
Unit title:	Procedural programming
Unit number:	16
Level:	4
Credit value:	15
Guided learning hours:	60
Unit reference number:	D/601/1293

UNIT AIM AND PURPOSE

This unit provides learners with a sound understanding of the principles of procedural programming and enables them to produce programs using a procedural programming language.

LEARNING OUTCOMES AND ASSESSMENT CRITERIA

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

Learning Outcome (LO)	Pass
The Learner will:	The assessment criteria are the pass requirements for this unit. The Learner can:
LO1 Understand the principles of procedural programming	1.1 discuss the principles, characteristics and features of procedural programming
LO2 Be able to design procedural programming solutions	2.1 identify the program units and data and file structures required to implement a given design 2.2 design a procedural programming solution for a given problem
LO3 Be able to implement procedural programming solutions	3.1 select and implement control structures to meet the design algorithms 3.2 correctly use parameter passing mechanisms 3.3 implement a procedural programming solution based on a prepared design
LO4 Be able to test procedural programming solutions	4.1 critically review and test a procedural programming solution 4.2 analyse actual test results against expected results to identify discrepancies 4.3 evaluate independent feedback on a developed procedural 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)	Distinction Criteria (D1, D2, D3)
(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.)	(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 procedural programming	
Characteristics	Low-level languages, high-level languages, generations
Principles	Comparison with other programming language types, uses (e.g. object-oriented, event-driven, logic programming, mark-up languages)
Programming standards	Use of comments, code layout, consistent indentation, sensible naming of variables, subroutines etc.
Data structures	Variables, arrays, file structures
Control structures	Loops, conditional statements, logical operators, input statements, output statements
Data types	Constants, literals, integer, byte, date, Boolean, others (depending on the language used).
LO2 Be able to design procedural programming solutions	
Requirements	Inputs, outputs, processing, user interface, hardware requirements, timescales, client, data handling, files
Program design	Design techniques such as flowcharts, structure diagrams, pseudo code, data flow diagrams, entity relationship diagrams
Program units	Data structures, control structures, data types
Documentation	Design specification, technical documentation.
LO3 Be able to implement procedural programming solutions	
Structures	Modules, as appropriate to the chosen language, control structures, data structures
Parameters	Data types, passing data, returning values, passing by reference, passing by parameter
Programming solution	Based on program design and specification, use of chosen language, following conventions.

LO4 Be able to test procedural programming solutions	
Testing strategies	Types of testing (black box, white box, alpha, beta, acceptance testing, etc.), test plan (including detailed test data), correction of errors (including syntax errors, runtime errors, logic errors), establish if requirements met
Test planning	Client, peers, other competent persons, identification of improvements, shortcomings
Analysis of results	Make improvements to a system based on testing results, discussion of the limitations of the system and what improvements could be possible
Onscreen help	User aids (with suitable HCI design), error messages, helpful dialogues, programme specific support, user instructions
Documentation	User guide, training materials, explanation of how the programme works, glossary, technical guide, explanation of coding decisions, hardware and software requirements, error handling, trouble shooting.

GUIDANCE

Delivery guidance
<p>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. Typical delivery contexts could include links with local employers, past students and community organisations.</p> <p>Learners will benefit from being encouraged to exercise autonomy and judgement to produce a solution to a programming problem, adapt their thinking and reach considered conclusions when analysing the requirements and producing the solution to a programming problem for a client.</p> <p>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, guest speakers may be able to discuss different approaches taken to analysing client needs and producing programming solutions.</p>

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

McGrath, M., *C Programming in Easy Steps* by, In Easy Steps Limited, 2012, ISBN 9781840785449

Stroustrup, B., The C++ Programming Language, Addison-Wesley Educational Publishers Inc, 2013, ISBN 9780321563842

McGrath, M., *C++ Programming in Easy Steps*, In Easy Steps Limited, 2011, ISBN 9781840784329

Gries, P., Campbell, J., Montojo, J., *Practical Programming: An Introduction to Computer Science Using Python 3*, The Pragmatic Programmers, 2013, ISBN 9781937785451

Dawson, M., *Python Programming for the Absolute Beginner*, Delmar Cengage Learning, 2010, ISBN 9781435455009

Websites

Procedural Programming Tutorials

www.wiziq.com/tutorials/procedural-programming

IDI Software Development and Project Methodology

www.idinews.com/

Programmers Stack Exchange

<http://programmers.stackexchange.com/>