

Unit 14: Event Driven Programming

Unit code:	F/601/7281
QCF Level 3:	BTEC National
Credit value:	10
Guided learning hours:	60

● Aim and purpose

This unit aims to enable learners to develop the skills and understanding required to design and develop event driven applications.

● Unit introduction

Event driven programming is a very flexible way of allowing programs to respond to many inputs or events. Unlike traditional programming, where the control flow is determined by the program structure, the control flow of event driven programs is largely driven by external events. Typically, event loops are pre-programmed to continually look for information to process.

This unit allows learners to become familiar with the underpinning concepts of event driven programming and subsequently to develop particular skills in an event driven language. The unit starts by looking at the features of event driven programming, explores the tools and techniques used in their development and takes learners through design and program development. Learners will use a structured approach to the design and development of applications, ensuring the solution is well documented and tested thoroughly against the original user requirement.

Event handling features in many languages including Visual Basic, Visual Basic for Applications and many other systems.

● Learning outcomes

On completion of this unit a learner should:

- 1 Understand the features of event driven programming
- 2 Be able to use the tools and techniques of an event driven language
- 3 Be able to design event driven applications
- 4 Be able to implement event driven applications.

Unit content

1 Understand the features of event driven programming

Key features: service oriented; time driven; event handlers; trigger functions; events eg mouse, keyboard, HTML object, form, user interface; event loops; flexibility; suitability for graphical interfaces; simplicity of programming; ease of development

Examples: operating systems as event driven systems; Graphical User Interfaces (GUIs)

Programming languages: eg Visual Basic (VB), Visual Basic for Applications (VBA), Coldfusion; Integrated Development Environments (IDEs), C#

2 Be able to use the tools and techniques of an event driven language

Triggers: eg key press, alarm, system event, touch screen event, mouse click

Tools and techniques: eg use of tool boxes and controls, selection, loops, event handlers, triggers, objects and object properties, menus; debugging tools

Variables: declaring variables; scope of variables; constants; data types

3 Be able to design event driven applications

Specification: input; output; processes; user need; purpose

Design: selecting and assigning properties to screen components; data storage; event procedures and descriptions; appropriate ways of representing the processing tasks

4 Be able to implement event driven applications

Creation of application: use of development environment; debugging; data validation; error handling and reporting

Programming language syntax: eg selecting, declaring and initialising variable and data structure types and sizes

Constructs: selection eg if ... then ... else, CASE; iteration eg while ... do, repeat ... until

Programming standards: eg use of comments; code layout; indentation

Testing: test strategy; test plan structure eg test, date, expected result, actual result, corrective action; error messages; specialist software tools eg debug

Review: against specifications requirements; interim reviews

Documentation: user; technical

Assessment and grading criteria

In order to pass this unit, the evidence that the learner presents for assessment needs to demonstrate that they can meet all the learning outcomes for the unit. The assessment criteria for a pass grade describe the level of achievement required to pass this unit.

Assessment and grading criteria		
To achieve a pass grade the evidence must show that the learner is able to:	To achieve a merit grade the evidence must show that, in addition to the pass criteria, the learner is able to:	To achieve a distinction grade the evidence must show that, in addition to the pass and merit criteria, the learner is able to:
P1 explain the key features of event driven programs	M1 discuss how an operating system can be viewed as an event driven application	D1 evaluate the suitability of event driven programs for non-graphical applications
P2 demonstrate the use of event driven tools and techniques	M2 give reasons for the tools and techniques used in the production of an event driven application [IE2, IE6]	
P3 design an event driven application to meet defined requirements [CT1]		
P4 implement a working event driven application to meet defined requirements [SM2, SM3]		
P5 test an event driven application [SM4]	M3 analyse actual test results against expected results to identify discrepancies [RL3]	D2 evaluate an event driven application. [IE4]
P6 create onscreen help to assist the users of a computer program. [RL6]	M4 create technical documentation for the support and maintenance of a computer program. [RL6]	

PLTS: This summary references where applicable, in the square brackets, the elements of the personal, learning and thinking skills applicable in the pass criteria. It identifies opportunities for learners to demonstrate effective application of the referenced elements of the skills.

