

Unit 20: Procedural Programming

Unit code: H/601/5099

QCF Level 2: BTEC First

Credit value: 10

Guided learning hours: 60

● Aim and purpose

The aim of this unit is to introduce learners to the features of procedural programming and the procedural development environment. They will develop the skills required to design, implement, test, document and review procedural programs.

● Unit introduction

Irrespective of framework or delivery platform, the development of procedural code is still at the core of many commercial applications development projects even though the perception may be that this is old-fashioned programming. Languages such as Cobol have been around for a long time and learners will consider them to be something from computing history. However, event driven systems and procedural platforms all use procedural code for the critical command content of their objects, events and listeners. The language chosen for study is not important as long as it allows learners to access all the assessment criteria.

The unit starts by looking at the features of procedural programming and the procedural environment. Learners will discuss how these types of language are used and their suitability for different uses. The unit then puts theory into practice and learners will design and develop procedural programs to meet specified needs. There are many concepts to be learned and practised to enable learners to design and implement programs confidently.

As with all programming, the focus is on developing solutions to meet identified needs is made along with emphasis on the importance of testing and reviewing. Good practice in testing and documenting programs will be emphasised and learners will develop techniques for ensure their program code is understandable and able to be maintained by others.

● Learning outcomes

On completion of this unit a learner should:

- 1 Know the features of procedural programming
- 2 Understand the procedural development environment
- 3 Be able to design and implement procedural programs
- 4 Be able to test, document and review procedural programs.

Unit content

1 Know the features of procedural programming

Key features: procedures; functions; pre-defined functions; local variables; global variables; parameter passing; modularity; programming libraries; suitability for graphical user interfaces(GUI); simplicity of programming; ease of development

Programming languages: eg Pascal, C, Cobol, Fortran, C++

2 Understand the procedural development environment

Environmental components: IDE; tool boxes and controls; toolbars; predefined functions; screen templates; help menus; debugging tools

3 Be able to design and implement procedural programs

Design: screen layouts; data storage; data structures; control structures; functions; data validation; error handling and reporting

Data structures: selecting, declaring and initialising variable and data structure types and sizes; operators eg logical, arithmetic, relational

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

Error handling: debugging eg: compiler/translator; errors eg logical, syntax, run-time

Data validation: eg type, range, presence, format

Implement: program and debug

4 Be able to test, document and review procedural programs

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

Review: against specifications requirements; suggest improvements

Documentation: listing of code; support and maintenance documentation

Programming standards: use of comments; code layout; indentation

Assessment and grading criteria

In order to pass this unit, the evidence that learners present 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 identify the key features of procedural programming	M1 describe the features of a procedural language that make it suitable for creating a GUI	D1 evaluate the suitability of procedural programs for non-graphical applications [IE4]
P2 explain how development environment components simplify the development process		
P3 design and implement a procedural program to meet defined requirements [IE1, CT1, CT6, SM2, SM3]	M2 implement data validation for inputs [CT1]	D2 implement error handling and reporting.
P4 test a procedural program [SM4]	M3 compare actual test results against expected results to identify discrepancies.	
P5 create documentation for the support and maintenance of a procedural program		
P6 review a procedural programme, suggesting possible improvements. [EP4]		

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

Essential guidance for tutors

Delivery

It is likely that, for most learners, this will not be their first contact with formal programming but it may be their first contact with procedural programming. It is very important that they use a methodical approach to creating procedural programs. This will be particularly valuable if should learners progress to higher-level courses that involve programming.

A number of small programs should be used to demonstrate procedural concepts. Any appropriate language can be chosen as the basis of the practical aspects of this unit. A procedural program should include a simple procedural design, fully documented structures, code, and interfaces (screens, forms, printouts, etc).

It is recommended that learners begin to program early delivery of this unit . To be most valuable, these programming activities should steadily increase in complexity and provide lots of opportunity for formative feedback.

Once learners have been given provided with the focus of an assessed task, they should be encouraged to break down the task and submit material in stages for assessment. These stages might naturally be design, production of program, testing/debugging and documentation.

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

Procedural programming:

- 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.

Designing a procedural program:

- 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
- mixture of tutor-led demonstration and learner exercises, repeated over a number of sessions.

Topic and suggested assignments/activities and assessment
Assignment 1 – How to Start Variables etc: <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 parameters • directed research – using tutor-specified sources, research programming standards • mixture of tutor-led demonstration and learner exercises, repeated over a number of sessions.
Assignment 2 – Design Work Testing and documenting: <ul style="list-style-type: none"> • whole-class exercise – tutor presentation on mechanisms for testing procedural coding • individual exercise – following a tutor presentation, prepare a set of support documentation • mixture of tutor-led demonstration and learner exercises, repeated over a number of sessions.
Assignment 3 – Finishing Off

Assessment

It is suggested that of this unit is assessed using three assignments as summarised in the *Programme of suggested assignments* table which follows this guidance.

