Higher National Unit specification

General information for centres

Unit title: Software Development: Introduction

Unit code: DF6C 34

Unit purpose: This Unit is designed to enable candidates to gain an understanding of the fundamental techniques used in the development of computer programs. The Unit should introduce candidates to the process of program development beginning with the analysis of a problem through to designing, coding, testing and evaluating a solution in the form of computer program that meets the requirements of a given specification.

On completion of the Unit candidates should be able to:

1. Describe the process of software development
2. Use the basic structure and features of a programming language
3. Produce a computer program to meet a given specification

Credit value: 1 HN Credit at SCQF level 7: (8 SCQF credit points at SCQF level 7*)

**SCQF credit points are used to allocate credit to qualifications in the Scottish Credit and Qualifications Framework (SCQF). Each qualification in the Framework is allocated a number of SCQF credit points at an SCQF level. There are 12 SCQF levels, ranging from Access 1 to Doctorates.*

Recommended prior knowledge and skills: Access to this Unit will be at the discretion of the Centre. However, it is essential that candidates have prior experience of using a computer system. This may be evidenced by the possession of relevant National Units, HN units or relevant experience.

Core skills: There may be opportunities to gather evidence towards core skills in this Unit, although there is no automatic certification of core skills or core skills components.

Context for delivery: If this Unit is delivered as part of a group award, it is recommended that it should be taught and assessed within the subject area of the group award to which it contributes.

Assessment: The Unit will be assessed using two assessments. In Outcome 1 candidates are asked to provide brief description based responses to a set of 15 questions about the process of program development. The threshold of achievement for Outcome 1 will be set at 60% of the available marks.

General information for centres

Outcomes 2 and 3 will be assessed by a practical exercise or series of small exercises that will test candidates' knowledge and/or skills in designing, coding, testing and evaluating a solution in the form of computer program that meets the requirements of a given specification. Candidates will also be required to produce appropriate program documentation which conforms to organisational standards.

Higher National Unit specification: statement of standards

Unit title: Software Development: Introduction

Unit code: DF6C 34

The sections of the Unit stating the Outcomes, knowledge and/or skills, and evidence requirements are mandatory.

Where evidence for Outcomes is assessed on a sample basis, the whole of the content listed in the knowledge and/or skills section must be taught and available for assessment. Candidates should not know in advance the items on which they will be assessed and different items should be sampled on each assessment occasion.

Outcome 1

Describe the process of software development

Knowledge and/or skills

- ♦ Software development life cycle
- ♦ Categories of programming language
- ♦ Methods of program code translation
- ♦ Code levels

Evidence requirements

Candidates will need evidence to demonstrate their knowledge and/or skills by producing answers to a set of 15 questions requiring a written description demonstrating that they can:

- ♦ Describe the software development life cycle with reference to an appropriate model;
- ♦ Describe four categories of programming language;
- ♦ Identify at least one language for each of category of programming language;
- ♦ Describe methods of program code translation: interpretation and compilation;
- ♦ Describe code levels: source code and object code;

Each of the knowledge and/or skills must be covered in the assessment. The assessment will be open book and is to be completed under supervised conditions within 1.5 hours. Candidates must attain 60% of the available marks in order to obtain a pass in this outcome.

Assessment guidelines

Description based evidence requirements may typically attract a response of between 20 – 50 words per question. Candidates may benefit from scope to illustrate aspects of their responses.

Candidates who have access to a suitable workplace can base their assessment work on suitable client-side computer programming situations drawn from their place of work. Where a workplace situation is used, care should be taken to ensure that it would provide candidates with sufficient opportunity to meet the evidence requirements of the Unit.

It may be appropriate for the assessor to ensure that a particular workplace environment will enable candidates to generate sufficient and suitable evidence. Assessment in the workplace environment must be carried out under supervised conditions sufficient to ensure confidence in the authenticity of each candidate's submission.

