Chapter 1: Visual Studio Express

Learning objectives

By the end of this chapter you will understand:

- the two programming applications used in this book
- how to code and save a basic program in Console Application
- how to obtain input data and provide output in Console Application
- how to code and save a basic program in Windows Forms Application
- how to use the main programming windows in Windows Forms Application
- the format of the event-driven subroutines used in a Windows Forms Application.

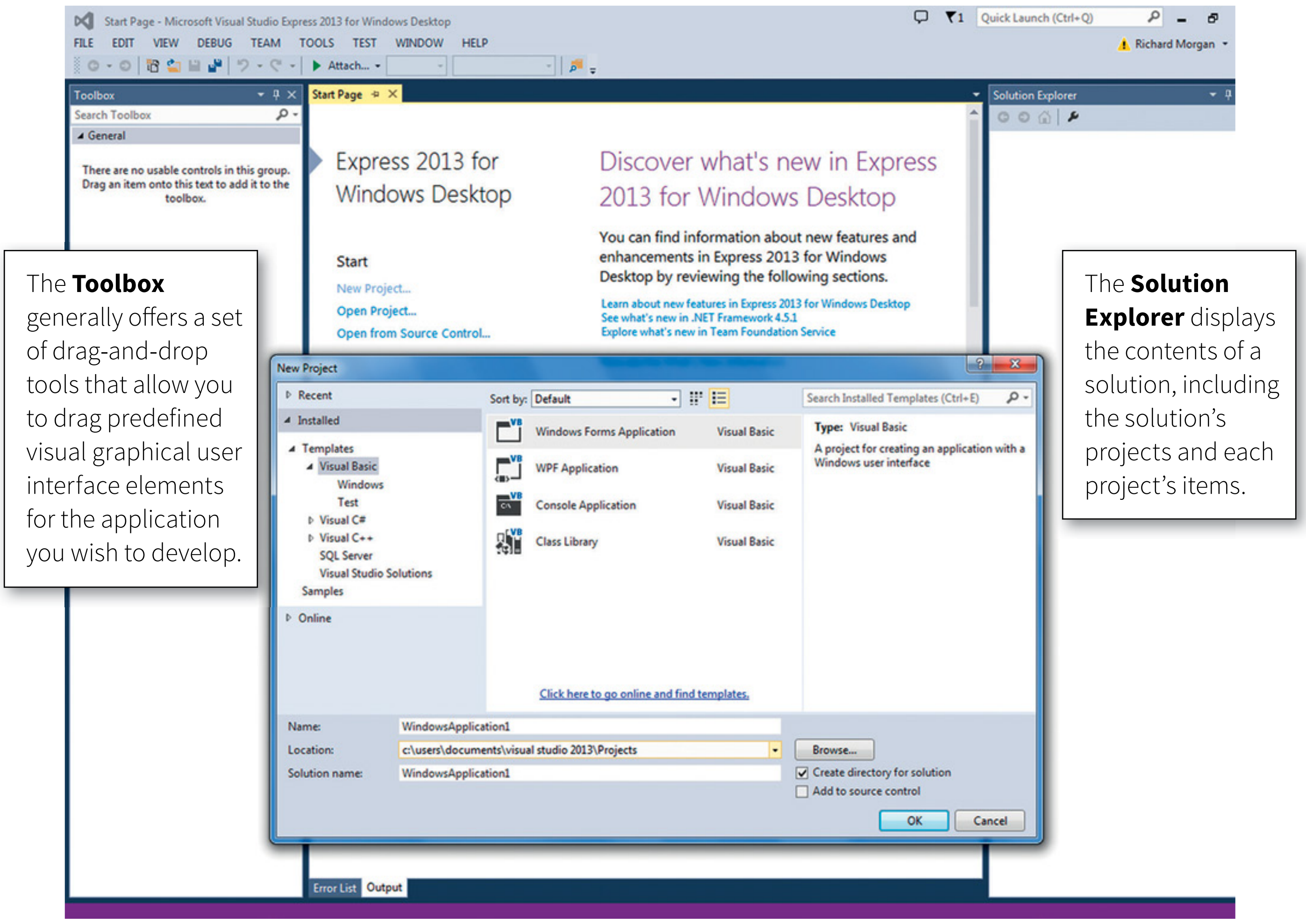
1.01 Getting Visual Studio Express 2013 for Windows

Visual Studio Express 2013 is the current version of free developer tools provided by Microsoft. They include the programming languages Visual Basic, Visual C++ and Visual C#. The examples in this book have been produced using Visual Studio Express for Windows.

