

**The Further Education and Training Awards Council (FETAC)
was set up as a statutory body on 11 June 2001
by the Minister for Education and Science.
Under the Qualifications (Education & Training) Act, 1999,
FETAC now has responsibility for making awards
previously made by FETAC.**



Module Descriptor

Computer Programming

Level 5 C20013

September 2001

www.fetac.ie

Level 5 Module Descriptor

Summary of Contents

Introduction	Describes how the module functions as part of the national vocational certificate framework.
Module Title	Indicates the module content. This title appears on the learner's certificate. It can be used to download the module from the website www.fetac.ie .
Module Code	An individual code is assigned to each module; a letter at the beginning denotes a vocational or general studies area under which the module is grouped and the first digit denotes its level within the national vocational certificate framework.
Level	Indicates where the module is placed in the national vocational certificate framework, from Level 3 to Level 6.
Credit Value	Denotes the amount of credit that a learner accumulates on achievement of the module.
Purpose	Describes in summary what the learner will achieve on successfully completing the module and in what learning and vocational contexts the module has been developed. Where relevant, it lists what certification will be awarded by other certification agencies.
Preferred Entry Level	Recommends the level of previous achievement or experience of the learner.
Special Requirements	Usually 'none' but in some cases detail is provided here of specific learner or course provider requirements. There may also be reference to the minimum safety or skill requirements that learners must achieve prior to assessment.
General Aims	Describe in 3-5 statements the broad skills and knowledge learners will have achieved on successful completion of the module.
Units	Structure the learning outcomes; there may be no units.
Specific Learning Outcomes	Describe in specific terms the knowledge and skills that learners will have achieved on successful completion of the module.
Portfolio of Assessment	Provides details on how the learning outcomes are to be assessed.
Grading	Provides details of the grading system used.
Individual Candidate Marking Sheets	List the assessment criteria for each assessment technique and the marking system.
Module Results Summary Sheet	Records the marks for each candidate in each assessment technique and in total. It is an important record for centres of their candidate's achievements.
Appendices	Can include approval forms for national governing bodies.
Glossary of Assessment Techniques	Explains the types of assessment techniques used to assess standards.
Assessment Principles	Describes the assessment principles that underpin FETAC approach to assessment.

Introduction

A module is a statement of the standards to be achieved to gain an FETAC award. Candidates are assessed to establish whether they have achieved the required standards. Credit is awarded for each module successfully completed.

The standards in a module are expressed principally in terms of specific learning outcomes, i.e. what the learner will be able to do on successful completion of the module. The other elements of the module - the purpose, general aims, assessment details and assessment criteria - combine with the learning outcomes to state the standards in a holistic way.

While FETAC is responsible for setting the standards for certification in partnership with course providers and industry, it is the course providers who are responsible for the design of the learning programmes. The duration, content and delivery of learning programmes should be appropriate to the learners' needs and interests, and should enable the learners to reach the standard as described in the modules. Modules may be delivered alone or integrated with other modules.

The development of learners' **core skills** is a key objective of vocational education and training. The opportunity to develop these skills may arise through a single module or a range of modules. The core skills include:

- taking initiative
- taking responsibility for one's own learning and progress
- problem solving
- applying theoretical knowledge in practical contexts
- being numerate and literate
- having information and communication technology skills
- sourcing and organising information effectively
- listening effectively
- communicating orally and in writing
- working effectively in group situations
- understanding health and safety issues
- reflecting on and evaluating quality of own learning and achievement.

Course providers are encouraged to design programmes which enable learners to develop core skills.

1	Module Title	Computer Programming
2	Module Code	C20013
3	Level	5
4	Credit Value	1 credit
5	Purpose	This module has been designed to introduce the learner to the concepts of programming and the techniques involved in constructing small programs. The module as described is language independent and includes only concepts which are fundamental to an imperative style of programming.
6	Preferred Entry Level	Level 4 Certificate, Leaving Certificate or equivalent qualifications and/or relevant life and work experiences.
7	Special Requirements	None.
8	General Aims	<p><i>Learners who successfully complete this module will:</i></p> <p>8.1 understand the concepts involved in programming</p> <p>8.2 be familiar with industry standard programming practices</p> <p>8.3 learn the principles of software design</p> <p>8.4 acquire skills to construct reliable software</p> <p>8.5 test programs effectively</p> <p>8.6 develop good work practices in the use and care of computing equipment.</p>