The programme of suggested assignments table suggests that the theoretical elements (P1, P2, M1 and D1) are covered by producing a leaflet or poster, but these could be evidenced using a presentation or any other suitable format.

To achieve a pass grade, learners must achieve the pass criteria listed in the assessment and grading criteria grid.

P1 requires learners to identify the key features of procedural languages. A written report is probably the easiest to generate evidence but a presentation or any other suitable method can be used.

For P2, learners must explain how environmental components make the development process easier. This requires more than a straightforward explanation of each element outlined in the unit content, the emphasis being on their purpose.

For P3, learners should be guided to create some form of basic procedural design. Centres may incorporate use of diagrams if they feel this is appropriate. If learners use a graphic user interface (GUI) within their program then it is important that they show evidence of the planning and design of the GUI interface at an early stage.

For P3 and P4, before and after screenshots of the implemented program design will demonstrate debugging. Witness statements may be used to assess the completed design. For P4, test plans and results are required as well as evidence of dealing with error messages.

For P5, documentation should include annotated and properly laid out code (to programming standards) along with the technical documentation needed to support and maintain the program.

P6 covers the review process and should include suggestions for improvements. This can be included with the documentation for P5.

To achieve a merit grade, learners must achieve the pass grade criteria and the merit grade criteria.

For M1, learners must describe the features that make a procedural language suitable for creating a GUI. This does not relate to a specific language.

For M2, learners must show evidence of implementing data validation on input.

For M3, a test plan showing expected test results will need to be provided along with the actual test results.

To achieve a distinction grade learners must achieve all the pass and merit criteria and the distinction criteria.

For D1, learners must evaluate the suitability of using procedural programs for non-graphical applications. As with all evaluations, this should consider the pros and the cons.

For D2, learners should provide evidence of error handling and reporting.

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 Edexcel 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.	Leaflet or poster
P3, M2, D2	Design Work	You have been asked to design an example procedural program to show off your coding skills.	Design work Witness statement Screenshots etc
P4, P5, P6, M3	Finishing Off	Your managers would now like you to write a set of support documentation for a program.	Screen shots, program code etc Test plan Witness statement Review

Links to National Occupational Standards, other BTEC units, other BTEC qualifications and other relevant units and qualifications

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 16: Procedural Programming

This unit maps to some of the underpinning knowledge from the following areas of competence in the Level 2 National Occupational Standards for IT (ProCom):

- 6.1 Information Management.

Essential resources

Whilst some procedural languages are commercially available, there are also free languages available incorporating a diverse range of commands that are commonly used on many platforms. You 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.

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

- Learning and Skills Network – www.vocationallearning.org.uk
- Local, regional business links – www.businesslink.gov.uk
- National Education and Business Partnership Network – www.nebpn.org
- Network for Science, Technology, Engineering and Maths Network Ambassadors Scheme – www.stemnet.org.uk
- Work-based learning guidance – www.aimhighersw.ac.uk/wbl.htm
- Work Experience/Workplace learning frameworks – Centre for Education and Industry (CEI University of Warwick) – www.warwick.ac.uk/wie/cei.

Indicative reading for learners

Textbooks

Davis S R – *C++ for Dummies* (Paperback)

McBride P K – *Turbo Pascal Programming Made Simple* (Made Simple Computer)

Parkin A – *Cobol for Learners* (Arnold) ISBN 0-340-64552-0

Perry G – *Absolute Beginner's Guide to C* (Paperback, SAMS publication)

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 ...
Independent enquirers	identifying questions to answer and problems to resolve when designing and implementing a procedural program to meet defined requirements
Creative thinkers	generating ideas and exploring possibilities to create a simple procedural design for a program, and adapting ideas as circumstances change
Self-managers	working towards goals, showing initiative, commitment and perseverance when design and implementing an procedural program organising time and resources when designing and implementing a procedural program anticipating, taking and managing risks when testing a procedural program
Effective participators	identifying improvements when reviewing 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 ...
Independent enquirers	analysing and evaluating the suitability of procedural programs for non-graphical applications
Creative thinkers	generating ideas and exploring possibilities when implementing data validation for inputs.

● Functional Skills – Level 2

Skill	When learners are ...
ICT – Using ICT	
Select, interact with and use ICT systems safely and securely for a complex task in non-routine and unfamiliar contexts	implementing a program
ICT – Developing, presenting and communicating information	
Combine and present information in ways that are fit for purpose and audience	creating documentation for maintenance and support
Evaluate the selection, use and effectiveness of ICT tools and facilities used to present information	reviewing the program and suggesting improvements.

