Creating an event driven computer program using C#

Level 2

Notes

for

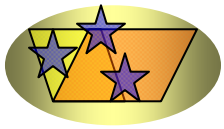
City & Guilds

7540 Unit 008

Written for Microsoft Visual C#® 2005 Express

Tench Computing Ltd
Pines
Glendale Road
Burgess Hill
West Sussex
RH15 0EJ

Web address: www.tenchcomputing.co.uk
Email address: jtench@globalnet.co.uk



About the author: *Jackie Tench MSc, ACIB, Cert Ed*

Jackie started her working career in branch banking with the Midland Bank (now HSBC) and was transferred to their Computing Department after achieving 100% in their ability test for programmers. She then worked for more than a decade in this department and was one of the first women to achieve a junior management grade at the age of 21. She attended a significant number of IBM programming training courses during her time there.

Jackie was the first woman to pass the ACIB (Associate Chartered Institute of Bankers) examinations in the Midland Bank (HSBC) and the youngest person at 21 years of age.

Jackie then left to raise a family but still found time to teach part-time at a college in Sheffield and to obtain a MSc in Computing and a Cert Ed in teaching.

When her children were old enough Jackie returned to work full-time and was a Senior Lecturer in Software Engineering and Computer Studies at a college in Brighton for nearly 10 years teaching all levels up to and including HND.

Therefore, Jackie has considerable business knowledge and qualifications plus wide experience in practical computing and training – covering areas such as structured design, analysis, coding, testing and implementing software applications plus training students to fulfil an important role in the computer industry.

Jackie has worked as a consultant for several blue chip companies and examination boards using her software engineering and educational training skills and is now one of the foremost experts in computing with an extensive knowledge of programming languages and applications.

Contents Page

[This page is intentionally blank]

Chapter 11

Objectives

- *Use the PasswordChar property*
- *Use the Console.WriteLine() method*
- *Locate the EXE file*

PasswordChar property

The PasswordChar property of a text box is used to specify which character is to be displayed when a password is entered. The character is displayed in the text box so that when the password is entered it is not visible on the screen. Instead for every character entered by the user the password character is displayed.

QUOTE program

The Quote program is to provide a quote on the screen. Before the quote can be completed the user has to enter a password.

- Create a new project, save the project as **QUOTE** and the form as **FormQUOTE.cs**.

Build the form from the following table.

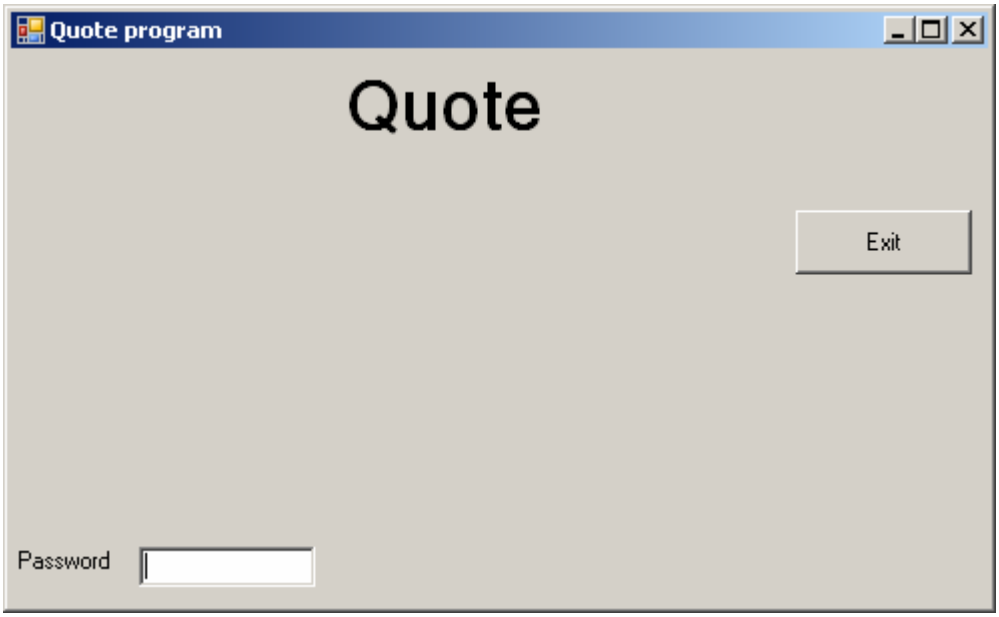
Object	Property	Setting
Form	Name	frmQuote
	Text	Quote program
	Size	496,304
	StartPosition	CenterScreen
Label	Name	lblQuote
	Text	Quote
	Font	26, Bold
	Location	160,8
GroupBox	Name	grpProduct
	Text	Product
	Visible	False
	Size	136,152
	Location	0,56
CheckBox	Name	chkA4Paper
(Draw inside grpProduct)	Text	A4 Blue Paper
	Location	8,24
Checkbox	Name	chkPaperClips
(Draw inside grpProduct)	Text	Paper Clips
	Location	8,56

