

Unit 16: Procedural Programming

Unit code:	L/601/7283
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 procedural programming applications.

● Unit introduction

Irrespective of framework or delivery platform, the development of procedural code is still at the core of many commercial applications development projects. Event driven systems and object oriented platforms all use procedural code for the critical command content of their objects, events and listeners.

This unit enables learners to become familiar with the underpinning concepts of procedural programming and subsequently to develop particular skills in a procedural programming language. The unit starts by looking at the features of procedural 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 thoroughly tested against the original user requirement.

Procedural programming languages include C and many others.

● Learning outcomes

On completion of this unit a learner should:

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

Unit content

1 Understand the features of procedural programming

Key features: procedures; functions; pre-defined functions; local variables; global variables; parameter passing; modularity; programming libraries

Control structures: eg fixed loop, pre-check loop, post-check loop, break-points; conditional commands eg if, case, conditional statements, use of Boolean operators

Programming languages: eg C

2 Be able to use the tools and techniques of a procedural language

Tools: eg functions, procedures, programming libraries

Variables: eg global, local, static

Techniques: parameters eg passing data, return values; debugging

3 Be able to design procedural applications

Specification: input; output; processes

Design: screen layouts; data storage; procedures; functions

Structure: selecting, declaring and initialising variables and data structure types and sizes

4 Be able to implement procedural 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 eg compiler errors, runtime errors, in code response, dry running; 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 procedural programs	M1 explain why modular elements are important for procedural programming	D1 evaluate the suitability of procedural programs for graphical applications
P2 demonstrate the use of procedural tools and techniques		
P3 design a procedural application to meet defined requirements [CT1]	M2 give reasons for the tools and techniques used in the production of a procedural application	
P4 implement a working procedural application to meet defined requirements		
P5 test a procedural application [IE2]	M3 analyse actual test results against expected results to identify discrepancies	D2 evaluate a procedural application.
P6 create onscreen help to assist the users of a computer program.	M4 create technical documentation for the support and maintenance of a computer program.	

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 allow them the opportunity to evidence all of the criteria fully. If this cannot be guaranteed then centres should not attempt to deliver this unit.

The suggested delivery pattern follows the order of the learning outcomes. 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. The learner will develop an application that must be procedural 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 many other solutions.

Tutors are advised that it is prudent to keep the delivery to one language, although many procedural languages now allow development in multiple platforms.

Learning outcome 1 covers all principles associated with the programming language selected. The advised delivery of the outcome is to cover all programming concepts whilst teaching the concepts of procedural 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 selected environment as well as the chosen programming language.

Implementation in learning outcome 3 must be based on a suitably structured problem that ensures use of more than two modular elements, a control structure and use of a conditional command.

Testing in learning outcome 4 must cover the code created in learning outcome 3 and designed for learning outcome 2. Software testing can be used to enhance (not replace) this learning outcome and give the learner 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 working environment and the HND project unit.

Centres 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 or extension to existing work. Therefore, learners completing this unit may contribute to many open source development projects or use them 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 procedural programming: <ul style="list-style-type: none">• whole-class exercise – tutor presentation on variables and their role in procedural 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 programming syntax• 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 modular elements and their role in procedural programming• individual exercise – use tutor-provided exercise materials to better understand the scope of variables• whole-class exercise – tutor presentation on parameters, and how they are used to pass data.
Assignment 2 – Design Work
Design a program: <ul style="list-style-type: none">• whole-class exercise – tutor presentation on 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 parameters• directed research – using tutor-specified sources, research programming standards• whole-class exercise – tutor presentation on mechanisms for testing procedural 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.
Assignment 4 – Finishing Off

Assessment

It is suggested that this unit is assessed using the four assignments 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 quite 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 learners' 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 the pass criteria.

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

For P2, learners must show that they can use procedural programming tools and techniques, including those listed in the unit content for LO2. A presentation would be a suitable form of evidence.

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

For P4, learners must create the procedural 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 procedural 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 a procedural program. The help screens must be coherent and laid out according to the standards that learners have been taught previously. 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 procedural program for which to write the appropriate help.

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

For M1, learners must explain the importance of modular programming. This is essentially an extension of P1, and is treated as such in the suggested assignments. As with P1, evidence should be a poster, leaflet or short report, although a presentation can also be used if learners or tutors prefer.

For M2, learners must justify their choice of tools and techniques used in the production of the procedural 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 highlighted by the testing. Evidence for this criterion should be a short report. This could be an extension of the report for P6.

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 done for P4 and P5, or to give learners a generic procedural 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 meet all of the pass criteria, all the merit criteria and both distinction criteria.

D1 is an extension of P1 and M1, and learners must evaluate the suitability of procedural programs for graphical applications. Evidence should be a short report, or similarly detailed presentation.

For D2, learners will evaluate a procedural application. There is the option here for the learner to review their P5 work, or to be given a generic procedural program to review. Alternatively, the tutor could give learners the work of one of their peers to review. Evidence for this 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 for a electronic games maker. Your managers have asked you to write a short guide to the basics of procedural programming and to demonstrate some of the techniques.	Leaflet Presentation
P3, M2,	Design Work	You have been asked to design an example procedural program demonstrating 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 14: Event Driven Programming
		Unit 15: Object Oriented Programming

Essential resources

Whilst some procedural languages are commercially available, there are also free languages available incorporating an diverse range of commands, commonly deployed on many platforms. Tutors must ensure that in the case of mobile platforms the applicable free emulators are available or where security policies dictate, local workstations are equipped with virtualised operating systems containing the programming environment.

Employer engagement and vocational contexts

Working with a local programming-based organisation or using internet-based open source projects, as well as source forge, would enhance learners' experience and offer a relevant vocational context.

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 possibilities to design a procedural program
Reflective learners	communicating their learning by writing support documentation for a procedural program.

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 ...
Effective participators	analysing and evaluating a given procedural program design, judging its relevance and value.

● Functional Skills – Level 2

Skill	When learners are ...
ICT – Using ICT	
Plan solutions to complex tasks by analysing the necessary stages	planning applications
Select, interact with and use ICT systems safely and securely for a complex task in non-routine and unfamiliar contexts	implementing and testing applications
ICT – Finding and selecting information	
Use appropriate search techniques to locate and select relevant information	researching standards
ICT – Developing, presenting and communicating information	
Combine and present information in ways that are fit for purpose and audience	producing documentation.