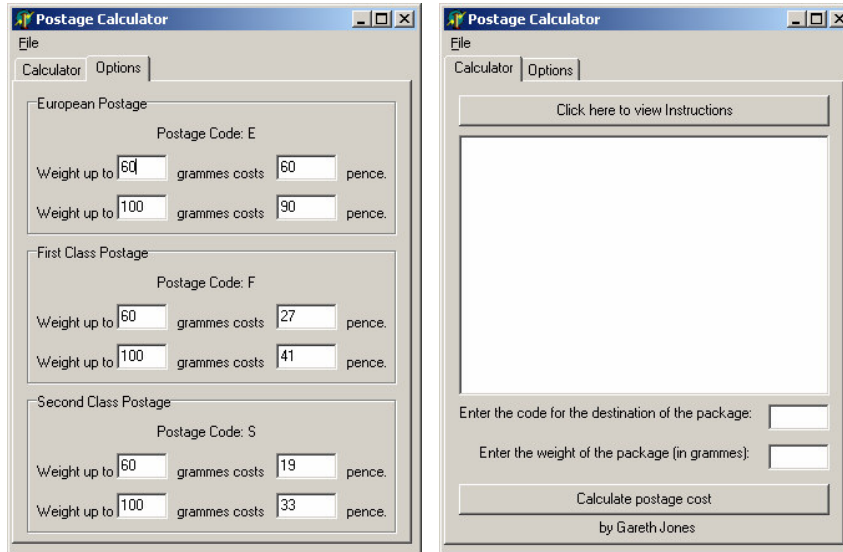


Developing the design:

Using the memo box to display the output from the program is inefficient and not clear. Instead I changed the design and removed the second memo field. The output from the program is now displayed in a popup message. This modified user interface can be seen to the right.

Final design:

In my final design I used tabs and added an "Options" tab. This allows the user to set the limits for the costs and weights of the postage, which may be useful if a different postal service is used or if the costs change. This design can be seen on the next page.



3. Problems

During the process of creating the program I encountered a number of errors.

I had mistakenly defined 'CC' as an Integer, and this gave the error message "Incompatible types 'Integer' and 'TCaption'." I resolved this by declaring 'CC' as a string.

When I was initially using the 'if' function I included a trailing ';' at the end of the command in the 'then' part. This generated the error "';' not allowed before 'else'." After removing this the program compiled.

If the weight entered was a boundary – by default 60 or 100 no message would be displayed. This was because after editing the message for other values, so that the weight remaining was not displayed I did not add another statement to include the boundaries. After adding this additional if statement the problem was resolved.

4. Unit Listing

Here is the Pascal source code copied from Delphi.

```
unit UPostage;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, Menus, StdCtrls, ComCtrls;  
  
type  
  TPostageCalculator = class(TForm)  
    mmuMainMenu: TMainMenu;  
    File1: TMenuItem;  
    Exit1: TMenuItem;  
    ViewInstructions1: TMenuItem;  
    ClearForm1: TMenuItem;  
    lblGareth: TLabel;  
    MainPage: TPageControl;  
    Calculator: TTabSheet;  
    Options: TTabSheet;  
    lblCountryCode: TLabel;  
    lblWeight: TLabel;  
    txtCountryCode: TEdit;  
    txtWeight: TEdit;
```