Object	Property	Setting
Checkbox	Name	chkStaples
(Draw inside grpProduct)	Text	Staples
	Location	8,88
Checkbox	Name	chkCalendar
(Draw inside grpProduct)	Text	Calendar
	Location	8,120
GroupBox	Name	grpPrice
	Text	Price
	Visible	False
	Size	80,152
	Location	136,56
Label	Name	lblA4Paper
(Draw inside grpPrice)	Text	£2.99
	Location	20,29
Label	Name	lblPaperClips
(Draw inside grpPrice)	Text	£1.99
	Location	20,61
Label	Name	lblStaples
(Draw inside grpPrice)	Text	£4.99
	Location	20,93
Label	Name	lblCalendar
(Draw inside grpPrice)	Text	£7.99
	Location	20,125
GroupBox	Name	grpQuantity
	Text	Quantity
	Visible	False
	Size	88,152
	Location	216,56
TextBox	Name	txtQtyA4Paper
(Draw inside grpQuantity)	Size	72,20
	Location	8,24
	TextAlign	Right
	TabIndex	0
TextBox	Name	txtQtyPaperClips
(Draw inside grpQuantity)	Size	72,20
	Location	8,56
	TextAlign	Right
	TabIndex	1

Object	Property	Setting
TextBox	Name	txtQtyStaples
(Draw inside grpQuantity)	Size	72,20
	Location	8,88
	TextAlign	Right
	TabIndex	2
TextBox	Name	txtQtyCalendar
(Draw inside grpQuantity)	Size	72,20
	Location	8,120
	TextAlign	Right
	TabIndex	3
Label	Name	lblPassword
	Text	Password
	Location	0,248
TextBox	Name	txtPassword
	PasswordChar	*
	Size	88,20
	Location	64,248
Button	Name	btnCalculate
	Text	Calculate
	Visible	False
	Size	88,32
	Location	392,48
Button	Name	btnExit
	Text	Exit
	Size	88,32
	Location	392,80
Label	Name	lblSubTotal
	Text	Sub Total
	Visible	False
	Location	320,192
Label	Name	lblTax
	Text	Tax 17.5%
	Visible	False
	Location	320,224
Label	Name	lblTotal
	Text	Total
	Visible	False
	Location	320,256
	Font	Bold

Object	Property	Setting
TextBox	Name	txtCalcSubTotal
	Visible	False
	TextAlign	Right
	Size	90,20
	Location	392,184
TextBox	Name	txtCalcTax
	Visible	False
	TextAlign	Right
	Size	90,20
	Location	392,216
TextBox	Name	txtCalcTotal
	Visible	False
	TextAlign	Right
	Font	10, Bold
	Size	90,22
	Location	392,248

When the program is started the following screen appears.



Most of the controls are not visible until the password **Test** is entered correctly.

