Unit title:	Event driven programming solutions
Unit number:	18
Level:	4
Credit value:	15
Guided learning hours:	60
Unit reference number:	H/601/0453

UNIT AIM AND PURPOSE

This unit will enable learners to gain an insight into how software solutions are created using event driven programming languages. Learners will have the opportunity to design, implement, test and document solutions and develop their programming skills.

LEARNING OUTCOMES AND ASSESSMENT CRITERIA

A pass grade is achieved by meeting **all** the requirements in the assessment criteria.

Learning Outcome (LO) The Learner will:	**Pass** The assessment criteria are the pass requirements for this unit. The Learner can:
LO1 Understand the principles of event driven programming	1.1 discuss the principles, characteristics and features of event driven programming
LO2 Be able to design event driven programming solutions	2.1 design an event driven programming solution to a given problem 2.2 identify the screen components and data and file structures required to implement a given design
LO3 Be able to implement event driven programming solutions	3.1 implement an event driven solution based on a prepared design 3.2 implement event handling using control structures to meet the design algorithms 3.3 identify and implement opportunities for error handling and reporting 3.4 make effective use of an Integrated Development Environment (IDE) including code and screen templates
LO4 Be able to test and document event driven programming solutions	4.1 critically review and test an event driven programming solution 4.2 analyse actual test results against expected results to identify discrepancies 4.3 evaluate independent feedback on a developed event driven programme solution and make recommendations for improvements

	4.4 create onscreen help to assist the users of a computer program 4.5 create documentation for the support and maintenance of a computer program

GRADING CRITERIA

A merit grade is achieved by meeting **all** the requirements in the pass criteria **and** the merit descriptors.

A distinction grade is achieved by meeting **all** the requirements in the pass criteria **and** the merit descriptors **and** the distinction descriptors.

Merit Criteria (M1, M2, M3) (M1, M2, and M3 are mandatory to achieve a merit grade. Each must be achieved at least once per unit to achieve a merit grade.)	**Distinction Criteria (D1, D2, D3)** (D1, D2, and D3 are mandatory to achieve a distinction grade. Each must be achieved at least once per unit to achieve a distinction grade.) (In order to achieve a distinction grade, all merit criteria must also have been achieved.)
MANDATORY TO ACHIEVE A MERIT GRADE	MANDATORY TO ACHIEVE A DISTINCTION GRADE
M1 Analyse concepts, theories or principles to formulate own responses to situations.	D1 Evaluate approaches to develop strategies in response to actual or anticipated situations.
M2 Analyse own knowledge, understanding and skills to define areas for development.	D2 Evaluate and apply strategies to develop own knowledge, understanding and skills.
M3 Exercise autonomy and judgement when implementing established courses of action.	D3 Determine, direct and communicate new courses of action.

TEACHING CONTENT

The Teaching Content describes what has to be taught to cover **all** Learning Outcomes.

Learners must be able to apply relevant examples to their work although these do not have to be the same as the examples specified.

LO1 Understand the principles of event driven programming	
Principles	Event triggering (e.g. user input, timers), event handling, event loops, forms
Characteristics	Service-orientated, ease of development, simplicity of programming, suitability for Graphical User Interfaces, example programming languages and IDEs (e.g. VB .Net using Visual Studio)
Features	Integrated Development Environment (IDE), Graphical User Interface (GUI), GUI objects, object properties, simplicity of programming, high-level code associated with triggered events.
LO2 Be able to design event driven programming solutions	
Specification	User needs problem definition, input/process/output, design methodology (eg Rapid Application Development, waterfall model).
Algorithm design	Triggering event, program flow, dry run using trace tables, algorithm efficiency.
Screen components	Text boxes, labels, command buttons, combo boxes, list boxes, images, check boxes, option buttons, menus, data connection, effective Human Computer Interface (HCI) design.
Data and file structures	File access (eg serial, sequential, indexed sequential, random), database structure (eg tables, records, fields, key field), internal data structures (eg arrays, 2-dimensional arrays, records, stacks, lists).
LO3 Be able to implement event driven programming solutions	
Implement a solution	Use of the IDE, declare and initialise variables and constants, data types and sizes, data storage, procedures, functions, programming standards (use of comments, naming of variables, indentation).
Control structures	Sequence, selection (IF…THEN, SELECT/CASE), iteration (DO…WHILE / DO…UNTIL, FOR…NEXT).

