Unit 224 Create Software Components using Visual Basic.NET Level 2 (Core)

Rationale

The aim of this unit is to enable candidates to understand the principles required to create software using the Visual Basic.NET programming language. Candidates will develop the skills required to create and test software components or small software systems to solve a given problem.

There are 5 outcomes to this unit. The candidate will be able to:

1. manage the development environment
2. create a Graphical User Interface (GUI)
3. create code for a specified software component
4. use the debug facilities of the development environment
5. test a software component and produce printed output.

Guided learning hours

The recommended guided learning hours for this unit are 90 hours.

Connections with other awards

NVQ links

Developing IT Programs Level 2
216.1 - Assist the creation of software
216.2 - Assemble and test software components

Key Skills links

Communication	C3.2
Application of Number	N1.1
Information technology	None
Working with others	None
Improving own learning	LP3.1, LP3.2, LP3.3
Problem solving	PS3.1, PS3.2, PS3.3

Assessment

Assessment will be by means of a **set assignment covering** practical activities, and a **multiple choice test** covering underpinning knowledge.

Outcome 1: Manage the development environment

	Candidate's signature	**Date**
Practical activities **The candidate will be able to:** 1. use the development environment • help facilities • menus • toolbar • toolbox • windows (code, debug, form, solution explorer, properties) • dialog boxes 2. use **Find** and **Find Next** to locate specified words and strings in a project 3. use **Replace** to replace specified words and strings throughout a project 4. add a control to and remove a control from the toolbox for a project 5. create and save form and project files with meaningful names.		
Underpinning knowledge **The candidate will be able to:** 1. describe the purpose and function of the following file types: • **EXE** • **VB** • **VBPROJ** • **SLN** 2. state that controls can be added to the toolbox for a project from a supplied list or purchased from third-parties		

Outcome 2: Create a Graphical User Interface (GUI)

	Candidate's signature	**Date**
Practical activities **The candidate will be able to:** 1. create a form and controls 2. place a control on a form by drawing, selecting and dragging into position and resizing using control handles 3. change default properties of forms and controls at design time 4. give meaningful names to forms and controls using a consistent naming convention 5. change the settings of the text related properties of controls 6. change the settings of the colour properties of controls 7. change the settings of the functional properties of controls 8. change the settings of the position related properties of controls 9. change the settings of the display related properties of controls 10. use a **GroupBox** control to group and contain other controls 11. select multiple controls on a form to drag the controls as a group or to set a common property for the group 12. use copy and paste to duplicate a control on a form 13. use one event to handle events from multiple controls		

Underpinning knowledge
The candidate will be able to:

1. identify **Form1** as the default project start-up form

2. describe in simple terms the functions of controls

3. state that each type of control possesses a subset of the total number of available properties

4. state that, depending on the property and the control, a property setting may be: changed at design time or run-time; changed only at design time; read at run-time; not available at run-time

5. describe the purpose of objects, methods and properties

6. describe how the events from multiple controls can be handled by one event

7. state that copied controls copy the property values from the original control

8. describe the use of dialog boxes and the main design features of forms used as dialog boxes

9. state the functions of the dialog controls' properties:
   - **FileName**
   - **Filter**
   - **FilterIndex**
   - **OverwritePrompt**
   - **FullOpen**

10. explain the use of controls:
    - **CheckBox**
    - **Button**
    - **ColorDialog**
    - **FontDialog**
    - **GroupBox**
    - **HScrollBar**
    - **Label**
    - **MainMenu**
    - **OpenFileDialog**
    - **PictureBox**
    - **RadioButton**
    - **SaveFileDialog**
    - **TextBox**
    - **Timer**
    - **VScrollBar**

11. state the purpose of the **AcceptButton, CancelButton** and **StartPosition** properties of a form

12. state the purpose of the text related properties of controls:
    - **Font**
    - **MultiLine**
    - **Lines**
    - **Name**
    - **PasswordChar**
    - **Text**
    - **TextAlign**
    - **WordWrap**

13. identify the settings of the colour properties of controls:
    - **BackColor**
    - **ForeColor**

14. identify the settings of the functional properties of controls:
    - **CheckAlign**
    - **Checked**
    - **CheckState**
    - **Enabled**
    - **Interval**
    - **LargeChange**
    - **Locked**
    - **Maximum**
    - **Minimum**
    - **SmallChange**
    - **Value**

15. state the purpose of the position related properties of controls:
    - **Location**
    - **Size**
    - **X**
    - **Y**

