

Unit 204 Create Software Components using Visual Basic 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 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. use components to 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**

Outcome	This award contributes to the knowledge and understanding of the following elements of NVQ(s)
1,2,3	<i>C&G 4300 - Developing IT Programs NVQ Level 2:</i> 216.1 Assist the creation of software
4,5	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
<p>Practical activities The candidate will be able to:</p> <ol style="list-style-type: none"> 1. access and exit the development environment 2. use the features of the development environment 3. use the Find and Find Next menu commands to locate specified words and strings in a project 4. use the Replace menu command to replace specified words and strings throughout a project 5. add a control to and remove a control from the toolbox for a project 6. create and save form (FRM) and project files (VBP), with meaningful names. 		
<p>Underpinning knowledge The candidate will be able to:</p> <ol style="list-style-type: none"> 1. describe the purpose and function of the following file types: <ul style="list-style-type: none"> • BAS • EXE • FRM • VBP 2. state that controls can be added to the toolbox for a project from a supplied list or purchased from third-parties 3. state the purpose of: <ul style="list-style-type: none"> • Help • Menus • Toolbar • Toolbox • windows (code, debug, form, project, properties) • dialog boxes. 		

Outcome 2: Use components to create a Graphical User Interface (GUI)

	Candidate's signature	Date
<p>Practical activities The candidate will be able to:</p> <ol style="list-style-type: none"> 1. create a form and controls 2. use the Menu Editor 3. place a control on a form by drawing, selecting and dragging into position and resizing using control handles 4. change default properties of forms and controls at design time 5. give meaningful names to forms and controls using a consistent naming convention 6. change the settings of the text related properties of controls 7. change the settings of the colour properties of controls 8. change the settings of the functional properties of controls 9. change the settings of the position related properties of controls 10. change the settings of the display related properties of controls 11. use a Frame control to group and contain other controls 12. select multiple controls on a form to drag the controls as a group or to set a common property for the group 13. use copy and paste to duplicate a control on a form 14. use the alignment grid to place controls accurately. 		

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 control type possesses a sub-set 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. state that copied controls copy the property values from the original control
6. explain why copied controls create an indexed control
7. describe the use of dialog boxes and the main design features of forms used as dialog boxes
8. state the functions of the **CommonDialog** control properties:
 - **CancelError**
 - **FileName**
 - **Filter**
 - **FilterIndex**
 - **Flags**
 - **ShowOpen**
 - **ShowPrinter**
 - **ShowSave**
9. explain the use of controls:
 - **CheckBox**
 - **CommandButton**
 - **CommonDialog**
 - **Frame**
 - **HscrollBar**
 - **Image**
 - **Label**
 - **Line**
 - **OptionButton**
 - **Shape**
 - **TextBox**
 - **Timer**
 - **VscrollBar**
10. explain the purpose of the **Menu Editor**

11. state the purpose of the text related properties of controls:

- **Alignment**
- **Caption**
- **Font**
- **MultiLine**
- **Name**
- **PasswordChar**
- **Text**
- **WordWrap**

12. identify the settings of the colour properties of controls:

- **BackColor**
- **BorderColor**
- **FillColor**
- **ForeColor**

13. identify the settings of the functional properties of controls:

- **Cancel**
- **Checked**
- **Default**
- **Enabled**
- **Interval**
- **LargeChange**
- **Max**
- **Min**
- **SmallChange**
- **Value**

14. state the purpose of the position related properties of controls:

- **Height**
- **Left**
- **Top**
- **Width**
- **X**
- **Y**

15. describe the purpose of display related properties of controls:

- **BackStyle**
- **BorderStyle**
- **BorderWidth**
- **FillStyle**
- **MousePointer**
- **Shape**
- **Transparent**
- **Visible**
- **WindowState.**

Outcome 3: Create code for a specified software component

	Candidate's signature	Date
<p>Practical activities The candidate will be able to:</p> <ol style="list-style-type: none"> 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 integer and string data types 5. declare and use constants and built-in predefined constants as appropriate 6. use operators: assignment operator = relational operators: =, <, >, <>, <=, >= arithmetic operators: +, -, *, / logical operators: AND, OR, NOT 7. create program constructs for iteration: <ul style="list-style-type: none"> • For • Do While 8. create program constructs for selection: <ul style="list-style-type: none"> • If • If...Else • Select Case 9. write functions and procedures 10. use the MsgBox statement and function to display a message to the user and to obtain a return value 11. use the InputBox\$ function to obtain an input string from the user 12. use methods: Move, Print, Refresh 13. read and write control properties during software execution 14. write event-handling code for events for controls 15. use menu controls to call up dialog boxes and position them 		

16. use the CommonDialog control to provide access to the standard set of dialog boxes		
17. use shortcut keys in captions, menus and controls		
18. use the DoEvents function		
19. access a sequential text file.		

Underpinning knowledge

The candidate will be able to:

1. explain the meaning of the terms ‘data type’ and ‘data type mismatch’
2. describe code as being in the form of Sub procedures: either as event procedures associated with controls, general procedures associated with forms, or general procedures in separate code modules
3. describe the functions of the Visual Basic syntax checker and the use of **Option Explicit**
4. describe the action which triggers the following events:
 - **Change**
 - **Click**
 - **DbClick**
 - **Form Load**
 - **MouseDown**
 - **MouseMove**
 - **MouseUp**
 - **Scroll**
 - **Timer**
 - Menu option
5. 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
6. explain the purpose of the **DoEvents** function
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 Visual Basic reserved words
11. describe the operation of iteration program constructs: **For, Do While**
12. describe the operation of selection program constructs: **If, If...Else, Select Case**
13. describe the structure of a sequential file and the method of access:
 - Input
 - Output
 - Append
14. describe the operation of the **FreeFile** and **EOF** functions
15. explain the operation of the **Open** and **Close** statements
16. explain the importance of closing a file.

Outcome 4: Use the debug facilities of the development environment

	Candidate's signature	Date
<p>Practical activities The candidate will be able to:</p> <ol style="list-style-type: none"> 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 the Debug.Print statement to obtain variable values. 		
<p>Underpinning knowledge The candidate will be able to:</p> <ol style="list-style-type: none"> 1. describe the operation and purpose of breakpoints in code. 		

Outcome 5: Test a software component and produce printed output

	Candidate's signature	Date
<p>Practical activities The candidate will be able to:</p> <ol style="list-style-type: none"> 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 images) 8. print listing of code 9. provide evidence that control properties comply with the specification. 		
<p>Underpinning knowledge The candidate will be able to:</p> <ol style="list-style-type: none"> 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. 		