Outcome 2

Use the basic structure and features of a programming language

Knowledge and/or skills

- ♦ Variables, arrays and data types are defined, initialised and used appropriately
- ♦ Language constructs and control structures are used effectively
- ♦ Standard file input and output is used effectively
- ♦ Readability of code is maintained
- ♦ An appropriate testing strategy is devised

Evidence requirements

Candidates will need evidence to demonstrate their knowledge and/or skills by using the basic structure and features of a programming language to produce program code containing evidence of the following:

- ♦ Declaration, initialisation and use of variables and arrays with meaningful names;
- ♦ Use of operators: arithmetic, logic, comparison;
- ♦ Use of selection;
- ♦ Use of iteration: conditional and unconditional;
- ♦ Use of nested control structures;
- ♦ Use of functions: predefined or user-defined;
- ♦ Data read in to the program from an existing file;
- ♦ Data from within the program is written to an external file;
- ♦ Readability of code maintained: internal documentation, indentation and naming conventions;
- ♦ Creation of an appropriate testing strategy with tests to be performed on normal and abnormal program usage and the result expected for each of the tests performed.

If being assessed as a discrete Outcome, the candidate will be required to produce a solution to a series of small problems. The evidence generated will cover each of the knowledge and/or skills and evidence requirements of this Outcome. Evidence must be produced in the form of a checklist of completed evidence, printout and soft copy of all relevant program code.

Higher National Unit specification: statement of standards (cont)

Unit title: Software Development: Introduction

In order to obtain a pass in this Outcome evidence must be provided showing that candidates have produced program code which satisfies each of the evidence requirements pertaining to the knowledge and/or skills of Outcome 2.

Assessment must be carried out in conditions sufficient to ensure the authenticity of the candidates work. The assessment will be open book and will most likely take place over an extended period with candidates having access to notes, reference books, the Internet and so forth. Candidates will not be permitted to use code taken directly from online resources, reference materials or reference personnel but may adapt and make use of code extracts from lecture notes.

Assessment guidelines

It is recommended that Outcome 2 be integrated with Outcome 3 and assessed by means of a larger, non-trivial, problem designed to assess the knowledge and/or skills and evidence requirements of both outcomes together.

Candidates who have access to a suitable workplace can base their assessment work on suitable software development situations drawn from their place of work.

Where a workplace situation is used, care should be taken to ensure that it would provide candidates with sufficient opportunity to meet the evidence requirements of the Unit. It may be appropriate for the assessor to ensure that a particular workplace environment will enable candidates to generate sufficient and suitable evidence.

Assessment in the workplace environment must be carried out under supervised conditions sufficient to ensure confidence in the authenticity of each candidate's submission.

Outcome 3

Produce a computer program to meet a given specification

Knowledge and/or skills

- ♦ Analyse a problem
- ♦ Produce paper based design
- ♦ Implementation of solution
- ♦ Testing and debugging of solution
- ♦ Evaluation of solution

Higher National Unit specification: statement of standards (cont)

Unit title: **Software Development: Introduction**

Evidence requirements

Candidates will need evidence to demonstrate their knowledge and/or skills by:

- ♦ Analysing and breaking down the problem
- ♦ Producing an outline design using one of: flowcharts, structure charts, pseudocode, or another appropriate design tool, which conforms to the given specification
- ♦ Producing program code that includes the use of variables, an array, a range of language constructs, control structures for selection and iteration and conforms to organisational standards in the use of appropriate indentation, naming conventions and internal documentation
- ♦ Producing a test strategy (pro-forma may be used) with data to test against normal and abnormal input and/or usage, indicating also the expected system response
- ♦ Using test strategy and test data to compare and log actual results against expected results;
- ♦ Using available debug facilities to resolve implementation issues
- ♦ Producing an evaluative summary stating: whether or not the solution solved the problem; the issues encountered; the issues outstanding; installation instructions and recommendations for improvement
- ♦ Producing a stand alone executable file or, if script, embedding of code into an appropriate environment

Specifications for candidates will require that a solution in the form of a computer program be devised to a non-trivial problem.

The program generated must closely match the specification. A testing strategy must be devised to ensure that the program produces, as far as possible, accurate results and meets the specification. Code must conform to organisational standards and must include the use of internal documentation and indentation.