Gareth Jones (JLB) – Computing (SAH)

```
mmoInstructions: TMemo;
btnInstructions: TButton;
btnCalculate: TButton;
grpEurope: TGroupBox;
lblEC: TLabel;
lblEWL: TLabel;
txtEWL: TEdit;
lblECosts: TLabel;
txtECosts: TEdit;
lblEPence: TLabel;
grpFirst: TGroupBox;
lblFC: TLabel;
lblFWL: TLabel;
lblFCosts: TLabel;
lblFPence: TLabel;
txtFWL: TEdit;
txtFCosts: TEdit;
lblEWL1: TLabel;
txtEWL1: TEdit;
lblECosts1: TLabel;
txtECosts1: TEdit;
lblEPence1: TLabel;
lblFWL1: TLabel;
txtFWL1: TEdit;
lblFCosts1: TLabel;
txtFCosts1: TEdit;
lblFPence1: TLabel;
grpSecond: TGroupBox;
lblSC: TLabel;
lblSWL: TLabel;
lblSCosts: TLabel;
lblSPence: TLabel;
lblSWL1: TLabel;
lblSCosts1: TLabel;
lblSPence1: TLabel;
txtSWL: TEdit;
txtSCosts: TEdit;
txtSWL1: TEdit;
txtSCosts1: TEdit;
ResetOptions1: TMenuItem;
procedure Exit1Click(Sender: TObject);
procedure btnInstructionsClick(Sender: TObject);
procedure btnCalculateClick(Sender: TObject);
procedure ClearForm1Click(Sender: TObject);
procedure ViewInstructions1Click(Sender: TObject);
procedure ResetOptions1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  PostageCalculator: TPostageCalculator;

implementation

{$R *.dfm}

procedure TPostageCalculator.Exit1Click(Sender: TObject);
begin
  //
  // Close the program using this choice from the File menu
  //
  Close;
end;

procedure TPostageCalculator.btnInstructionsClick(Sender: TObject);
begin
  //
  // Clear instruction box and print instructions to the memofield on the Calculator tab
```

Gareth Jones (JLB) – Computing (SAH)

```
//
    mmoInstructions.Clear;
    mmoInstructions.Lines.Add('To use the Postage calculator program follow these steps.');
```

mmoInstructions.Lines.Add('1. Enter the country code in the box below. Enter E for Europe, F for first class or S for second class.');

```
    mmoInstructions.Lines.Add('2. Enter the weight of the parcel in grams in to the box marked weight.');
```

mmoInstructions.Lines.Add('3. Click the calculate button and the postage cost will be displayed');

```
    mmoInstructions.Lines.Add('-----');
    mmoInstructions.Lines.Add('To change the options use the options tab.');
```

mmoInstructions.Lines.Add('Enter the two weight limits and the costs for each postage region.');

```
    mmoInstructions.Lines.Add('The second weight limit must be larger than the second.');
```

mmoInstructions.Lines.Add('The weight must be in grammes, and the cost of postage in pence.');

```
end;
```



```
//
// This procedure is executed when the main calculate postage cost button is clicked
//
procedure TPostageCalculator.btnCalculateClick(Sender: TObject);
var
    CC: String;
    W, EW, ECosts, EWeight, EW1, ECosts1, EWeight1, SW, SW1, FW, FCosts, FWeight, FW1, FCosts1, FWeight1, SCosts,
    SWeight, SCosts1, SWeight1: Integer;
begin
    //Defining variables
    // The Country Code the user enters on the main tab
        CC:= txtCountryCode.Text;
    // The weight the user enters on the main tab
        W:= StrToInt(txtWeight.Text);
    // European first weight limit
        EW:= StrToInt(txtEWL.Text);
    // Cost of first European weight limit
        ECosts:= StrToInt(txtECosts.Text);
    // First weight limit minus weight entered gives additional weight that can be added to package
        EWeight:= EW - W;
    // European second weight limit
        EW1:= StrToInt(txtEWL1.Text);
    // Cost of second European weight limit
        ECosts1:= StrToInt(txtECosts1.Text);
    // Second weight limit minus weight entered gives additional weight that can be added to the package [Second
    limit]
        EWeight1:= EW1 - W;
    // The first class variables follow the trend of the European replacing E with F
        FW:= StrToInt(txtFWL.Text);
        FCosts:= StrToInt(txtFCosts.Text);
        FWeight:= FW - W;
        FW1:= StrToInt(txtFWL1.Text);
        FCosts1:= StrToInt(txtFCosts1.Text);
        FWeight1:= FW1 - W;
    // The second class variables follow the trend of the European replacing E with F
        SW:= StrToInt(txtSWL.Text);
        SCosts:= StrToInt(txtSCosts.Text);
        SWeight:= SW - W;
        SW1:= StrToInt(txtSWL1.Text);
        SCosts1:= StrToInt(txtSCosts1.Text);
        SWeight1:= SW1 - W;
    // Series of IF statements calculating postage cost
    //
    // ** EUROPE POSTAGE**
    //
    // If the first weight is smaller than or equal to the second on the options page an error is generated
    // Following are other conditions which determine the postage cost
    // CC is the Country Code entered by the user on the main tab
    // EW is the first maximum weight value
    // EW1 is the second maximum weight value
    // W is the weight the user enters on the main tab
        if (EW >= EW1)
            then ShowMessage('There is an error on the options page. In the European postage options the
second weight must be larger than the first.')
            else
                if (CC = 'E') and (W >= 0) and (W < EW)
```