- Enter the following code for the **frmQuote_Load()** event function.

```
private void frmQuote_Load()
{
      //Move the focus to the txtPassword TextBox
      txtPassword.Select();
}
```

- Enter the following code for the **txtPassword_TextChanged()** event function.

```
private void txtPassword_TextChanged()
{
      int i,j;
  //Check that the user entered the password "Test" correctly
      if (txtPassword.Text == "Test")
      {
    //Change the background colour of the password text box
          txtPassword.BackColor = Color.Purple;
    //Refresh the text box otherwise the colour does not appear
          txtPassword.Refresh();
    //Do a loop within a loop to count for a delay so that
    //the colour can be seen
          for (i = 1; i <= 20000; i++)
          {
              for (j = 1; j <= 10000; j++);
          }
      //Make the password label and text box not visible
          txtPassword.Visible = false;
          lblPassword.Visible = false;
      //Make the other controls visible
          grpProduct.Visible = true;
          grpQuantity.Visible = true;
          grpPrice.Visible = true;
          btnCalculate.Visible = true;
          btnExit.Visible = true;
          lblSubTotal.Visible = true;
          txtCalcSubTotal.Visible = true;
          lblTax.Visible = true;
          txtCalcTax.Visible = true;
          lblTotal.Visible = true;
          txtCalcTotal.Visible = true;
      }
}
```

- Enter the following code for the **btnExit_Click()** event function.

```
private void btnExit_Click()
{
      Application.Exit();
}
```

- Enter the following code for the **btnCalculate_Click()** event function.

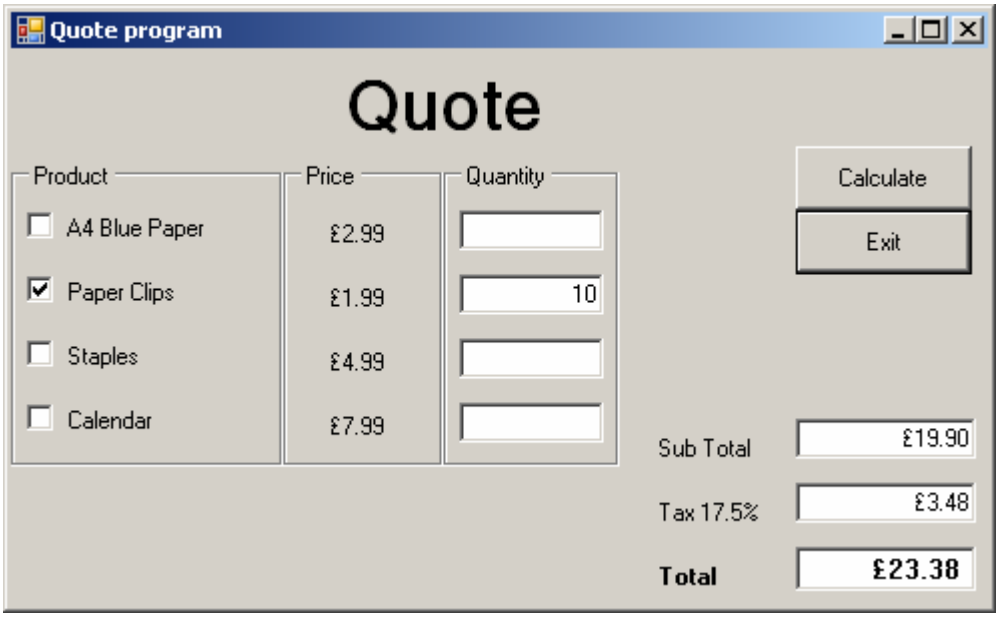
```
private void btnCalculate_Click()
{
   //Declare the prices and Tax rate as constants so
   //that they can be changed easily. float or double is the
   //data type used for numbers that contain decimal places
   const double PRICEA4PAPER = 2.99;
   const double PRICEPAPERCLIPS = 1.99;
   const double PRICESTAPLES = 4.99;
   const double PRICECALENDAR = 7.99;
   const double TAXRATE = 17.5;
   double SubTotal = 0, Tax, Total;
   string StrSubTotal, StrTax, StrTotal;
   if (chkA4Paper.Checked == true && txtQtyA4Paper.Text != "")
   {
      SubTotal = SubTotal +
            Convert.ToDouble(txtQtyA4Paper.Text)
              * PRICEA4PAPER;
   }
   if (chkPaperClips.Checked == true && txtQtyPaperClips.Text
                != "")
   {
      SubTotal = SubTotal +
            Convert.ToDouble(txtQtyPaperClips.Text)
              * PRICEPAPERCLIPS;
   }
   if (chkStaples.Checked == true && txtQtyStaples.Text != "")
   {
      SubTotal = SubTotal +
            Convert.ToDouble(txtQtyStaples.Text)
              * PRICESTAPLES;
      }
   if (chkCalendar.Checked == true && txtQtyCalendar.Text
                != "")
   {
      SubTotal = SubTotal +
            Convert.ToDouble(txtQtyCalendar.Text)
              * PRICECALENDAR;
   }
//Format the SubTotal, Tax and Total values to Fixed Point "F"
//This will allow the value to be displayed as 2 decimal places
      StrSubTotal = SubTotal.ToString("F");
      txtCalcSubTotal.Text = "£" + StrSubTotal;
      Tax = SubTotal * TAXRATE / 100;
      StrTax = Tax.ToString("F");
      txtCalcTax.Text = "£" + StrTax;
      Total = SubTotal + Tax;
      StrTotal = Total.ToString("F");
      txtCalcTotal.Text = "£" + StrTotal + " ";
}
```