Evidence must be produced in the form of a checklist and in the form of a printout and soft copy of program code and, where applicable, an executable file with all required support or library files (DLL's, text files, database files, etc).

In order to obtain a pass in this Outcome evidence must be provided showing that candidates have produced program code which demonstrates correct and consistent use of all of the elements above. The solution need not be fully operational in order for the candidate to achieve a pass, but must satisfy all of the knowledge and/or skills and evidence requirements being assessed.

The assessment must be carried out in conditions that ensure the authenticity of the candidates work. The assessment will be open book and will most likely take place over an extended period with candidates having access to notes, reference books, the Internet and so forth.

Higher National Unit specification: statement of standards (cont)

Unit title: Software Development: Introduction

Candidates will not be permitted to use code taken directly from online resources, reference materials or reference personnel but may adapt and make use of code extracts from lecture notes.

Assessment guidelines

Outcomes 2 and 3 could be assessed together by means of a larger, non-trivial, problem designed to assess the knowledge and/or skills and evidence requirements of both outcomes using a single instrument of assessment.

Candidates who have access to a suitable workplace can base their assessment work on suitable software development situations drawn from their place of work.

Where a workplace situation is used, care should be taken to ensure that it would provide candidates with sufficient opportunity to meet the evidence requirements of the Unit. It may be appropriate for the assessor to ensure that a particular workplace environment will enable candidates to generate sufficient and suitable evidence. Assessment in the workplace environment must be carried out under supervised conditions sufficient to ensure confidence in the authenticity of each candidate's submission.

Administrative Information

Unit code:	DF6C 34
Unit title:	Software Development: Introduction
Superclass category:	CB
Original date of publication:	December 2003
Version:	02 (April 2009)

History of changes:

Version	Description of change	Date
02	Outcome 1 Evidence Requirements changed to describing four programming languages.	24/04/09

Source: SQA

SQA acknowledges the valuable contribution that Scotland's colleges have made to the development of Higher National qualifications.

Higher National Unit specification: support notes

Unit title: Software Development: Introduction

This part of the Unit specification is offered as guidance. The support notes are not mandatory.

While the exact time allocated to this Unit is at the discretion of the centre, the notional design length is 40 hours.

Guidance on the content and context for this Unit

Terminology should be presented in context throughout the Unit.

Outcome 1 relates to the process of program development. Candidates will learn about the software development lifecycle and be able to describe at least one model that could be used to in development activity related to Outcome 3. Categories of programming languages such as object-oriented, scripting, declarative and mark up should be discussed, with candidates being able to name at least one programming language within each of the categories selected. Code translation and code levels are also covered by Outcome 1.

Outcome 2 relates to the use of a programming language. Initialisation of variables, arrays and the use of meaningful user-defined names should be emphasised. Candidates should gain experience of declaring, initialising and using one-dimensional arrays. If opportunities present themselves, it may be advantageous to the candidates learning experience and depth of understanding to construct and manipulate two-dimensional and three-dimensional arrays. Candidates should be able to use a range of operators, (e.g. arithmetic, comparators, logic, Boolean and so forth) and would ideally be made aware of operator precedence. Candidates should learn how to use predefined routines (e.g. procedures and functions) or create their own as appropriate to the programming language used. Candidates should also gain an understanding of the reasons for, and use of, Selection and Iteration. For selection, IF..THEN..ELSE and/or CASE structures would be appropriate. Pre-tested, post-tested and fixed iterations may be examined. Nesting of control structures should be utilised as appropriate. Language category specific features should be utilised as appropriate. For example, if using an event-driven language the candidate would be expected to have an awareness features such as events, event-handlers and methods. Candidates should learn how to read data in from an external text file (for example, reading from a comma delimited text file or database file into an array) and writing data from the program (for example, data from a variable or array to a comma delimited text file or database file).