I have used Visual Basic with both GCSE and A Level students for the last five years as it provides an interface that allows students to develop programming skills while at the same time producing satisfying systems. Visual Basic also provides programmers with access to a large class library. Classes are templates that hold prewritten code that support functionality of objects. As students' skills increase they are able to use this feature-rich development environment to produce and publish complex systems. System requirements and download options can be found at www.visualstudio.com/products/visual-studio-express-vs.

1.02 The Integrated Development Environment (IDE)

The default start page for Visual Studio Express 2013 is shown in Figure 1.01. It consists of a number of connected windows which offer different functionality based on the type of project that you open. To begin select New Project.


Start Page - Microsoft Visual Studio Express 2013 for Windows Desktop
FILE EDIT VIEW DEBUG TEAM TOOLS TEST WINDOW HELP
Quick Launch (Ctrl+Q)
Richard Morgan
Attach...
Toolbox
Search Toolbox
General
Start Page
Solution Explorer
Express 2013 for Windows Desktop
Start
New Project...
Open Project...
Open from Source Control...
Discover what's new in Express 2013 for Windows Desktop
Learn about new features in Express 2013 for Windows Desktop
See what's new in .NET Framework 4.5.1
Explore what's new in Team Foundation Service
New Project
Recent
Installed
Templates
Visual Basic
Windows
Test
Visual C#
Visual C++
SQL Server
Visual Studio Solutions
Samples
Online
Sort by: Default
Search Installed Templates (Ctrl+E)
Windows Forms Application — Visual Basic
WPF Application — Visual Basic
Console Application — Visual Basic
Class Library — Visual Basic
Click here to go online and find templates.
Browse...
OK
Cancel
Error List
Output

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

You can find information about new features and enhancements in Express 2013 for Windows Desktop by reviewing the following sections.

Type: Visual Basic
A project for creating an application with a Windows user interface

Name: WindowsApplication1
Location: c:\users\documents\visual studio 2013\Projects
Solution name: WindowsApplication1
☑ Create directory for solution
☐ Add to source control

The **Toolbox** generally offers a set of drag-and-drop tools that allow you to drag predefined visual graphical user interface elements for the application you wish to develop.

The **Solution Explorer** displays the contents of a solution, including the solution's projects and each project's items.

Figure 1.01 Visual Studio Express start window

The New Project window provides you with a choice of types of application. The two that are used in this book are Console Applications and Windows Forms Applications. Both make use of very similar coding approaches to algorithms but differ in the way that a user interfaces with them.

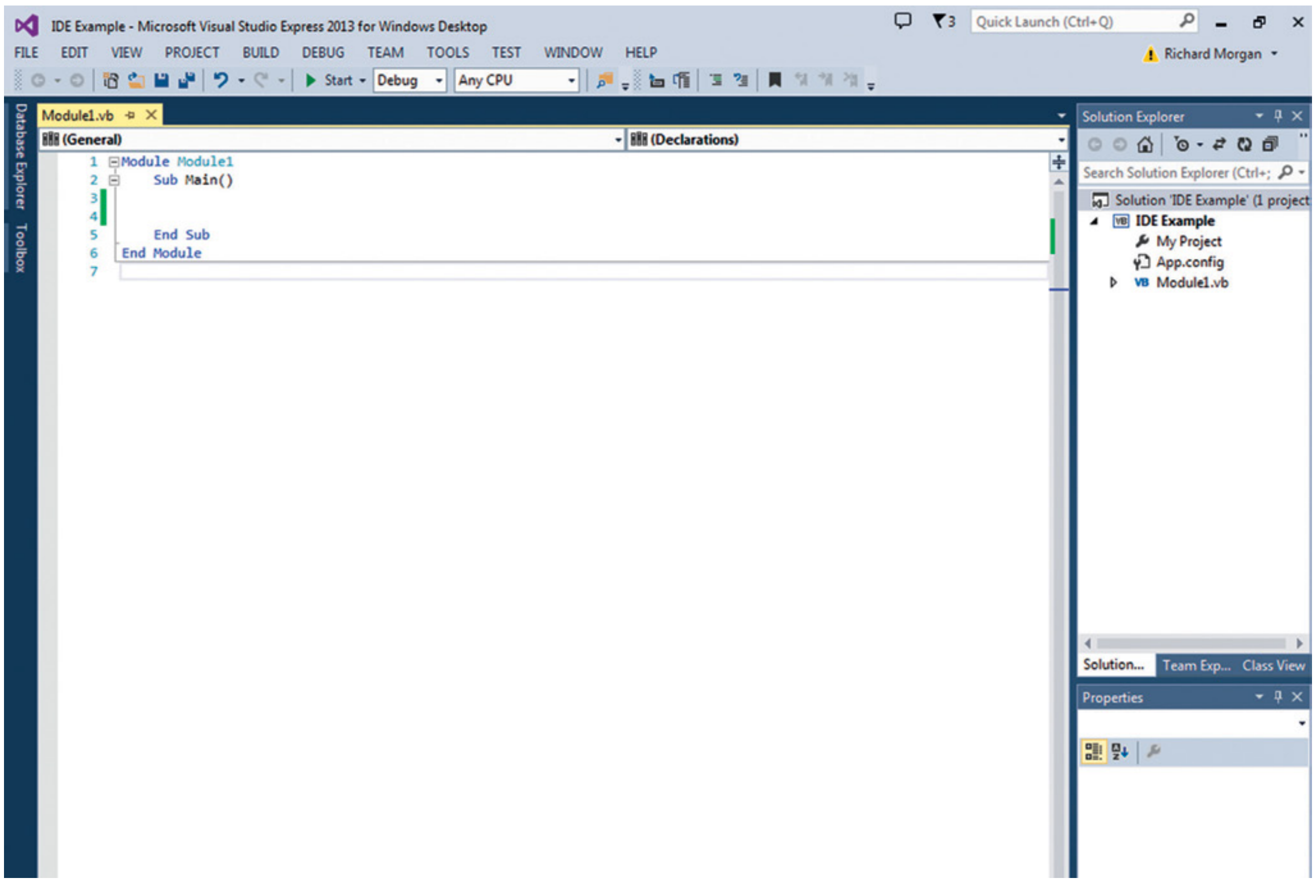
A Console Application provides a textual interface uncluttered by the need to support a graphical user interface (GUI). Although IGCSE does not stipulate the use of any programming language or mode the Console Application is the required format in the A Level syllabus.