A constant is declared with the keyword **const**. The value of a constant cannot be changed while is program is running. Values that may need to be changed at a later date should be declared as constants and not written directly in the code. Constants are normally in uppercase to distinguish them from variables.

A constant declared outside of any function can be seen by all functions. A constant declared inside a function can only be seen by that function.

- Save the project.
- Run the program. Enter the password **Test** and note that the text of the password is hidden and displayed as asterisks. Select a check box and enter a number in the Quantity for that product. When you have selected all the products required click the **Calculate** button.

The following screen shows Paper Clips selected with a Quantity of 10 entered.



String formats

String data which contains numeric data can be formatted so that it appears in a numeric format e.g. with decimal places or with a currency symbol. The following table shows the basic types of formatting that can be used.

Format	**Result**
"D" or "d"	Decimal - integers
"C" or "c"	Currency – currency symbol and 2 decimal places
"F" or "f"	Fixed point – 2 decimal places
"P" or "p"	Percentage – percentage sign inserted at end of string

The code for the formatting of the totals for the QUOTE program could also be done using the Currency format "C" which would mean that the "£" sign would not need to be inserted separately.

The following code shows how the Currency format can be used for the same calculation.

```
//Format the SubTotal, Tax and Total values to Currency "C"
      //This will allow the value to be displayed
      //with a currency symbol and with 2 decimal places
      StrSubTotal = SubTotal.ToString("C");
      txtCalcSubTotal.Text = StrSubTotal;
      Tax = SubTotal * TAXRATE / 100;
      StrTax = Tax.ToString("C");
      txtCalcTax.Text = StrTax;
      Total = SubTotal + Tax;
      StrTotal = Total.ToString("C");
      txtCalcTotal.Text = StrTotal + " ";
```