Outcome 3 relates to the creation and testing of computer programs to meet a given specification. Candidates should learn how to design a solution before coding. Candidates should learn how to use at least one method of design (e.g. structure charts, structured English, flowcharts or any other suitable programming design method). Candidates should also learn how to generate programs to closely match any given specification and how to devise a simple testing strategy to ensure that programs produced, as far a possible, are robust, provide accurate results and meet the specification. Error trapping techniques would be beneficial to ensuring robustness and deepening the candidate's knowledge and understanding. Candidates should be encouraged to use any available debugging facilities as appropriate. Finally, candidates need to understand that code should conform to organisational standards, including the use of internal documentation and indentation. It is recommended that candidates are introduced to the idea of documenting programs and are taught how to implement appropriate internal documentation from the beginning of Unit delivery.

Candidates should have individual access to a workstation or PC. It is recommended that an integrated program development environment be used. A simple text editor may also be used, where appropriate.

Higher National Unit specification: support notes

Unit title: Software Development: Introduction

Guidance on the delivery and assessment of this Unit

Candidates should have opportunities to develop their practical skills throughout the Unit, and should be assessed at appropriate points. Terminology should be presented in context throughout the Unit. It is recommended that this Unit is delivered in Outcome order.

Outcome 1 is theoretical in nature and relates to the process of software development at an introductory level. Candidates should be introduced to an appropriate software development lifecycle suited to the style of delivery and proposed development language. Models that may be suitable for use include: Waterfall, Iterative, Spirial, Fountain, Rational (Booch), OPEN, (Concurrent) Incremental, DSDM, prototyping, evolutionary, etc. Candidates may benefit from an overview of a selection of models placed in context. Candidates should be able to describe their chosen model. For example, if the Waterfall model were to be chosen then it would be appropriate for the candidate to be able to identify and briefly describe stages within the model (stages may typically include, but are not restricted to: Requirements Specification, Analysis, Design, Implementation, Testing and Maintenance). It would also be appropriate for candidates to be aware of the importance of 'evaluation' as being a key, reflective, stage. Centres may also want to include stages such as: Feasibility study, documentation, integration and deployment.

Candidates should be able to describe categories of programming language and key characteristics (for example: procedures, events, event-handlers, methods, objects, classes, inheritance, etc). Categories that may be considered include, object-oriented, scripting, declarative and mark up.

Candidates should be able to describe in general terms the purpose of a compiler and interpreter and give examples of languages or programs that are interpreted or compiled. Candidates should also be able to describe the terms 'source code' and 'object code'.

Candidates will need evidence to demonstrate their knowledge and/or skills by producing answers to a set of 15 questions about the process of program development. Sampling is permitted provided that all knowledge and/or skills are evidenced within the instrument of assessment. It is suggested that the questions be structured in such as way that candidates must provide a brief descriptive response, typically in the region of 20-50 words, for each question. Some candidates may find it helpful if they were permitted to provide an illustration as part of a description. The assessment will be open book and is to be completed under supervised conditions within 1.5 hours. Candidates must obtain 60% of the available marks in order to obtain a pass in this outcome.

In Outcome 2 candidates will be assessed on: initialisation of variables and arrays, and the use of meaningful names, using a range of operators and operator precedence; using predefined routines and/or user defined functions; use of sequence, selection and iteration control structures. It would be appropriate for candidates to be provided with specifications that will require them to produce a number of simple computer programs coving the knowledge and/or skills Outcome 2 as formative exercises. The knowledge and skills gained from developing solutions to a range of simple problems may serve as building blocks to prepare candidates for tackling a larger, more complex problem that could cover assessed knowledge, skills and evidence requirements of Outcome 2 and Outcome 3.

Evidence should be produced in the form of a checklist and in the form of hard copies and disk copies of program code. In order to obtain a pass in Outcome 2 evidence should be provided showing that candidates have produced program code which demonstrates correct and consistent use of all of the Knowledge and/or Skills elements. The assessment will be open book and will probably take place over an extended period of time. Candidates should be allowed access to notes, reference texts, the Internet and so forth.