A Windows Forms Application provides a graphical user interface that can be customised to provide users with visual input, output and processing options. The flow of the program is largely controlled by routines that are triggered in response to the user interacting with the interface.

To create a new project select the preferred application type (in this case, Console Application), change the default name to something meaningful and select OK.

1.03 Console Application

The default layout of the console mode consists of the main **programming window** (see Figure 1.02) which provides an area in which you write the program code required to accept inputs, process data and produce the required outputs. The **Solution Explorer** displays the contents of a solution, which includes the solution's projects and each project's items. This is where you will find the program you have written, listed as a .vb file.


IDE Example - Microsoft Visual Studio Express 2013 for Windows Desktop
Quick Launch (Ctrl+Q)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM TOOLS TEST WINDOW HELP
Richard Morgan
Start
Debug
Any CPU
Database Explorer
Toolbox
Module1.vb
(General)
(Declarations)
Solution Explorer
Search Solution Explorer (Ctrl+;
Solution 'IDE Example' (1 project
IDE Example
My Project
App.config
Module1.vb
Solution... Team Exp... Class View
Properties

```
Module Module1
    Sub Main()


    End Sub
End Module
```

Figure 1.02 Console application programming window

1.04 Make Your First Program using Console Mode

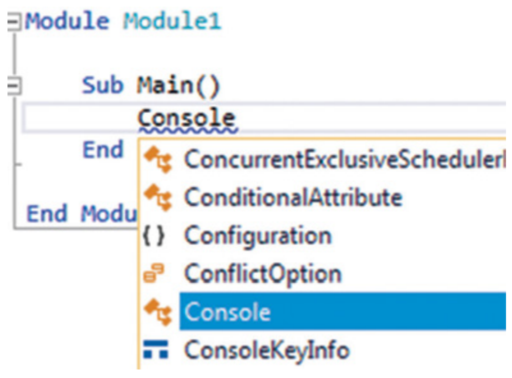
When the console mode is first loaded the code window will contain four lines of code.

```
Module Module1

    Sub Main()
    End Sub

End Module
```

A module is a container of code and can hold a number of subroutines that perform specific actions. `Sub Main()` is the entry point for the program and `End Sub` indicates the end of the subroutine. Code written between these points will be executed when the application is run. You can use the Enter key to add additional lines.

To produce the text 'Hello World' you need to code the application to display the required text.

In Visual Basic the functionality of a class is accessed by use of the dot symbol. Reading inputs and displaying outputs makes use of the Console class which provides access to a library of methods that allow the user to interact with the console.