9 Units **The specific learning outcomes are grouped into 4 units.**

Unit 1	Programming Constructs
Unit 2	Array Processing
Unit 3	Character and String Processing
Unit 4	Procedures and Functions

10 Specific Learning Outcomes

Unit 1 **Programming Constructs**

Learners should be able to:

- 10.1.1 define a program
- 10.1.2 define a programming language
- 10.1.3 identify the different generations of programming languages
- 10.1.4 describe the relative advantages and disadvantages of each generation
- 10.1.5 distinguish between system software and application software
- 10.1.6 list examples of system software
- 10.1.7 list examples of application software
- 10.1.8 list the uses of an editor
- 10.1.9 use an editor to write simple program text
- 10.1.10 use standard editor facilities to include:
 - find and replace
 - block copy
 - block insertion
 - block deletion
- 10.1.11 lay out program text legibly
- 10.1.12 indent program text efficiently
- 10.1.13 document the program code
- 10.1.14 distinguish between a compiler and an interpreter
- 10.1.15 use a compiler to create executable code

- 10.1.16** execute a program and enter requested data
- 10.1.17** understand and use the following programming constructs:
- input/output
 - cursor and screen handling (position cursor, clear screen, reverse video)
 - assignment statement
- 10.1.18** explain what a variable is
- 10.1.19** distinguish between different simple data types such as integer, real, character and boolean
- 10.1.20** explain the syntax and semantics of the conditional statement
- 10.1.21** solve problems using an if .. statement
- 10.1.22** explain the syntax and semantics of an iteration (loop) statement
- 10.1.23** solve problems which require a loop construct as a solution
- 10.1.24** list the stages in constructing a loop
- initialise values of variables
 - place guard on loop
 - develop body of loop
 - progress towards termination
- 10.1.25** write code to read data and process it
- 10.1.26** explain the role of a sentinel (i.e. a value appended to a list to denote the end of the list)
- 10.1.27** devise an outline schema for processing lists
- e.g. `read(x)`
`while x <> sentinel do`
`begin`
`{process x }`
`read(x)`
`end`
- 10.1.28** explain the technique: top-down development
- 10.1.29** use the top-down strategy to devise a program to solve a simple problem
- 10.1.30** explain the need for data validation (e.g. to check if a month number entered is in the range: 1-12)
- 10.1.31** define the boolean operators and, or, not

- 10.1.32 construct compound boolean expressions
- 10.1.33 evaluate the truth value of compound boolean expressions
- 10.1.34 solve problems using boolean expressions
- 10.1.35 design data to test all the programming statements
- 10.1.36 test written programs with relevant data to check that the outputs are correct.

Unit 2 Array Processing

Learners should be able to:

- 10.2.1 explain why the data structure “array” (table) is necessary
- 10.2.2 define a linear (1-D) array
- 10.2.3 distinguish between the value of an element in an array and its corresponding index value
- 10.2.4 use arrays of different data types
- 10.2.5 construct loops to process the elements in an array
- 10.2.6 write a program to search (linearly) for an element in an array
- 10.2.7 solve problems whose solution requires the use of an array.

Unit 3 Character and String Processing

Learners should be able to:

- 10.3.1 explain the A.S.C.I.I. table
- 10.3.2 explain and list examples of control characters
- 10.3.3 justify the statement: the A.S.C.I.I. table is an ordinal set of values
- 10.3.4 explain the role of the extended A.S.C.I.I. set
- 10.3.5 write simple programs to process the character set (e.g. solve problems such as read a character and print one of “upper case letter”, “lower case letter” or “not alphabetic character”)
- 10.3.6 use the extended character set to draw graphical shapes

- 10.3.7 define a string
- 10.3.8 list the relational operators for strings
- 10.3.9 define the length of a string
- 10.3.10 distinguish between the length of a string and its dimension
- 10.3.11 write programs to process text data in the form of strings.