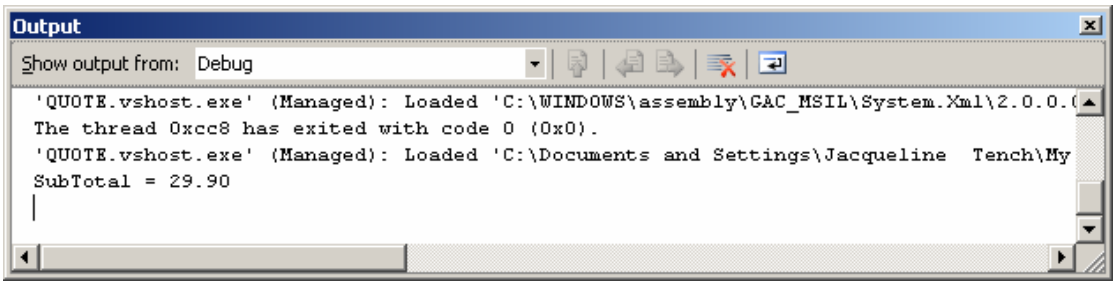
Console.WriteLine() method

The Console.WriteLine method can be inserted in code to output the value in a variable to the Output window. This can help when debugging because you can track the value of a variable as the program is running.

The Output window appears at the bottom of the screen when the program is running in the development environment.

The following line of code will output `SubTotal =` and then the value in the SubTotal variable at this point in the code as shown in the following Output window.

```
Console.WriteLine("SubTotal = {0}", SubTotal.ToString("F"));
```



The output is formatted as Fixed point ("F") which displays with 2 decimal places.

Locate the EXE file

The location of the EXE file depends on whether you have used the default directory to store your projects. The default directory for Visual C# Express Edition 2005 is Visual Studio 2005. The path from this top level directory is then \Projects*projectname*\bin\Debug. The *projectname*.EXE should be stored in this Debug directory.

Note that there is also a Release directory at this level. The Debug directory contains the EXE file which has debug information included. This makes the EXE file a larger size. Normally once a program is completed and tested it is compiled without any debugging information and placed in the Release directory. You need a professional version of Visual C# to be able to create a Release version of the EXE which does not include the Debug information.

The program can be run outside the development environment by locating the EXE file and double clicking on it.

Questions 10

1. Using the QUOTE project:
   - Create test data and calculate the expected results for the program.
   - Test the program and compare the expected results to the actual results.
   - Produce a screen print to show the result for a quote.
   - Add a new button named **btnClear** and set the Text property to **Clear**. When the Clear button is clicked all the checkboxes must be deselected and all the text boxes (except the text box for the password) must be set to empty.
2. Create a new project, save the project as **STUDENT** and the form as **FormSTUDENT**.cs.

   Build the form from the following table.

Object	**Property**	**Setting**
Form	Name	frmStudent
	Text	Student program
	BackColor	light yellow
	Size	384,280
	StartPosition	CenterScreen
Label	Name	lblTitle
	Text	MARKS
	Font	Bold, 14
	Location	144,9
Label	Name	lblName
	Text	Name
	Visible	False
	Location	16,40
Label	Name	lblMaths
	Text	Maths
	Visible	False
	Location	16,80
Label	Name	lblIT
	Text	IT
	Visible	False
	Location	16,120
Label	Name	lblFrench
	Text	French
	Visible	False
	Location	16,160
Label	Name	lblResult
	Text	Result
	Visible	False
	Location	16,200