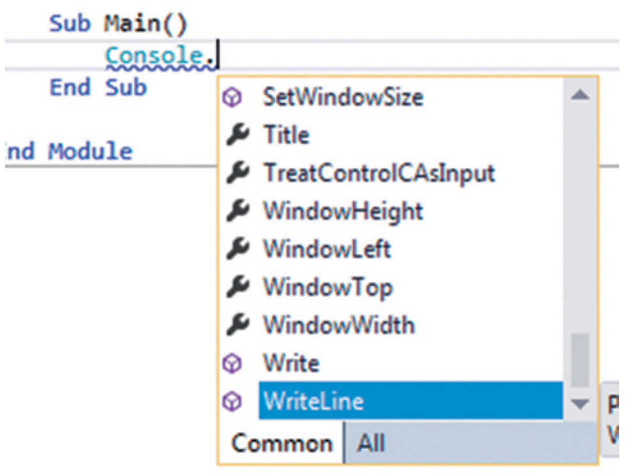


```
Module Module1

    Sub Main()
        Console
    End
End Modu
```

Figure 1.03 Auto-completion window

Type the word 'Console' into the code window (see Figure 1.03). As you type you will notice that the IDE provides an auto-completion window listing all the code inputs or objects that match the letters you have typed. You can double click the correct item, or press Spacebar when the item is highlighted, to auto-complete the entry. This will speed up your coding as once you have typed in the first few characters of the instruction the software will automatically highlight the closest match.

When Console has been completed type a dot symbol. This will show a list of all the available methods for the Console class (see Figure 1.04). The method we need is WriteLine which will display a textual value in the console window when the code is executed. The method has to be passed the required text. To provide the required text the full line will be:



```
    Sub Main()
        Console.
    End Sub

nd Module
```

Figure 1.04 Methods window

```
Console.WriteLine("Hello World")
```

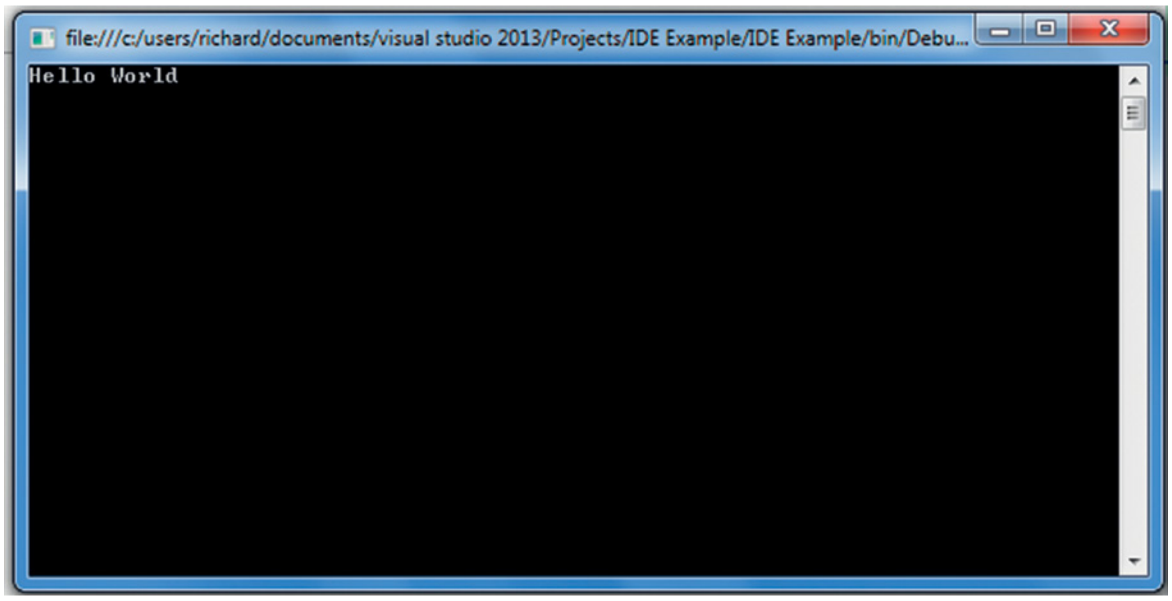
Note that the text to be included is in speech marks to indicate that it is text and not a reference to another object. This code will display the required text in a console window when the application is run but the window will close as soon as the code has been executed. To prevent this, the console ReadKey method is used to pause the execution until a key is pressed on the keyboard. The final code will be:

```
Module Module1
    Sub Main()
        Console.WriteLine("Hello World")
        Console.ReadKey()
    End Sub
End Module
```

TIP

Type **Imports System.Console** before the **Module Module1** line to avoid having to type 'Console' every time you need to use the class.

To run the code click the Start option on the toolbar ▶ Start or use F5 from the keyboard. This will launch the console window (Figure 1.05) and display the text 'Hello World'. The execution of the code will be halted until a key is pressed at which time the window will close.


file:///c:/users/richard/documents/visual studio 2013/Projects/IDE Example/IDE Example/bin/Debu...

```
Hello World
```

Figure 1.05 Console window

Your first console mode program is complete.

Use Save All or the **Save All** option under the **FILE** menu to save your project.

1.05 Windows Forms Application

Console mode makes use of a single user interface to accept text-based inputs and display text-based outputs. Windows Forms provides a visually richer environment which makes use of a range of graphical user interface tools, to produce systems that have more in common with commercial applications.