Unit 4 Procedures and Functions

Learners should be able to:

- 10.4.1 explain the need for procedures
- 10.4.2 define a procedure
- 10.4.3 write down the standard syntax for a procedure definition in the chosen language
- 10.4.4 write simple procedures without using parameters
- 10.4.5 write programs to test a given procedure
- 10.4.6 explain scope rules of variables
- 10.4.7 take a sample program and for each variable declared identify its scope
- 10.4.8 define a function
- 10.4.9 distinguish between user defined functions and standard functions such as cos, sqrt
- 10.4.10 write expressions which use standard functions
- 10.4.11 write user defined functions
- 10.4.12 test user defined functions
- 10.4.13 explain the difference between a function and a procedure.

11 Portfolio of Assessment

Please refer to the glossary of assessment techniques and the note on assessment principles at the end of this module descriptor.

All assessment is carried out in accordance with FETAC regulations.

Assessment is devised by the internal assessor, with external moderation by FETAC.

Summary	Assignments (2)	60%
	Examination (Theory -Based)	40%

11.1 Assignments (2)

The internal assessor will devise two briefs that require candidates to produce evidence that demonstrates a mastery of computer programming skills.

Candidates will be required to construct and test 2 computer programs. Each program will require the use of at least one of the following:

- the conditional statement
- a loop construct
- an array to store data
- strings and procedures.

Each assignment carries equal marks.

11.2 Examination

The internal assessor will devise a theory-based examination that assesses candidates' ability to recall and apply theory and understanding, requiring responses to a range of structured questions. These questions may be answered in different media such as in writing or orally.

The examination will be based on a range of specific learning outcomes and will be 2 hours in duration.

The format of the examination will be as follows:

4 structured questions

Candidates are required to answer 3 (40 marks each).

12 Grading

Pass	50 - 64%
Merit	65 - 79%
Distinction	80 - 100%

Individual Candidate Marking Sheet 1		Computer Programming C20013 Assignments (2) 60%
---	--	--

Candidate Name: _____ **PPSN.:** _____

Centre: _____ **Centre No.:** _____

Assessment Criteria	Maximum Mark	Candidate Mark	
		Assignment 1	Assignment 2
<ul style="list-style-type: none"> • clearly documented program code 	15		
<ul style="list-style-type: none"> • excellent screen layout 	15		
<ul style="list-style-type: none"> • effective code layout • accurate programming (syntactically and semantically) 	15 30		
<ul style="list-style-type: none"> • appropriate testing 	15		
Subtotal	90		
TOTAL MARKS <i>This mark should be transferred to the Module Results Summary Sheet</i>	180		

Internal Assessor's Signature: _____ **Date:** _____

External Authenticator's Signature: _____ **Date:** _____

Individual Candidate Marking Sheet 2	Computer Programming C20013 Examination (Theory -Based) 40%
---	--

Candidate Name: _____ PPSN.: _____

Centre: _____ Centre No.: _____

Assessment Criteria	Maximum Mark	Candidate Mark
Structured Questions 4 structured questions, answer any 3 (40 marks each) (Indicate questions answered) Question No.:* _____ _____	40 40 40	
TOTAL MARKS <i>This mark should be transferred to the Module Results Summary Sheet</i>	120	

Internal Assessor's Signature: _____ *Date:* _____

External Authenticator's Signature: _____ *Date:* _____

* The internal assessor is required to enter here the question numbers answered by the candidate.

FETAC Module Results Summary Sheet

Module Title: Computer Programming

Module Code: C20013

Assessment Marking Sheets

Maximum Marks per Marking Sheet

Mark Sheet 1
180

Mark Sheet 2
120

Total Marks
300

Total 3 100%

Grade*

<i>Candidate Surname</i>	<i>Candidate Forename</i>

Signed: _____
Internal Assessor: _____ Date: _____

This sheet is for internal assessors to record the overall marks of individual candidates. It should be retained in the centre. The marks awarded should be transferred to the official FETAC Module Results Sheet issued to centres before the visit of the external Authenticator.

- Grade*
- D: 80 - 100%
- M: 65 - 79%
- P: 50 - 64%
- U: 0 - 49%
- W: candidates entered who did not present for assessment

Glossary of Assessment Techniques

Assignment *An exercise carried out in response to a brief with specific guidelines and usually of short duration.*

Each assignment is based on a brief provided by the internal assessor. The brief includes specific guidelines for candidates. The assignment is carried out over a period of time specified by the internal assessor.

Assignments may be specified as an oral presentation, case study, observations, or have a detailed title such as audition piece, health fitness plan or vocational area profile.

