

SQA Advanced Unit specification: general information

Unit title: Developing Software: Introduction

Unit code: HP1R 47

Superclass: CB

Publication date: August 2017

Source: Scottish Qualifications Authority

Version: 01

Unit purpose

This unit is designed to enable candidates to develop basic software development skills. The design and implementation of the constructs of programming (variables, sequence, selection, iteration, functions and parameter passing) will be covered in the context of a development environment. Test plans, test cases and programme documentation will also be introduced. This introduction would provide a basis for further study in software development using a range of programming languages. This is a core unit for the SQA Advanced Certificate in Computing group award aimed at introducing candidates to the skills required to develop programmes using an appropriate development environment. The unit may also be studied on a standalone basis by a candidate with an interest in programming.

On completion of the unit the candidate should be able to:

- 1 implement and test code to carry out tasks following a given design.
- 2 prepare technical documentation in line with good practice.

Recommended prior knowledge and skills

Entry is at the discretion of the centre but it is recommended that candidates should have achieved the Core Skill of *Problem Solving* at SCQF level 5 or practical experience/appreciation of program design. A logical approach to problem solving would advantageous.

Credit points and level

1 SQA 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 National 1 to Doctorates.*

SQA Advanced Unit Specification

Core Skills

Achievement of this unit gives automatic certification of the following:

Core Skill component *Critical Thinking* at SCQF level 6

There are also opportunities to develop aspects of Core Skills which are highlighted in the support notes of this unit specification.

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.

The assessment exemplar for this unit provides assessment and marking guidelines that exemplify the national standard for achievement. It is a valid, reliable and practicable instrument of assessment. Centres wishing to develop their own assessments should refer to the assessment exemplar to ensure a comparable standard. Assessment exemplars are available on SQA's secure website.

Unit specification: statement of standards

Unit title: Developing Software: Introduction

The sections of the unit stating the outcomes, Knowledge and/or Skills, and evidence requirements are mandatory.

Please refer to *Knowledge and/or Skills for each outcome and the evidence requirements for the unit*.

Outcome 1

Implement and test code to carry out tasks following a given design.

Knowledge and/or Skills

- ◆ Make appropriate use of variables
- ◆ Use programming constructs (selection and iteration) within code
- ◆ Use functions and parameter passing
- ◆ Test the code using a recognised test strategy and given test cases
- ◆ Document the results of testing

Outcome 2

Prepare technical documentation in line with good practice.

Knowledge and/or Skills

- ◆ Apply meaningful names to variables and functions
- ◆ Ensure that code is well laid out to aid maintainability
- ◆ Internally document the code by adding comments
- ◆ Produce technical guide

Evidence requirements for the unit

Candidates will need to provide evidence to demonstrate their Knowledge and/or Skills by showing that they can successfully develop code in line with good practice.

Evidence for this unit will be in the form of working code and a technical guide which will include description of the system, detailed design (both assessor provided) a structured listing (commented and well laid out code) and completed test logs.

Candidates will produce evidence which demonstrates that they can successfully write error free code to carry out at least two different tasks from a given design. The given design should contain one example of selection (if statement or case statement) and one example of iteration (for loop, repeat until, do while). The design will also include at least two variables of different data types and determine the scope (global or local) and use of inbuilt functions and a user defined function which will include parameter passing.

In addition to the design documentation a statement of good practice must be provided to the candidate that will exemplify the expected coding standard acceptable and the expectations of the technical guide. This will include naming conventions for variables and functions, use of global and local variables, use of indentation and comments. The code produced must contain internal documentation to aid maintainability and to demonstrate the candidates understanding of the software development process.

SQA Advanced Unit Specification

Candidates will be presented with a top level design which is further broken down to a detailed second level design. Candidates will also be presented with a test plan which will allow thorough testing of the system. This will include a description of the test, test data and expected results. This will be completed by the candidate and will provide documentary evidence in the form of a test log which demonstrates that the code meets the desired result of the given test plan.

All evidence must be undertaken in controlled supervised conditions. Candidates may have access to notes and text and appropriate online help

Unit specification: support notes

Unit title: Developing Software: 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

The outcomes of this unit are aligned to selected parts of the National Occupational Standards (NOS) for IT and Telecoms at Pre-Entry/Junior Technician Role in Discipline 5 Solution Development and Implementation sub-discipline 5.2 and 5.3:

5.2 Software development:

Competence (5.2.J.1): Perform specified software development activities

- c) Correctly use specified software development procedures, tools and techniques to create software that meets given designs.*
- d) Functionally test that given designs have been met.*
- e) Use naming conventions and standards in line with organisational standards.*
- f) Use appropriate programming constructs to produce effective software.*