The interface is more complex as programmers are required to design and produce the graphic user interface that will allow the user to interface with the system. Visual Basic is an event-driven procedural language in which events trigger subroutines that execute the code within them. In this first Windows Forms application clicking a button on the form will trigger an event that delivers the message 'Hello World'.

The default layout (Figure 1.06) contains five main windows.

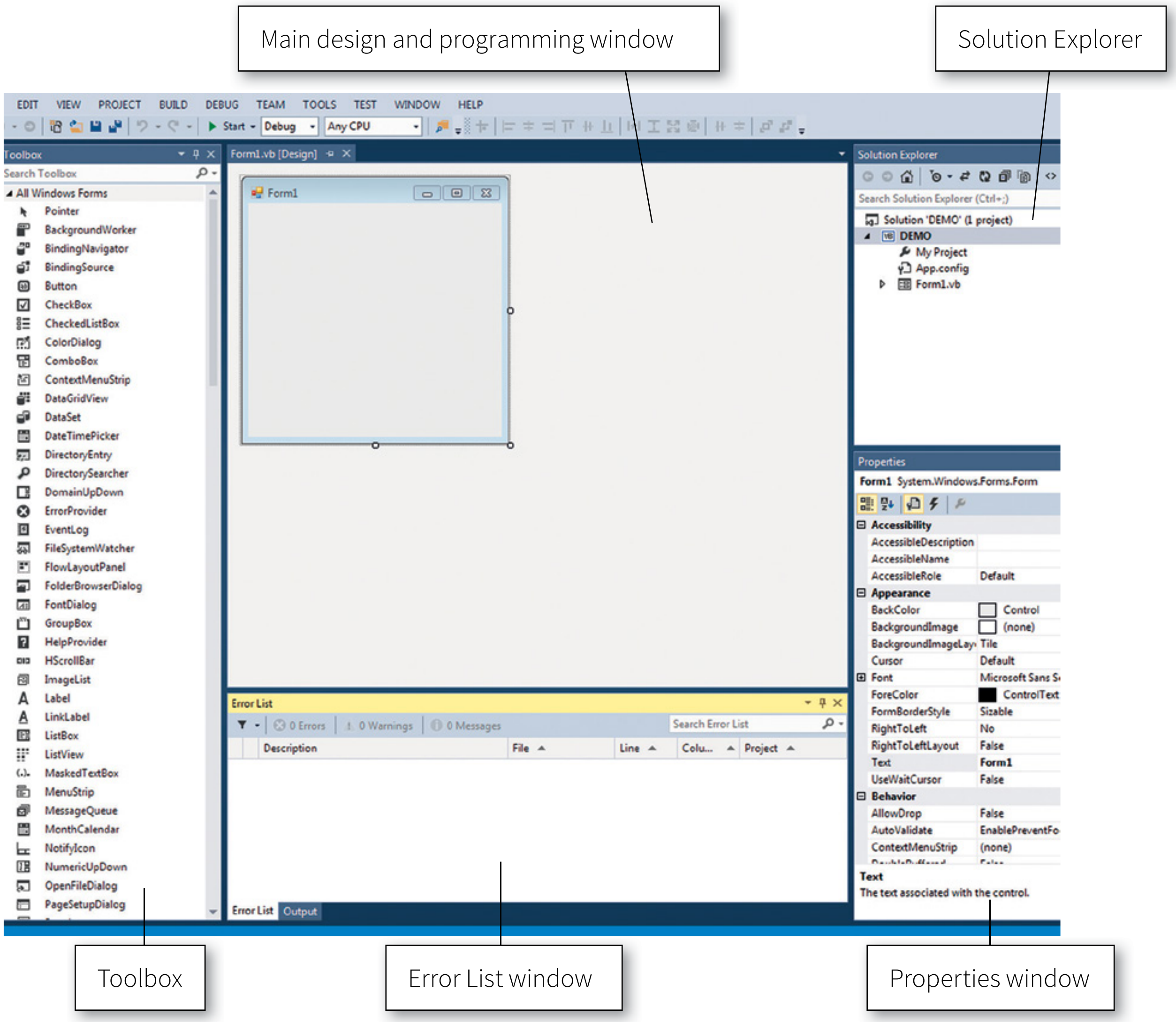


Figure 1.06 Windows Forms application programming interface

Main design and programming window provides an area in which you design your system's interface and write the program code required to accept inputs, process data and produce the required outputs.

Solution Explorer displays the **contents of a solution**, which includes the solution's projects and each project's items. This is where you will find your forms and the program files that support them.

Toolbox provides a set of tools that will allow you to use predefined visual GUI and control objects for the application you wish to develop.

Properties window is used to view and edit configuration-independent, design-time properties and events of selected objects.

Error List window displays any errors, warnings or messages produced as you edit and compile code.