Gareth Jones (JLB) – Computing (SAH)

```
        then ShowMessage('To send a package up to ' + IntToStr(EW) + 'gm in weight to Europe will
cost ' + IntToStr(ECosts) + ' pence. You can add another ' + IntToStr(EWeight) + ' grammes to the package
within the current cost.')
    else
        if (CC = 'E') and (W = EW)
            then ShowMessage('To send a package up to ' + IntToStr(EW) + 'gm in weight to Europe will
cost ' + IntToStr(ECosts) + ' pence.')
        else
            if (CC = 'E') and (W > EW) and (W < EW1)
                then ShowMessage('To send a package up to ' + IntToStr(EW1) + 'gm in weight to Europe will
cost ' + IntToStr(ECosts1) + ' pence. You can add another ' + IntToStr(EWeight1) + ' grammes to the package
within the current cost.')
            else
                if (CC = 'E') and (W = EW1)
                    then ShowMessage('To send a package up to ' + IntToStr(EW1) + 'gm in weight to Europe will
cost ' + IntToStr(ECosts1) + ' pence.')
                else
                    if (CC = 'E') and (W > EW1)
                        then ShowMessage('You have entered an invalid weight. Weight must be between 1 and ' +
IntToStr(EW1) + ' grammes.')
                    else
                        if (CC = 'E') and (W < 0)
                            then ShowMessage('You have entered an invalid weight. Weight must be between 1 and ' +
IntToStr(EW1) + ' grammes.')
                        else
//
// ** FIRST CLASS POSTAGE **
//
// If the first weight is smaller than or equal to the second on the options page an error is generated
// Following are other conditions which determine the postage cost
// CC is the Country Code entered by the user on the main tab
// FW is the first maximum weight value
// FW1 is the second maximum weight value
// W is the weight the user enters on the main tab
        if (FW >= FW1)
            then ShowMessage('There is an error on the options page. In the European postage options the
second weight must be larger than the first.')
        else
            if (CC = 'F') and (W >= 0) and (W < FW)
                then ShowMessage('To send a package up to ' + IntToStr(FW) + 'gm in weight to Europe will
cost ' + IntToStr(FCosts) + ' pence. You can add another ' + IntToStr(FWeight) + ' grammes to the package
within the current cost.')
            else
                if (CC = 'F') and (W = FW)
                    then ShowMessage('To send a package up to ' + IntToStr(FW) + 'gm in weight to Europe will
cost ' + IntToStr(FCosts) + ' pence.')
                else
                    if (CC = 'F') and (W > FW) and (W < FW1)
                        then ShowMessage('To send a package up to ' + IntToStr(FW1) + 'gm in weight to Europe will
cost ' + IntToStr(FCosts1) + ' pence. You can add another ' + IntToStr(FWeight1) + ' grammes to the package
within the current cost.')
                    else
                        if (CC = 'F') and (W = FW1)
                            then ShowMessage('To send a package up to ' + IntToStr(FW1) + 'gm in weight to Europe will
cost ' + IntToStr(FCosts1) + ' pence.')
                        else
                            if (CC = 'F') and (W > FW1)
                                then ShowMessage('You have entered an invalid weight. Weight must be between 1 and ' +
IntToStr(FW1) + ' grammes.')
                            else
                                if (CC = 'F') and (W < 0)
                                    then ShowMessage('You have entered an invalid weight. Weight must be between 1 and ' +
IntToStr(FW1) + ' grammes.')
                                else
//
// ** SECOND CLASS POSTAGE **
//
// If the first weight is smaller than or equal to the second on the options page an error is generated
// Following are other conditions which determine the postage cost
// CC is the Country Code entered by the user on the main tab
// SW is the first maximum weight value
```