Key	IE – independent enquirers	RL – reflective learners	SM – self-managers
	CT – creative thinkers	TW – team workers	EP – effective participators

Essential guidance for tutors

Delivery

Learners must have access to facilities that give them the opportunity to evidence all of the assessment criteria fully. If this cannot be guaranteed centres should not attempt to deliver this unit.

It is likely that learners will need plenty of opportunities to develop their skills in the particular language chosen and in the event handling aspects of that language. It is advised, therefore, that several small, well-defined exercises are used to build competence. Good use should be made of formative assessments to build confidence and ensure that learners are following good practice.

It is important that learners accept that solutions, however small, must be fit for purpose. This requires focusing on capturing requirements, defining a specification, designing a solution and then testing and reviewing it. Generic and underpinning skills will be of value later on, no matter what language is actually used.

The suggested delivery pattern follows the order of the learning outcomes in the unit specification. This is not the only sequence that may be used and tutors can follow their own preference.

This unit is not designed with any specific programming language or delivery platform in mind; centres may focus on one or more languages for teaching. Learners will develop an application that must be event driven and may work on a range of platforms, therefore it may be command line, web based, graphical user-interface based, games-console based or a deliverable for a mobile platform among other solutions.

Tutors are advised to keep delivery to one language, although many languages now allow development in multiple platforms.

Learning outcome 1 covers all principles associated with the selected programming language. It is advised to deliver all of the outcome to cover all programming concepts whilst teaching the concepts of event driven systems in parallel.

The design in learning outcome 2, may use a range of design methodologies, ensuring that the selected method is suited to the environment selected, as well as the chosen programming language of choice.

Implementation in learning outcome 3 must be based on a suitably structured problem that ensures use of more than two trigger types, and some simple event driven code, both between events and as a result of events.

Testing in learning outcome 4 must cover the code created in learning outcome 3 and designed in learning outcome 2. Software testing can be used to enhance (not replace) this learning outcome and give learners an extended software development experience.

Whilst this is, ideally, an introductory unit, developing learners' understanding of programming, in selecting the programming design and implementation for learning outcomes 2, 3 and 4 learners could be encouraged to devise their own mini-project to develop their higher learning and project management skills in preparation for the work environment and the HND project unit.

A centre may select a programming activity, or use an external source (employer, commissioner, open source), the design of the programming solution does not need to be a stand-alone application and may be an enhancement of, or extension to, existing work. Therefore learners completing this unit may contribute to many open source development projects or use these as a basis for their learning experience.

Outline learning plan

The outline learning plan has been included in this unit as guidance and can be used in conjunction with the programme of suggested assignments. The outline learning plan demonstrates one way in planning the delivery and assessment of this unit.

Topic and suggested assignments/activities and/assessment
Introduction to the unit
Features of event driven programming: <ul style="list-style-type: none">• whole-class exercise – tutor presentation on the key features of event driven languages and their role in programming• whole-class exercise – learners work from tutor-provided materials to create basic data and control structures• individual exercise – working from tutor-provided materials, learners practise the basics of event driven programming• whole-class exercise – tutor presentation on why programming standards are needed and how to implement them.
Assignment 1 – How to Start
Tools and techniques: <ul style="list-style-type: none">• whole-class exercise – tutor presentation on program design• directed research – working from tutor-specified sources, research triggers and their role in event driven programming• whole-class exercise – tutor presentation on the integration of event driven code into event driven programming, between triggers and in modules activated by triggers• individual exercise – use tutor-provided exercise materials to better understand simple event driven code• individual exercise – use tutor-provided exercise materials to better understand the scope of variables• whole-class exercise – tutor presentation on the integration of event driven code into events.
Assignment 2 – Design Work
Design and implement: <ul style="list-style-type: none">• whole-class exercise – tutor presentation how to combine modular and control elements into a coherent structure• individual exercise – using tutor-provided materials, understand how to use simple type variables, data structures and storage• directed research – using tutor-specified sources, research programming standards• whole-class exercise – tutor presentation on mechanisms for event driven coding• individual exercise – prepare a test plan• individual exercise – test and evaluate the program.
Assignment 3 – Implement and Test
Documentation: <ul style="list-style-type: none">• whole-class exercise – tutor presentation on documentation• individual exercise – following a tutor presentation, prepare different types of support documentation.