TIP

Individual windows can be docked or set as floating. It is possible to open windows via the **View** menu, and the **Window** menu.

1.06 Make Your First Windows Forms Application

In a Windows Forms Application the traditional 'Hello World' message will be achieved in two steps:

1 Design and construct the user interface

2 Code the program that will generate the required output.

Design the Interface

Find the button object Button in the Toolbox. Click to select the tool and move the mouse over the form in the main design window. The mouse icon will change to show the icon of the selected tool. Click and drag will generate a button on the form. Using the standard Windows mouse controls it is possible to resize and move the button.

Use the same process to generate a textbox object TextBox on the form.

It is considered good practice to give objects meaningful identifiers. An identifier is the name of the object. The default identifier structure is the type of object followed by an increasing number of the objects of that type on the form, such as Button1. The Properties window provides the interface to change the properties of the each object. As you select an object on the form, or the actual body of the form itself the Properties window will change to reflect the properties of the object or form. Objects have many properties that can be configured by the designer but we are initially interested in the properties in Table 1.01.

Table 1.01

Property	What it is
⊟ Design (Name)	The identifier (name) used in the code to identify the object
⊞ Font Microsoft Sans Text Button1	The font style used to display text on or in the object The text that will appear on the object

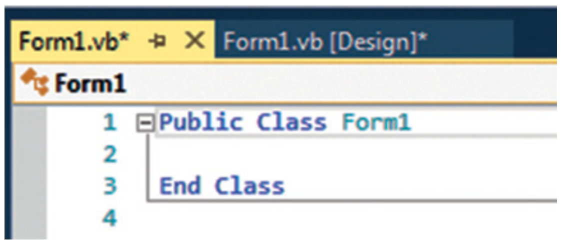
Use properties to give the button and the textbox meaningful identifiers.

Code the Program

To create or edit the program code that will be activated by the form you have designed you will need to open the code window.

To open the code window select <> Code from the **VIEW** menu. The code window will open as an additional tab in the main design window. Clicking the tabs will switch between design and code windows.

It will contain only two lines of code but they are important as they indicate the start and end of the code attached to the form (Figure 1.07). All additional code will be placed between these two indicators. Generally code placed outside will not be part of the form and will cause an error. The Enter key can be used to make additional lines.



```
Public Class Form1

End Class
```

Figure 1.07 Windows Forms application code window

Visual Basic is an event-driven procedural language in which events trigger subroutines that execute the code within them. In this first program clicking the button on the form will trigger an event that delivers the message 'Hello World'. Before writing code the event subroutine has to be created. Creating an event requires you to select the general object and then the specific declaration required. For example, the Button object can have declaration events such as 'click' and 'mouse over'. These different events can be used to trigger different subroutines.

Figure 1.08 shows both of the drop-down menus but it is not possible for you to view both lists simultaneously in the IDE. When you select an object (from the list shown on the left), the list of declarations (shown on the right) shows only the events that are relevant to this object. Click on the Button1 object to select it. The drop-down menu provides a list of all the possible button events. Select 'Click' to insert the Button Click event.