Gareth Jones (JLB) – Computing (SAH)

```
// SW1 is the second maximum weight value
// W is the weight the user enters on the main tab
    if (SW >= SW1)
        then ShowMessage('There is an error on the options page. In the European postage options the
second weight must be larger than the first.')
    else
        if (CC = 'S') and (W >= 0) and (W < SW)
            then ShowMessage('To send a package up to ' + IntToStr(SW) + 'gm in weight to Europe will
cost ' + IntToStr(SCosts) + ' pence. You can add another ' + IntToStr(SWeight) + ' grammes to the package
within the current cost.')
        else
            if (CC = 'S') and (W = SW)
                then ShowMessage('To send a package up to ' + IntToStr(SW) + 'gm in weight to Europe will
cost ' + IntToStr(SCosts) + ' pence.')
            else
                if (CC = 'S') and (W > SW) and (W < SW1)
                    then ShowMessage('To send a package up to ' + IntToStr(SW1) + 'gm in weight to Europe will
cost ' + IntToStr(SCosts1) + ' pence. You can add another ' + IntToStr(SWeight1) + ' grammes to the package
within the current cost.')
                else
                    if (CC = 'S') and (W = SW1)
                        then ShowMessage('To send a package up to ' + IntToStr(SW1) + 'gm in weight to Europe will
cost ' + IntToStr(SCosts1) + ' pence.')
                    else
                        if (CC = 'S') and (W > SW1)
                            then ShowMessage('You have entered an invalid weight. Weight must be between 1 and ' +
IntToStr(SW1) + ' grammes.')
                        else
                            if (CC = 'S') and (W < 0)
                                then ShowMessage('You have entered an invalid weight. Weight must be between 1 and ' +
IntToStr(SW1) + ' grammes.')
                            else
                                //
                                // If an invalid country code is entered (i.e., not E, F or S) the following message is displayed
                                //
                                ShowMessage('Invalid destination and/or weight entered. Destination must be either E, F or
S.')
                                end;

procedure TPostageCalculator.ClearForm1Click(Sender: TObject);
//
// Clear the two memo fields and two text boxes on the Calculator tab (no change to options tab)
//
begin
    mmoInstructions.Clear;
    txtCountryCode.Clear;
    txtWeight.Clear;
end;

procedure TPostageCalculator.ViewInstructions1Click(Sender: TObject);
begin
//
// Clear instruction box and print instructions to the memofield on instructions tab (no change to options
tab)
//
    mmoInstructions.Clear;
    mmoInstructions.Lines.Add('To use the Postage calculator program follow these steps.');
```

first class or S for second class.');

```
    mmoInstructions.Lines.Add('1. Enter the country code in the box below. Enter E for Europe, F for
    mmoInstructions.Lines.Add('2. Enter the weight of the parcel in grams in to the box marked weight.');
```

mmoInstructions.Lines.Add('3. Click the calculate button and the postage cost will be displayed');

```
    mmoInstructions.Lines.Add('-----');
    mmoInstructions.Lines.Add('To change the options use the options tab.');
```

mmoInstructions.Lines.Add('Enter the two weight limits and the costs for each postage region.');

```
    mmoInstructions.Lines.Add('The second weight limit must be larger than the second.');
```