Assessment

It is suggested that this unit is assessed using four assignments as summarised in the *Programme of suggested assignments* table.

Finding a scenario which covers all aspects of all criteria is difficult, but the one suggested is acceptable. It places the user in a role which is at an acceptable level for their experience, which is important when devising assignments.

Some of the evidence required to complete the assignments could be naturally occurring within their work for other units within the qualification, or for other courses they are undertaking, and tutors are encouraged to use such evidence.

Evidence produced for this assignment can be used towards the evidence required for other criteria in this unit.

In order to gain a pass grade, learners must meet all of the pass criteria.

For P1, learners must explain the features required to implement a given design. This refers to the features section of the unit content for learning outcome 1. In order to achieve this criterion, learners must describe the features clearly and logically, showing they have recognised the underpinning principles and in particular, the reasons why triggers and timing are important. A presentation or leaflet would be a suitable form of evidence.

For P2, learners must show that they are able to use event driven programming tools and techniques, including those listed in the unit content. A presentation would be a suitable form of evidence.

For P3, learners must design an event driven program. The program only needs to be basic, as suited to the level of learners, but obviously this is at the discretion of the tutor and the individual learners. The design should be clear and have no obvious errors.

For P4, learners must create the program they worked on for P3. This program should be fully functional, and fulfil the design aims set down.

For P5, learners must develop and apply an appropriate test plan for the program they worked on for P4. The test plan should test functionality and demonstrate that the program fulfils the design aims and other requirements. Evidence is likely to be in the form of a short report on the test plan and results, illustrated with screen grabs.

For P6, learners must produce onscreen help for an event driven program. The help screens must be coherent and laid out according to the standards that learners have previously been taught. It is up to the tutor at this stage whether they wish to have learners use the work they have produced for P4 and P5, or to give learners a generic event driven program for which to write the appropriate help.

In order to gain a merit, learners must achieve all the pass criteria, and all of the merit criteria.

For M1, learners must discuss how an operating system can be viewed as an event driven application. This will probably be the operating system that learners are working with but tutors may choose a different one if they wish. As with P1, evidence should be a poster, leaflet or short report but a presentation can also be used if the learner or tutor would prefer.

For M2, learners must justify their choice of tools and techniques used in the production of the event driven application created in P3. Evidence should be a short report, or similarly detailed presentation.

For M3, learners must analyse the results of their testing in P6. The analysis should compare expected to actual results to identify discrepancies. It would also be expected that learners would suggest what actions should be taken to resolve any problems shown up by the testing. Evidence for this criterion should be a short report. This could be an extension of the P6 report.

For M4, learners will create technical documentation for the support and maintenance of a computer program. As with P6, it is up to the tutor whether they wish to have learners use the work they have carried out for P4 and P5, or to give learners a generic event driven program. The documentation must be coherent and laid out according to the standards that learners have been taught previously.

In order to gain a distinction grade, learners must achieve all of the pass and merit criteria and both the distinction criteria.

For D1, learners will evaluate the suitability of event driven programs for non-graphical applications. Evidence for this criterion should be a short report, or similarly detailed presentation.

For D2, learners will evaluate an event driven application. There is the option here for the learner to review their P5 work, or to be given a generic program to review. Alternatively, the tutor could give learners the work of one of their peers to review. Evidence should be a short report, or similarly detailed presentation.

Programme of suggested assignments

The table below shows a programme of suggested assignments that cover the pass, merit and distinction criteria in the assessment and grading grid. This is for guidance and it is recommended that centres either write their own assignments or adapt any Pearson assignments to meet local needs and resources.

