Unit 18: Procedural Programming

Unit code: D/601/1293

QCF Level 4: BTEC Higher National

Credit value: 15

Aim

To provide learners with an understanding of the principles of procedural programming and to enable them to design and implement procedural programming solutions.

Unit abstract

Irrespective of framework or delivery platform, the development of procedural code is still at the core of many commercial application 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 allows learners to become familiar with the underpinning principles of procedural programming. Many languages have the capacity to develop procedural code and it is not important which language is chosen for this unit.

Ideally, for learners who are new to programming, this unit would be considered the starting point before progressing onto one (or all) of the many programming units. Whilst the learner is not expected to develop any complex code in this unit, the foundations will enable the development of their programming skills.

Learning outcomes

On successful completion of this unit a learner will:

1 Understand the principles of procedural programming

2 Be able to design procedural programming solutions

3 Be able to implement procedural programming solutions

4 Be able to test procedural programming solutions.

Unit content

1 Understand the principles of procedural programming

Characteristics of programming: low-level languages; high-level languages; generations eg first, second, third, fourth, fifth; programs; applications; instructions; algorithms

Types of language: procedural languages; object-oriented; event-driven; others eg script and mark-up languages; simple overviews and uses

Reasons for choice of language: organisational policy; suitability of features and tools; availability of trained staff; reliability; development and maintenance costs; expandability

Data structures: variables eg naming conventions, local and global variables, arrays (one-dimensional, two-dimensional); file structures; loops eg conditional (pre-check, post-check, break-points), fixed; conditional statements; case statements; logical operators; assignment statements; input statements; output statements

Data types: constants and literals; integer; floating point; byte; date; boolean; others eg character, string, small int; choice of data types eg additional validation, efficiency of storage

Programming syntax: command rules, variable declaration, *Standards*: use of comments, code layout, indentation

2 Be able to design procedural programming solutions

Requirements specification: inputs, outputs, processing, user interface; constraints eg hardware platforms, timescales for development; units; data; file structures.

Program design: tools eg structure diagrams, data flow diagrams, entity relationship models, flow charts, pseudo code

Technical documentation: requirements specification; others as appropriate to language eg form design, flowcharts, pseudo code, structured English, action charts, data dictionary, class and instance diagrams

3 Be able to implement procedural programming solutions

Modular design: elements eg functions, procedures, method, widgets, Graphical User Interface (GUI) components, symbols

Software structures: as appropriate to language chosen eg iteration, decisions, units, functions, procedures; control structures; conditional commands

Parameters: data types, passing data, return values

Scope of variables: global, local, static, overloaded results, instance

Programming: use of programming standards; relationship to program design

4 **Be able to test procedural programming solutions**

Mechanisms: valid declarations, debugging code, checking naming conventions; checking functionality against requirements, error detection, error messages, compiler errors, runtime errors, in code response, dry running

Supportive documentation: test plan; test results; programmer guidance; user guidance; onscreen help

Review: design against specification requirements, interim reviews

Learning outcomes and assessment criteria

Learning outcomes On successful completion of this unit a learner will:	Assessment criteria for pass 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.

Guidance

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

The learning outcomes associated with this unit are closely linked with:

Level 3	Level 4	Level 5
Unit 6: Software Design and Development	Unit 19: Object Oriented Programming	Unit 35 Web Applications Development
Unit 14: Event Driven Programming	Unit 20: Event Driven Programming Solutions	Unit 39: Computer Games Design and Development
Unit 15: Object Oriented Programming	Unit 21: Software Applications Testing	Unit 40: Distributed Software Applications
Unit 16: Procedural Programming	Unit 22 Office Solutions Development	Unit 41: Programming in Java
	Unit 23: Mathematics for Software Development	Unit 42: Programming in .NET

This unit has links to the Level 4 and Level 5 National Occupational Standards for IT and Telecoms Professionals, particularly the areas of competence of:

- Software Development.

Essential requirements

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

Learners must have access to facilities, which allow them the opportunity to fully evidence all of the criteria of the unit. If this cannot be guaranteed then centres should not attempt to deliver The learner must develop a procedural program that can work on a range of platforms, therefore it may be command line, web based, Graphical User Interface (GUI) based, games console or a deliverable for a mobile platform amongst many other solutions.

To ensure success centres must keep the delivery to one language. However, as many procedural languages now allow development in multiple platforms, learners may access this if it is locally realistic.

Centres must use a range of design methodologies, ensuring that the method selected is suited to the environment selected as well as the programming language of choice.

Implementation must be based on a suitably structured problem that ensures the use of modular elements, control structures and conditional commands.

Centres must 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 standalone application and may be an enhancement or extension to existing work.

Resources

Books

Davis S R – *C++ for Dummies* (Wiley, 2009) ISBN-10: 0470317264

McBride P K – *Turbo Pascal Programming Made Simple* (Made Simple, 1997) ISBN 0750632429

McGrath M – *C Programming in Easy Steps* (In Easy Steps Limited, 2009) ISBN 184078363X

Parkin A and Yorke R – *Cobol for Students* (Butterworth Heinemann, 1995) ISBN 0340645520

Websites

http://library.thinkquest.org/27297/

www.cplusplus.com/doc/tutorial/

www.cprogramming.com/

www.csis.ul.ie/cobol/

Employer engagement and vocational contexts

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