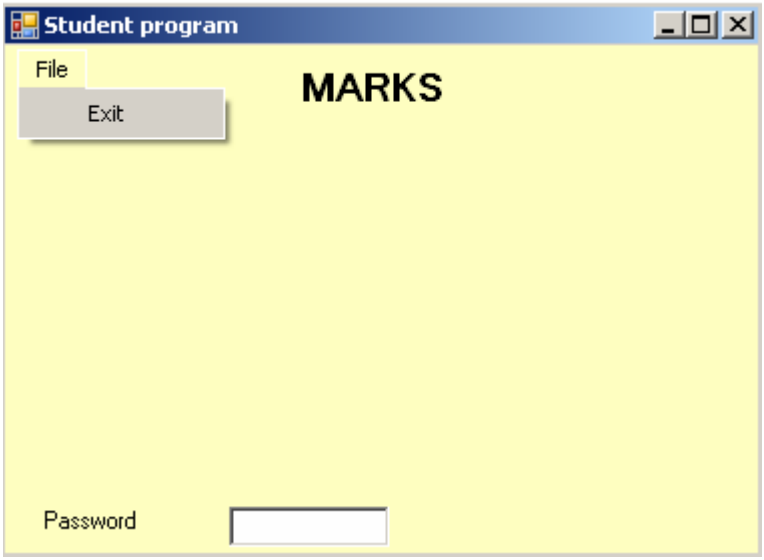
Object	Property	Setting
Label	Name	lblPassword
	Text	Password
	Visible	True
	Location	16,230
TextBox	Name	txtName
	Visible	False
	TabIndex	1
	Size	184,20
	Location	112,40
TextBox	Name	txtMaths
	Visible	False
	TabIndex	2
	TextAlign	Right
	Size	72,20
	Location	112,80
TextBox	Name	txtIT
	TabIndex	3
	Visible	False
	TextAlign	Right
	Size	72,20
	Location	112,120
TextBox	Name	txtFrench
	Visible	False
	TabIndex	4
	TextAlign	Right
	Size	72,20
	Location	112,160
TextBox	Name	txtResult
	Visible	False
	TabIndex	5
	TextAlign	Right
	Size	72,20
	Location	112,200
TextBox	Name	txtPassword
	PasswordChar	*
	TabIndex	0
	Size	80,20
	Location	112,230
	Visible	True
Button	Name	btnCalculate
	Text	Calculate
	Visible	False
	Size	80,40
	Location	272,200

Object	Property	Setting
Button	Name	btnNext
	Text	Next
	Visible	False
	Size	80,40
	Location	272,149

- Add a **MenuStrip** control to the form.
- Create the following menu:

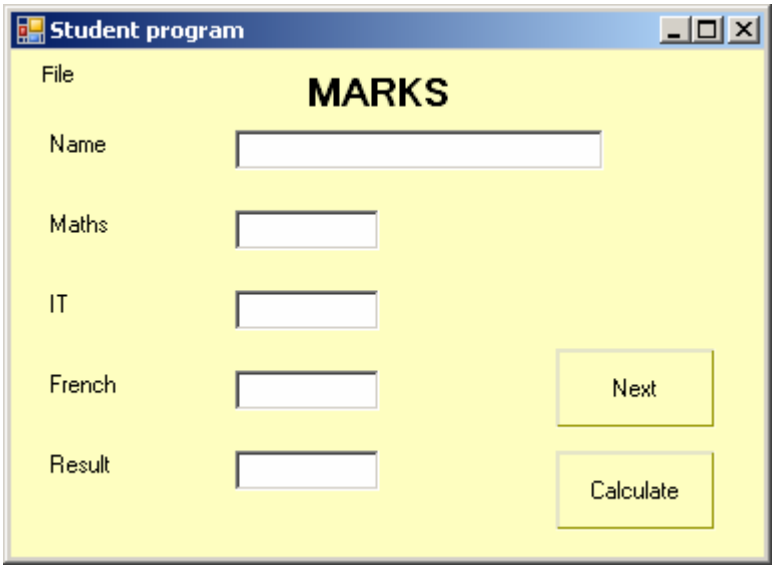
Text	Name
File	mnuFile
Exit	mnuExit

When the program starts running the form should appear as shown below.



- When the correct password is entered the label and text box for the password must be removed and the other controls must be made visible.

The form should then appear as follows.



- Write code for the **mnuExit_Click()** event function to end the program.
- Write code for the **btnCalculate_Click()** event function to work out the average of the marks for Maths, IT and French. A result should be calculated as follows:

  If the average is less than 40 then the result is “Fail”

  If the average is between 40 and 64 (inclusive) then the result is “Pass”

  If the average is between 65 and 84 (inclusive) then the result is “Merit”

  If the average is 85 or more then the result is “Distinction”
- Write code for the **btnNext_Click()** event function to clear the text boxes.
- Add try…catch as appropriate.
- Create test data and calculate the expected results for the program.
- Test the program and compare the expected results to the actual result.

[This page is intentionally blank]