Criteria covered	Assignment title	Scenario	Assessment method
P1, P2, M1, D1	How to Start	You are working as a junior programmer for an electronic games maker. Your managers have asked you to write a short guide to the basics of event driven programming and to demonstrate some of the techniques.	Leaflet Presentation
P3, M2,	Design Work	You have been asked to design an example event driven program to demonstrate your coding skills.	Practical work Report
P4, P5, M3, D2	Implement and Test	Your design has been approved and you are now asked to implement and test the program.	Practical work
P6, M4	Finishing Off	Your managers would now like you to show how to write a set of support documentation for a program.	Practical work Short report

Links to other BTEC units

This unit forms part of the BTEC in IT sector suite. This unit has particular links with the following unit titles in the IT suite:

Level 1	Level 2	Level 3
		Unit 6: Software Design and Development
		Unit 15: Object Oriented Programming
		Unit 16: Procedural Programming
		Unit 22: Developing Computer Games

Essential resources

Learners will need individual access to a development environment that allows them to design and develop event driven applications. It is also advised that appropriate numbers of manuals and help sheets are readily available and easily accessible to learners.

Employer engagement and vocational contexts

The use of vocational context is essential in the delivery and assessment of this unit. Learners will require access to computer equipment to enable them to gain a practical awareness and enable them to apply their knowledge and understanding in a practical situation.

There are a range of organisations that may be able to help to centres engage and involve local employers in the delivery of this unit, for example:

- Learning and Skills Network
- Network for Science, Technology, Engineering and Maths Network Ambassadors Scheme
- National Education and Business Partnership Network
- Local, regional Business links
- Work-based learning guidance.

Delivery of personal, learning and thinking skills

The table below identifies the opportunities for personal, learning and thinking skills (PLTS) that have been included within the pass assessment criteria of this unit.

Skill	When learners are ...
Creative thinkers	generating ideas and exploring the possibilities to design an event driven application to meet defined requirements
Reflective learners	communicating their learning by creating onscreen help to assist users of a computer program
Self-managers	working towards goals, showing initiative, commitment and perseverance when developing an event driven application organising time and resources when developing an event driven application anticipating, taking and managing risks when testing an event driven application.

Although PLTS are identified within this unit as an inherent part of the assessment criteria, there are further opportunities to develop a range of PLTS through various approaches to teaching and learning.

Skill	When learners are ...
Independent enquirers	planning and carrying out research on an event driven application, appreciating the consequences of decisions supporting conclusions when giving reasons for the tools and techniques used in the production of an event driven application, using reasoned arguments and evidence analysing and evaluating an event driven application, judging its relevance and value
Reflective learners	reviewing progress on an object oriented application, acting on the outcomes communicating their learning by creating technical documentation for the support and maintenance of a computer program.

● Functional Skills – Level 2

Skill	When learners are ...
ICT – Using ICT	
Plan solutions to complex tasks by analysing the necessary stages	designing an event driven application
Select, interact with and use ICT systems safely and securely for a complex task in non-routine and unfamiliar contexts	implementing an event driven application
ICT – Finding and selecting information	
Use appropriate search techniques to locate and select relevant information	explaining the key features of event driven programs
ICT – Developing, presenting and communicating information	
Enter, develop and refine information using appropriate software to meet requirements of a complex task	implementing an event driven application
Combine and present information in ways that are fit for purpose and audience	creating onscreen help and technical documentation to assist users, and to provide support of a computer program
Evaluate the selection, use and effectiveness of ICT tools and facilities used to present information	giving reasons for the tools and techniques used in the production of an event driven application
Mathematics – Representing	
Understand routine and non-routine problems in familiar and unfamiliar contexts and situations	designing and implementing an event driven application
Identify the situation or problems and identify the mathematical methods needed to solve them	designing and implementing an event driven application
Choose from a range of mathematics to find solutions	designing and implementing an event driven application.