mmoInstructions.Lines.Add('The weight must be in grammes, and the cost of postage in pence.');

```
end;

procedure TPostageCalculator.ResetOptions1Click(Sender: TObject);
var
EWL, EWL1, ECosts, ECosts1, FWL, FWL1, FCosts, FCosts1, SWL, SWL1, SCosts, SCosts1: String;
```

```

begin
//
// Variables are defined - these are the default values for the fields
//
// Two european weight limits
    EWL:= '60';
    EWL1:= '100';
// Two european costs (relation to weight limit)
    ECosts:= '60';
    ECosts1:= '90';
// First class and second class follow the european style, again replacing E with F or S as appropriate
    FWL:= '60';
    FWL1:= '100';
    FCosts:= '27';
    FCosts1:= '41';
    SWL:= '60';
    SWL1:= '100';
    SCosts:= '19';
    SCosts1:= '33';
//
// When this menu option is selected the default values are assigned to the field on the options tab
//
    txtEWL.Text:= EWL;
    txtEWL1.Text:= EWL1;
    txtECosts.Text:= ECosts;
    txtECosts1.Text:= ECosts1;
    txtFWL.Text:= FWL;
    txtFWL1.Text:= FWL1;
    txtFCosts.Text:= FCosts;
    txtFCosts1.Text:= FCosts1;
    txtSWL.Text:= SWL;
    txtSWL1.Text:= SWL1;
    txtSCosts.Text:= SCosts;
    txtSCosts1.Text:= SCosts1;
end;
end.

```

5. Testing

Test	Why testing?	Expected outcome	Actual outcome
Code E, default limits, values: -23, -10, 0, 28, 33, 59, 60, 62, 88, 90, 99, 100, 106, 2000	Checking correct cost is generated, and that correct weight to be added within current price range is correct, and displayed as appropriate.	Error messages are given for values out of range, and appropriate values are generated for postage costs. Only for values where weight can be added within range is the weight that can be added message displayed.	All as expected except from 60 and 100 where nothing happened when the button was clicked. This was due to me forgetting to enter an if statement.

Having checked this I found that the statement was also missing for the first and second class. I added these, and then began the testing again.

Test	Why testing?	Expected outcome	Actual outcome
Code E, default limits, values: -23, -10, 0, 1, 28, 33, 59, 60, 62, 88, 90, 99, 100, 106, 2000	Checking correct cost is generated, and that correct weight to be added within current price range is correct, and displayed as appropriate.	Error messages are given for values out of range, and appropriate values are generated for postage costs. Only for values where weight can be added within range is the weight that can be added message displayed.	As expected.
Code F, default limits, values: -23, -10, 0, 1, 28, 33, 59, 60, 62, 88, 90, 99, 100, 106, 2000	Checking correct cost is generated, and that correct weight to be added within current price range is correct, and displayed as appropriate.	Error messages are given for values out of range, and appropriate values are generated for postage costs. Only for values where weight	As expected.

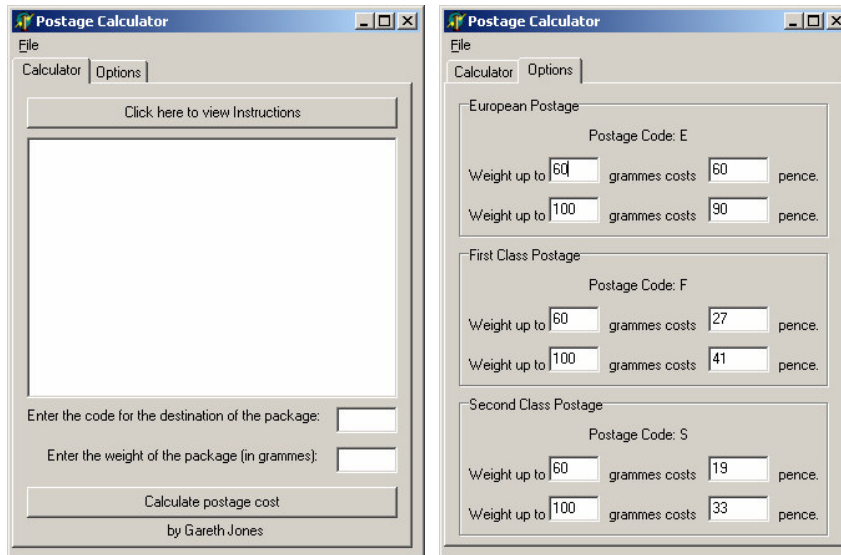
		can be added within range is the weight that can be added message displayed.	
Code S, default limits, values: -23, -10, 0, 1, 28, 33, 59, 60, 62, 88, 90, 99, 100, 106, 2000	Checking correct cost is generated, and that correct weight to be added within current price range is correct, and displayed as appropriate.	Error messages are given for values out of range, and appropriate values are generated for postage costs. Only for values where weight can be added within range is the weight that can be added message displayed.	As expected.
Code E, changing weight limits to 54, 78 and costs to 63, 72. Perform tests for -14, 0, 1, 50, 54, 55, 60, 78, 79, 100	Checking that when the weights and costs are changed on the options page and effect is had.	Weight changes are reflected in the results, and messages demonstrate this.	As expected
Code F, changing weight limits to 54, 78 and costs to 63, 72. Perform tests for -14, 0, 1, 50, 54, 55, 60, 78, 79, 100	Checking that when the weights and costs are changed on the options page and effect is had.	Weight changes are reflected in the results, and messages demonstrate this.	Weight limits defined in the European postage section on the options page are being used.