Outcome 3 relates to the creation and testing of computer programs to meet a given specification. A solution should be designed before coding commences, but this need not go to multiple levels of development. The design may take the form of structure charts, structured English, flowcharts or any other suitable programming design method. The program generated should closely match the given specification. A testing strategy should be devised to ensure that, as far as possible, programs produce expected results and meet the specification. Code should conform to organisational standards. This should include the use of internal documentation and indentation.

An example of a typical problem that a candidate may be given to solve, assessing Outcome 2 and Outcome 3, might be to produce a multiple choice quiz program in which questions, choices and answers are read into an array or series of arrays from a comma delimited text file. The program would also include an incrementing score counter for correct answers and loss of lives for incorrect responses. A text file with leader board information could be written from memory to another text file. Care should be taken to construct a scenario in which all of the assessed knowledge and/or skills and evidence requirements are covered.

Evidence should be produced in the form of a checklist and in the form of hard and disk copies of program code.

The assessment will be open book and will probably take place over an extended period of time. Candidates should be allowed access to notes, reference books, the Internet and so forth. However, candidates should not be permitted to use a solution taken directly from online resources, reference materials or reference personnel but may adapt and make use of code extracts from lecture notes.

In order to obtain a pass in this Outcome evidence should be provided showing that candidates have produced program code which demonstrates correct and consistent use of all of the Knowledge and/or Skills elements.

Higher National Unit specification: support notes

Unit title: Software Development: Introduction

Open learning

If this Unit is delivered by open or distance learning methods, additional planning and resources may be required for candidate support, assessment and quality assurance. A combination of new and traditional authentication tools may have to be devised for assessment and re-assessment purposes.

Disabled candidates and/or those with additional support needs

The additional support needs of individual candidates should be taken into account when planning learning experiences, selecting assessment instruments, or considering whether any reasonable adjustments may be required. Further advice can be found on our website **www.sqa.org.uk/assessmentarrangements**

General information for candidates

Unit title: Software Development: Introduction

This Unit is designed to enable you to gain an understanding of the fundamental programming concepts and structures required to produce computer programs.

On completion of the Unit you should be able to:

1. Describe the process of software development.

2. Use the basic structure and features of a programming language.

3. Produce a computer program to meet a given specification.

Throughout the Unit you will learn about the terminology used in software development.

In Outcome 1 you should learn about the process of program development. You will learn about the software development life cycle and be able to describe at least one lifecycle model. You will learn about categories of programming language and be able to identify languages within each category. The Outcome ends by looking at methods of program code translation and code levels. The assessment for Outcome 1 will involve you in producing answers to a set of 15 questions about the process of program development with 1.5 hours. You will need to gain 60% of the marks available in order to achieve a pass in Outcome 1.

In Outcome 2 you should begin to learn how to use a programming language. You will learn about variable and arrays, initialisation and the use of meaningful variable and array names, how to use a range of operators, e.g. arithmetic, comparators, logic and so forth, and about operator precedence. You will gain an understanding of the reasons for and use of Sequence, Selection and Iteration in programming in controlling programming flow and about the use of use predefined routines, e.g. procedures and functions, as appropriate to the programming language used. You should also learn how to input and output data from and to a file. You will also gain an appreciation of readability of code through the user of logical naming conventions, indentation and internal documentation.

In Outcome 3 you should learn how to develop and test computer programs to meet a given specification. You should also learn how to design a solution before coding your design into a program. You should learn how to use at least one method of design, e.g. structure charts, structured English, or flowcharts. You should learn how to generate programs, both simple and non-trivial or complex, to match any given specification and how to devise a simple testing strategy to help ensure that programs produce accurate results and meet the specification. You will also learn how to produce program documentation that conforms to organisational standards.

Outcomes 2 will either be assessed by a number of discrete practical exercises or, if integrated with Outcome 3, by a single larger problem that will test your knowledge and/or skills in analysing a problem, designing paper based solution, implementing the solution from the design, testing and bugging the solution, then evaluating the solution.

General information for candidates (cont)

In order to obtain a pass in Outcomes 2 and 3 you will be required to produce program code from a given specification which demonstrates correct and consistent use of all of the assessed elements within Outcomes 2 and 3 and to document your program code in accordance with organisational standards.