Unit 112: Computer Programming Techniques

Unit code: **D/602/2231**

QCF Level: **4**

Credit value: **15**

Aim

This unit aims to develop learners' understanding of computer programming techniques and will enable them to design and develop programs for a variety of applications.

Unit abstract

In this unit learners will design and develop program code in order to produce programs to a desired standard. They will use construct programs from designs, using appropriate functions and procedures. The unit will enable learners to produce and correctly present both user and technical documentation for programs. They will also construct and use test data and use appropriate techniques for detecting errors.

Programs should be written to defined quality standards and problem solving tools (structure diagrams, pseudo code etc) should be used. Emphasis should be placed on the need for modularity and an indication should be given of the link between modularity and object-based development.

Learning outcomes

On successful completion of this unit a learner will:

1 Be able to design and develop code using structured programming methods

2 Be able to use modularisation appropriate to the chosen programming language

3 Be able to produce appropriate documentation for a given program application

4 Be able to create and apply appropriate test schedules

Unit content

1 **Be able to design and develop code using structured programming methods**

Storage: the concepts of data storage within a computer program, using variables, constants and literals; for a third generation language, the pre-defined data types, integers, floating-point, character, Boolean (logical), strings, 1D and 2D arrays of simple types, and simple files, consequences of using these types, and the available operators within the supplied language

Control structures: identify and select appropriate iterative and selection structures when writing simple programs

Programming language syntax: the facilities and rules of the language (operators, I/0 commands etc)

Program design: employment of an algorithmic approach for the development of a solution to a problem (structure charts, pseudo code etc); producing tested programs to meet given specifications

Programming standards and practice: use of comments; code layout eg consistent indentation and descriptive identifiers

2 **Be able to use modularisation appropriate to the chosen programming language**

Use of functions/procedures: use/create functions/procedures both pre-defined and user-defined, map structured design onto a program using functions/procedures

Scope of variables: global, local, static and external variables

Parameters: passing data by value and reference, using return values

3 **Be able to produce appropriate documentation for a given program application**

Presentation of documentation: software applications (word processor or graphics); analysis, design and implementation documentation; professional standards; needs of industry

User documentation: user documentation for specified programming applications; purpose and operation of the program developed

Program documentation: documentation that covers technical aspects of a given programming application including algorithms implemented, data table, syntax (selection, iteration) structures used, user interface methods adapted

4 **Be able to create and apply appropriate test schedules**

Error types: semantic, syntax and run-time

Test documentation: test plan and related evidence of testing (may include reading sample inputs from a file and/or writing test results to a file)

Test data and schedules: black box, white box and dry testing

Error detection techniques: compiler and linker error messages, debugging tools and structured walk-through

Learning outcomes and assessment criteria

Learning outcomes **On successful completion of this unit a learner will:**	**Assessment criteria for pass** **The learner can:**
LO1 Be able to design and develop code using structured programming methods	1.1 select appropriate pre-defined data types 1.2 use simple input/output and appropriate operators with the above 1.3 use appropriate selection structures and loop structures for the given task 1.4 produce programs to desired standards
LO2 Be able to use modularisation appropriate to the chosen programming language	2.1 construct a program from a design and use appropriate functions/procedures 2.2 demonstrate the effect of scope and life-time of variables 2.3 pass data effectively between modules
LO3 Be able to produce appropriate documentation for a given program application	3.1 produce user documentation for a completed programming application including the user interface design 3.2 develop technical documentation for a predescribed program application
LO4 Be able to create and apply appropriate test schedules	4.1 demonstrate discrimination between semantic and syntax errors 4.2 produce test documentation 4.3 successfully construct and use test data and schedules to detect logic errors 4.4 use appropriate techniques for detecting errors

Guidance

Links

There are no links for this unit..

Essential requirements

Appropriate computer hardware and software is needed, as is a quality framework for the development of code.

Employer engagement and vocational contexts

Delivery of this unit will benefit from centres establishing strong links with employers willing to contribute to the delivery of teaching, work-based placements and/or detailed case study materials.