16. describe the purpose of display related properties of controls:
    - **AutoSize**
    - **BorderStyle**
    - **Cursor**
    - **DialogResult**
    - **Image**
    - **ImageAlign**
    - **ScrollBars**
    - **SizeMode**
    - **TabIndex**
    - **TabOrder**
    - **TextAlign**
    - **Visible**
    - **WindowState**

Outcome 3: Create code for a specified software component

	Candidate's signature	**Date**
Practical activities **The candidate will be able to:** 1. use comments to document code 2. use consistent indentation and presentation of code to improve readability 3. declare all variables before use 4. declare and use data types eg integer, long, string, Boolean and create new objects 5. declare and use constants and built-in predefined constants as appropriate 6. use operators: assignment operator = relational operators: **=, <, >, <>, <=, >=** arithmetic operators: **+, -, *, /, \, MOD** logical operators: **AND, OR, NOT** 7. create program constructs for iteration: • **For** • **Do While…Loop** • **Do…Loop While** 8. create program constructs for selection: • **If…End If** • **If…Else…End If** • **Select Case…End Select** 9. use the **MsgBox** or **MessageBox.Show** function to display a message to the user and to obtain a return value 10. use the **InputBox** function to obtain an input string from the user 11. use methods for controls 12. read and write control properties during software execution 13. write event-handling code for events for controls		

14. create menus using the **MainMenu** control 15. use dialog controls 16. use a **Graphics** object to draw lines, circles, ellipses and rectangles 17. set and change the **Pen** object **Width** and **Color** properties 18. set and change the **SolidBrush** object **Color** property 19. use methods for a **Graphics** object • **CreateGraphics** • **DrawLine** • **DrawEllipse** • **DrawRectangle** • **FillEllipse** • **FillRectangle** • **Clear** 20. use shortcut keys in menus and controls 21. access a sequential text file for reading writing or appending		

Underpinning knowledge
The candidate will be able to:

1. explain the meaning of the terms ‘data type’ and ‘data type mismatch’

2. state the difference between **Private** and **Public** declarations

3. describe code as being in the form of Sub procedures: either as event procedures associated with controls or general procedures associated with forms

4. describe the functions of the syntax checker and the use of **Option Explicit**

5. describe the action which triggers the following events:
   - **Click**
   - **DoubleClick**
   - **Load**
   - **MouseDown**
   - **MouseMove**
   - **MouseUp**
   - **Scroll**
   - **TextChanged**
   - **Tick**

6. identify ‘idle time’ as the time between event processing and know that idle time is essential for a program to be able to respond to new events

7. state the syntax for: comments; Sub procedures; writing and reading property values

8. describe the logical and relational operators, the precedence rules for arithmetic and the effects of parenthesis

9. describe software start-up sequence with reference to **Form1** and the **Sub Main** procedure

10. state limitations on the use of reserved words

11. describe the operation of iteration program constructs: **For, Do…While, Do…Loop While**

12. describe the operation of selection program constructs: **If…End If, If…Else…End If, Select Case…End Select**

13. describe the structure of a sequential file and the mode of access for reading, writing or appending

14. describe the operation of the **File.Exists** method and **EOF** function

15. explain the importance of closing a file

Outcome 4: Use the debug facilities of the development environment

	Candidate's signature	**Date**
Practical activities **The candidate will be able to:** 1. use debug facilities to locate logical errors 2. break software execution using **Ctrl+Break** 3. set and clear breakpoints in code 4. use single-step mode to trace code execution 5. obtain variable values at breakpoints using available display windows 6. use **Console.WriteLine** to display output in the Output window		
Underpinning knowledge **The candidate will be able to:** 1. describe the operation and purpose of breakpoints in code.		

Outcome 5: Test a software component and produce printed output

	Candidate's signature	**Date**
Practical activities **The candidate will be able to:** 1. produce a diagram or chart for a simple application program to show the relationship between controls and associated event processing and outcomes 2. use test data to determine the expected results from a software component 3. compare the expected to the actual results and correct any errors 4. resolve syntax, logical and run-time errors found during testing. 5. provide evidence that the program complies with the specification 6. create an **EXE** file and test its operation 7. print forms (screen print) 8. print listing of code		
Underpinning knowledge **The candidate will be able to:** 1. describe and distinguish between syntax errors and logical errors 2. identify the cause of a run-time error 3. state the reasons for testing a software component prior to implementation 4. identify that testing for expected output can assist in determining whether or not a program is working correctly and conforms to the specification.		