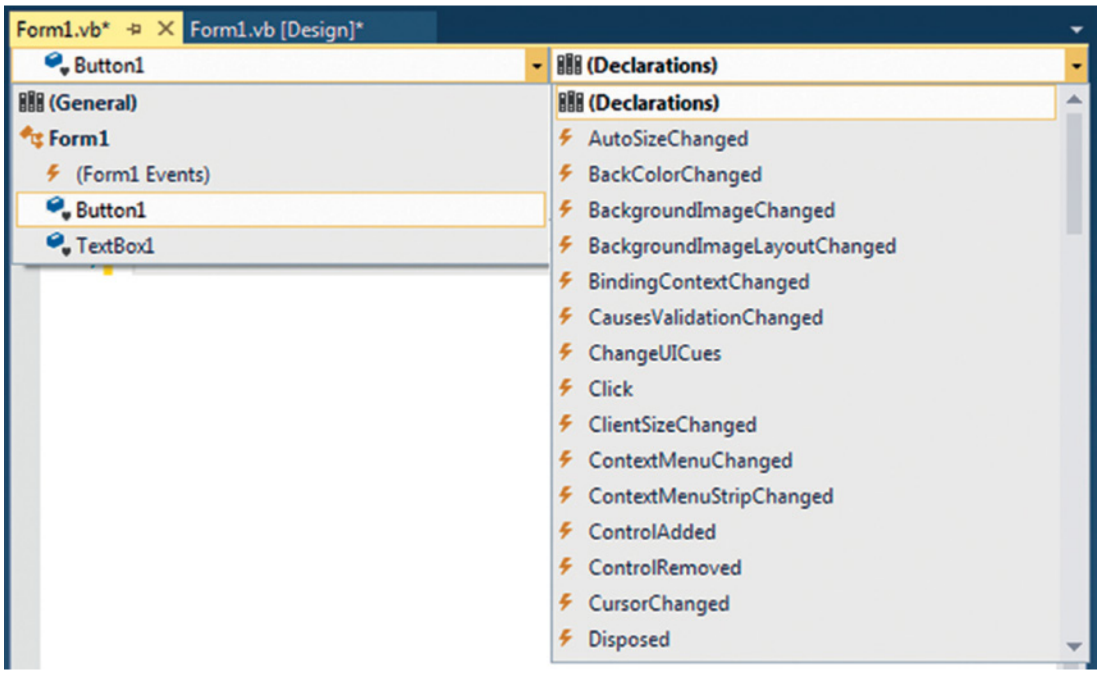


Figure 1.08 Object and Event Declaration selection

TIP

Double clicking an object in the Design window is a shortcut to opening the code window and creating the event subroutine. The software will open the code window and automatically create the default event associated with the object. For example, a Button object's default event is the Click event.

When you select an event the event code will be inserted into your code window.

```
Private Sub Button1 _ Click(sender As Object, e As EventArgs) Handles Button1.Click

End Sub
```

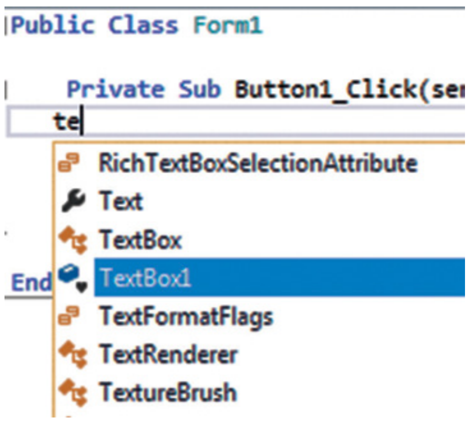
Let us examine the code to identify what each element achieves. Although this makes use of object-oriented language and is outside of the scope of IGCSE, it is useful to have some understanding of how the process works.

Table 1.02

Code element	Description
`Private Sub`	The start of an individual subroutine. Private means that the subroutine is only accessible by this form.
`Button1 _ Click`	The name of the subroutine. The automatic default is to name the routine after the object and event that will trigger the subroutine; however it is possible to rename the subroutine.
`(sender As Object, e As EventArgs)`	The arguments, or data, that are associated with the event. As this is a button click event the arguments are limited – either the button was clicked or it was not. However events associated with mouse activation, for example, will hold data about the location of the mouse on the form and which mouse button was clicked. You should not change or delete any of this data as your subroutine might not work. As you become a more advanced programmer, you will learn how you can manipulate these sections.
`Handles Button1. Click`	The named events that the subroutine can handle. In this case clicking Button1 will call the subroutine and execute the code it contains. It is possible to have a single subroutine triggered by multiple events.
`End Sub`	The end of the subroutine. All the code that is to be executed when the subroutine is called is placed between Sub and End Sub.

In Visual Basic the functionality of a class is accessed by use of the dot symbol. In this example Button1 is an object derived from the Button class and one of the functions available is the Click function. The class library's prewritten code is used when an object of the Button class is clicked. You have no need to write the code – it is contained within the class library and is pretested. As you develop your knowledge of Visual Basic you will find that it makes extensive use of class libraries to provide programmers with functionality.

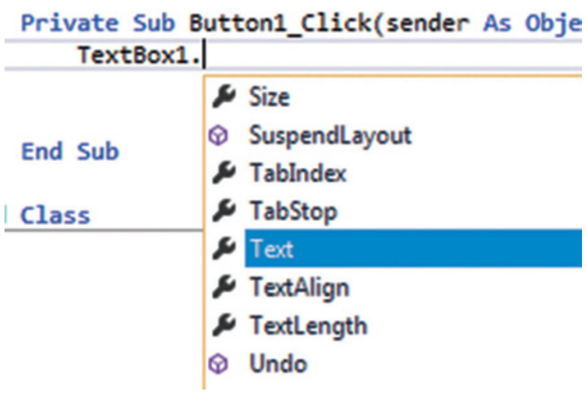
It is now that we can start to write some code. Within the Button Click subroutine, type in the name of your textbox. As you type you will notice that Visual Basic provides an auto-completion window listing all the code inputs or objects that match the letters you have typed (Figure 1.09). You can double click the correct item, or press Spacebar, to auto-complete the entry.



```
Public Class Form1

    Private Sub Button1_Click(se
        te
End
```