I had copied the code instead of completely re-writing it and neglected to change these references. Once these references were changed this resolved the problem, and I then continued testing.

Test	Why testing?	Expected outcome	Actual outcome
Code F, changing weight limits to 54, 78 and costs to 63, 72. Perform tests for -14, 0, 1, 50, 54, 55, 60, 78, 79, 100	Checking that when the weights and costs are changed on the options page and effect is had.	Weight changes are reflected in the results, and messages demonstrate this.	As expected.
Code S, changing weight limits to 54, 78 and costs to 63, 72. Perform tests for -14, 0, 1, 50, 54, 55, 60, 78, 79, 100	Checking that when the weights and costs are changed on the options page and effect is had.	Weight changes are reflected in the results, and messages demonstrate this.	As expected.
Code E, weight limits 100,100 prices 80 and 70. Then weight limits 140,120 and prices 79, 134.	The code I have written requires the first weight to be less than the second weight. I have incorporated a validation routine, and am here checking it.	Entering two weight limits the same, or the first weight limit greater than the second should produce an error message.	As expected.
Code F, weight limits 100,100 prices 80 and 70. Then weight limits 140,120 and prices 79, 134.	The code I have written requires the first weight to be less than the second weight. I have incorporated a validation routine, and am here checking it.	Entering two weight limits the same, or the first weight limit greater than the second should produce an error message.	As expected.
Code S, weight limits 100,100 prices 80 and 70. Then weight limits 140,120 and prices 79, 134.	The code I have written requires the first weight to be less than the second weight. I have incorporated a validation routine, and am here checking it.	Entering two weight limits the same, or the first weight limit greater than the second should produce an error message.	As expected.
Enter invalid country code (not E, F or S)	Check that an invalid country code is not accepted	An error message should be displayed informing the user what they have entered is not valid.	As expected.

6. Sample run

Here are two screenshots demonstrating the two tabs that make up the program.



7. Appraisal

I have met the requirements of this task, however not necessarily in the most efficient manner. I have used the basic understanding I currently have of the Delphi environment and the small range of components I understand how to use to achieve this. This task could probably have been more effectively constructed using radio buttons to select options instead of the user entering a weight.

Printing the instructions into a memo field is also not commonly used as the length of the instructions are limited by the size of the box. Further improvements could be made by allowing the postage code to be editable – I chose a single letter to make it easy and quick to enter but either radio buttons or a drop down menu would be a suitable replacement. I could also improve the way the data the user enters is processed, as currently the first weight limit must be less than the second for the code to function.

An option to select urgent or non-urgent posting and use of an options window instead of a tab would be another improvement.

I have used the 'if' function extensively in this program, and dealt with lots of variables. These two aspects make the program's code inefficient but helped me through practise to become more familiar with the operations I have so far learnt.

When troubleshooting I also found appropriate formatting, appropriate identifiers for objects and annotated code to be useful. In this project I named initial versions of a variable by a name, say FC and then a second version of that variable FC1. This is slightly confusing to read as the suffixed '1' implies the variable is the second. Therefore, in future when I know there will be more than one occurrence of a variable I will suffix it with a '1', and the second occurrence with a '2' etc.