Error handling and reporting	Debugging tools (e.g. breakpoints, variable watch, stepping), error trapping, error types (syntax errors, logic errors, runtime errors), common causes of errors (e.g. overflow, type mismatch, division by zero), translator diagnostics
Use of IDE tools	Source code editor, GUI designer, translation (compiler, interpreter), code templates, screen templates, toolbox items, debugger.
LO4 Be able to test and document event driven programming solutions	
Test strategies	White box testing, black box testing, alpha testing, beta testing, acceptance testing, peer feedback, independent feedback
Test planning	Design of a test plan (e.g. purpose of test, expected result, actual result, action required), test data (normal, erroneous, extreme/borderline/boundary)
Analysis of results	Use of test plan, recording results, comparison of results against expected results, corrective actions, use of feedback to recommend improvements
Documentation and help	Technical documentation, user documentation, onscreen help.

GUIDANCE

Delivery guidance

It will be beneficial to deliver this unit in a way that uses actual events, industry forecasts or sector specific contexts which offer the learner the opportunity to explore, develop and apply the fundamental principles of the sector or subject area. The unit has a strong theoretical content but it is vital that learners are able to apply this in a practical project (or series of projects). This could be done on an individual basis, as part of a team or as a combination of both approaches.

Learners will benefit from being encouraged to exercise autonomy and judgement to design and implement suitable event driven programming solutions, adapt their thinking and reach considered conclusions when testing and analysing solutions based on a foundation of relevant knowledge, understanding and/or practical skills.

Learners would benefit from being presented with subject/sector-relevant problems from a variety of perspectives and from being given the opportunity to explore them using a variety of approaches and schools of thought. For example, it may be possible for centres to establish links with software companies or other businesses in the IT sector that produce event driven solutions, giving an opportunity to explore the approach of experienced programmers. Alternatively, a focus on businesses with a specific problem to be solved would allow learners to explore the needs of users and how these needs can best be met with an event driven solution.

Assessment evidence guidance
Evidence must be produced to show how a learner has met each of the Learning Outcomes. This evidence could take the form of assignments, project portfolios, presentations or, where appropriate, reflective accounts. Where group work/activities contribute to assessment evidence, the individual contribution of each learner must be clearly identified. All evidence must be available for the visiting moderator to review. Where learners are able to use real situations or observations from work placement, care should be taken to ensure that the record of observation accurately reflects the learner's performance. This should be signed, dated, and included in the evidence. It is best practice to record another individual's perspective of how a practical activity was carried out. Centres may wish to use a witness statement as a record of observation. This should be signed and dated and included in the evidence.

RESOURCES

Books

Foxall J., *Teach Yourself Visual Basic 2012 in 24 Hours,* Sams, 2012. ISBN-10: 0672336294 ISBN-13: 978-0672336294

Palmer G., *Java Event Handling,* Prentice Hall, 2001. ISBN-10 : 0130418021, ISBN-13 : 978-0130418029

Petzold C., *Programming Windows: Writing Windows 8 Apps With C# And XAML*, 6th Edition, Microsoft Press, 2013. ISBN-10 : 0735671761 ISBN-13 : 978-0735671768

Open University Course Team, *Event-driven Programming,* Open University, 2009. ISBN-10 : 0749219920, ISBN-13 : 978-0749219925

Campbell S *et al, 101 Visual Basic.NET Applications,* Microsoft Press, 2003. ISBN-10 : 0735618917, ISBN-13 : 978-0735618916

Websites

http://en.wikipedia.org/wiki/Event-driven programming
http://eventdrivenpgm.sourceforge.net/
www.homeandlearn.co.uk/net/vbNet.html
http://en.wikibooks.org/wiki/Java_Programming/Event_Handling