Figure 1.09 Auto-complete window

When the name of the textbox has been entered type a dot symbol. As the textbox is an object of the Textbox class this will show a list of all the available methods which can be attached to a textbox object. The method we need is the Text method which will either set text into a textbox or get text from a textbox (Figure 1.10). Double Click, or press Spacebar, to select this method.



```
Private Sub Button1_Click(sender As Obje
    TextBox1.
End Sub
Class
```

Figure 1.10 List of methods

To indicate the actual text that the method will show in the textbox complete the code as follows:

```
TextBox1.Text = "Hello World"
```

Note that the text to be included is in speech marks to indicate that it is new text and not a reference to another object. The final code will look like this:

```
Public Class Form1

    Private Sub Button1 _ Click(sender As Object, e As EventArgs) Handles Button1.Click

        TextBox1.Text = "Hello World"

    End Sub

End Class
```

To run the code click the Start option on the toolbar ▶ Start ▾ or use F5 from the keyboard. This will launch the form as a separate interactive window. Click on the button and the text 'Hello World' will appear in the textbox.

Your first Windows Forms Application program is complete. You have made use of the design window and the Toolbox to create the interface. You have generated a subroutine called by the Click method of a Button object. Within that subroutine you have used the Text method of a Textbox object to place programmer-defined text into the textbox.

Use Save All or the **Save All** option under the **FILE** menu to save your project.

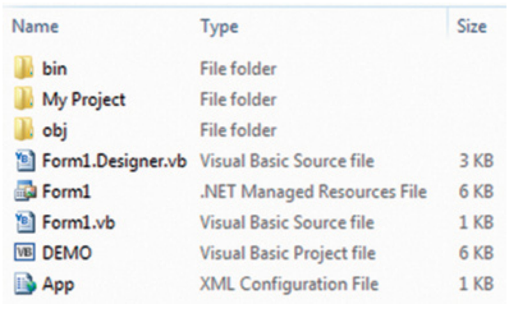
1.07 The Code Behind the Form

As you may have expected the interface objects created by the Toolbox are supported by code that draws the objects on the form. This is generated for you (Figure 1.11) as you build the required interface.

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class Form1
    Inherits System.Windows.Forms.Form

    'Form overrides dispose to clean up the component list.
    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Me.Button1 = New System.Windows.Forms.Button()
        Me.TextBox1 = New System.Windows.Forms.TextBox()
        Me.SuspendLayout()
        '
        'Button1
        '
        Me.Button1.BackColor = System.Drawing.SystemColors.ButtonFace
        Me.Button1.Location = New System.Drawing.Point(37, 43)
        Me.Button1.Name = "Button1"
        Me.Button1.Size = New System.Drawing.Size(79, 42)
        Me.Button1.TabIndex = 0
        Me.Button1.Text = "Button1"
        Me.Button1.UseVisualStyleBackColor = False
```

Figure 1.11 Example of automatic system generated code supporting the GUI

The file that holds this code is stored in the project folder (Figure 1.12). Until you decide to publish your applications you will not need to have a detailed understanding of the role of the project files.



Name	Type	Size
bin	File folder	
My Project	File folder	
obj	File folder	
Form1.Designer.vb	Visual Basic Source file	3 KB
Form1	.NET Managed Resources File	6 KB
Form1.vb	Visual Basic Source file	1 KB
DEMO	Visual Basic Project file	6 KB
App	XML Configuration File	1 KB

Figure 1.12 Solution Explorer showing project files

1.08 Choosing a Console Application or a Windows Forms Application

Throughout this book the various tasks are completed using one, or sometimes both, of these types of application.

Console Applications offer the benefit of more accurately reflecting the programming style of the IGCSE syllabus and will help prepare you for the expectations of the A level syllabus. In the course, you will not be expected to produce algorithms in any specific language; you will use pseudocode and flowcharts to detail answers to questions. Console applications do not involve the additional complexity of having to reference objects from GUI forms.

Windows Forms Applications will offer a richer visual experience and produce systems similar to those commercially available.

I suggest that making use of both applications will best support the development of your computational thinking.

1.09 Additional Support

The intention of this book is to introduce programming concepts making use of the non-language specific formats included in the syllabus. Visual Basic is used to provide the opportunity for you to use a real programming language to develop your understanding of these concepts. Additional support and guidance on the Visual Basic programming language and Visual Studio Express 2013 can be accessed directly from the Microsoft Virtual Academy.

A range of video tutorials and links to other support can be accessed from http://www.microsoftvirtualacademy.com/training-courses/vb-fundamentals-for-absolute-beginners

Summary

Visual Studio Express provides two coding windows:

- Console Applications provide a simple interface and are one of the required language formats used in A Level Computing. The interface is a simple text based console through which user inputs and outputs are handled.
- Windows Forms Applications offer a richer visual interface for the user of your programs. They involve the use of a design window and a coding window. They offer more flexibility over the way in which user inputs and outputs are handled.