Collection of Work

A collection and/or selection of pieces of work produced by candidates over a period of time that demonstrates the mastery of skills.

Using guidelines provided by the internal assessor, candidates compile a collection of their own work. The collection of work demonstrates evidence of a range of specific learning outcomes or skills. The evidence may be produced in a range of conditions, such as in the learning environment, in a role play exercise, or in real-life/work situations.

This body of work may be self-generated rather than carried out in response to a specific assignment eg art work, engineering work etc.

Examination

A means of assessing a candidate's ability to recall and apply skills, knowledge and understanding within a set period of time (time constrained) and under clearly specified conditions.

Examinations may be:

- practical, assessing the mastery of specified practical skills demonstrated in a set period of time under restricted conditions
- oral, testing ability to speak effectively in the vernacular or other languages
- interview-style, assessing learning through verbal questioning, on one-to-one/group basis
- aural, testing listening and interpretation skills
- theory-based, assessing the candidate's ability to recall and apply theory, requiring responses to a range of question types, such as objective, short answer, structured, essay. These questions may be answered in different media such as in writing, orally etc.

Learner Record

A self-reported record by an individual, in which he/she describes specific learning experiences, activities, responses, skills acquired.

Candidates compile a personal logbook/journal/diary/daily diary/record/laboratory notebook/sketch book.

The logbook/journal/diary/daily diary/record/laboratory notebook/sketch book should cover specified aspects of the learner's experience.

Project

A substantial individual or group response to a brief with guidelines, usually carried out over a period of time.

Projects may involve:

- research – requiring individual/group investigation of a topic
- process – eg design, performance, production of an artefact/event

Projects will be based on a brief provided by the internal assessor or negotiated by the candidate with the internal assessor. The brief will include broad guidelines for the candidate. The work will be carried out over a specified period of time.

Projects may be undertaken as a group or collaborative project, however the individual contribution of each candidate must be clearly identified.

The project will enable the candidate to demonstrate: (*some of these – about 2-4*)

- understanding and application of concepts in (specify area)
- use/selection of relevant research/survey techniques, sources of information, referencing, bibliography
- ability to analyse, evaluate, draw conclusions, make recommendations
- understanding of process/planning implementation and review skills/ planning and time management skills
- ability to implement/produce/make/construct/perform
- mastery of tools and techniques
- design/creativity/problem-solving/evaluation skills
- presentation/display skills
- team working/co-operation/participation skills.

Skills

Demonstration

Assessment of mastery of specified practical, organisational and/or interpersonal skills.

These skills are assessed at any time throughout the learning process by the internal assessor/another qualified person in the centre for whom the candidate undertakes relevant tasks.

The skills may be demonstrated in a range of conditions, such as in the learning environment, in a role-play exercise, or in a real-life/work situations.

The candidate may submit a written report/supporting documentation as part of the assessment.

Examples of skills: laboratory skills, computer skills, coaching skills, interpersonal skills.

FETAC Assessment Principles

- 1** Assessment is regarded as an integral part of the learning process.
- 2** All FETAC assessment is criterion referenced. Each assessment technique has **assessment criteria** which detail the range of marks to be awarded for specific standards of knowledge, skills and competence demonstrated by candidates.
- 3** The mode of assessment is generally local i.e. the assessment techniques are devised and implemented by internal assessors in centres.
- 4** Assessment techniques in FETAC modules are valid in that they test a range of appropriate learning outcomes.
- 5** The reliability of assessment techniques is facilitated by providing support for assessors.
- 6** Arising from an extensive consultation process, each FETAC module describes what is considered to be an optimum approach to assessment. When the necessary procedures are in place, it will be possible for assessors to use other forms of assessment, provided they are demonstrated to be valid and reliable.
- 7** To enable all learners to demonstrate that they have reached the required standard, candidate evidence may be submitted in written, oral, visual, multimedia or other format as appropriate to the learning outcomes.
- 8** Assessment of a number of modules may be integrated, provided the separate criteria for each module are met.
- 9** Group or team work may form part of the assessment of a module, provided each candidate's achievement is separately assessed.

© FETAC 2001

Any part of this publication may be copied
for use within the centre.



The development of the National Qualifications Framework is funded by the Department of Education and Science with assistance from the European Social Fund as part of the National Development Plan 2000-2006.