Competence (5.3.J.1): Carry out specified IT/Technology solution testing activities under direction

- a) Follow organisational standards for the systems development lifecycle.*
- b) Correctly use specified IT/Technology solution testing procedures, tools and techniques, as directed by superiors.*
- d) Assist with the collation and documentation of information relating to specified IT/Technology solution testing activities.*
- h) Perform specified IT/Technology solution testing activities as directed.*
- g) Document own software development activities.*

This unit is designed to develop basic skills in software development. Skills will be developed through practical activities but as this unit is a component unit within the *HR9J 47 Computing: Graded Unit 1 Examination* theoretical aspects of basic software development techniques must be demonstrated throughout practical activities. Formative assessment may also be used to support and develop a candidate's knowledge.

This unit is introductory and not designed to create extensive software development skills, but rather to ensure a foundation on which to build more complex skills.

It would be beneficial to schedule this unit at an early stage in a course programme, as it will develop a logical approach and understanding of many computing concepts.

SQA Advanced Unit Specification

The choice of development environment is at the discretion of the centre and could be influenced by the expected SQA Advanced Diploma that candidates are likely to progress to. If this is likely to be Software Development then a structured programming environment could be used. A windows based environment such as Visual Basic C++ or C# would not necessarily be suitable as the interface may distract from the coding aspects. For Technical Support or Networking candidates a scripting environment such as a shell script would be ideal and may be stressed to such candidates that writing scripts is likely to be used in job roles that they may follow. Use of a shell script is useful as this can use commands which could be demonstrated beyond the script, thus stressing that a program is simply a collection of commands. It may also be useful to include discussion on the appropriate selection of different types of environment for different uses.

It may be useful, for software development candidates, to introduce the software development life cycle and what happens at each stage including deliverables. This could provide a basis for delivery. Candidates should be encouraged to recognise the tasks that they carry out in relation to the life cycle.

By encouraging candidates to follow designs this should assist in developing an understanding of stepwise refinement and pseudo code by making use of these throughout delivery. It should be stressed that these documents are environment independent and are designed to aid the development process.

Different test strategies are likely to be introduced although it is unlikely that candidates will make use of more than one strategy. Test documentation should be introduced at an early stage and as a minimum normal, boundary and extreme testing concepts should be discussed in terms of creating robust code.

Different types and levels of documentation are likely to be discussed. Candidates will be encouraged to include comments as internal documentation in all their work. The difference between technical and user documentation and their contents may also be discussed. Candidates are required to provide their final submission in a technical document although there is no requirement for a user guide to be produced.

The concept of variables of different data types will be introduced and used in code. It is likely that these will have global scope in the script but local use of variables may also be appropriate. Candidates may be given program shells which have variables declared and the candidate will perform completion exercises. These may also be useful to demonstrate slightly more complex programming constructs.

Outcome 1

Candidates will be introduced to the concept of sequencing and program flow. This will be demonstrated by simple tasks

Candidates will be introduced to more complex constructs and program flow. Techniques used for decision making will be demonstrated and candidates should follow the flow of the program based on all eventualities of the decision.

Different types of loops will be demonstrated and candidates should understand how they are executed and how they can be used within code.

SQA Advanced Unit Specification

Use of functions will be demonstrated. This will include both user and predefined functions. The concept of parameter passing to both types of function will be introduced but it is likely that argument lists will be short.

Test plans and test logs will be used throughout coding exercises. It is recommended that all exercises undertaken by candidates throughout the learning process be accompanied by detailed design and test logs.

Outcome 2

This outcome relates to development of good practice. It would be useful if the centre produces a statement of standards at an early stage and this would then form the basis of good practice development throughout all exercises. It may be useful practice to present the candidates with code which is not commented or indented and have them bring this to the appropriate standard. This is useful to gauge understanding of coding.

Guidance on the delivery of this unit

This unit has a notional length of 40 hours. Time may be allocated as follows:

Environments and syntax	2 hours
Variables and data types	3 hours
Test strategies and techniques	2 hours
Decision making	10 hours
Repetition	10 hours
Functions	5 hours
Parameter passing	2 hours
Good practice and documentation	2 hours
Assessment	4 hours

Much of the delivery of this unit will be through practical activities. The concept of a simple script/program carrying out a simple task should be introduced as the starting point and this should then be further developed as each concept is introduced. Candidates should be encouraged to think in terms of input, processing and output and the concept of variables and data types and the temporary nature of these should be demonstrated by simple practical exercises.

The concept of detailed design should be an underlying aspect of this unit as should the importance of testing. It is recommended that all practical exercises be accompanied by design. As this is intended as an introductory unit importing the detailed design as comments into the development environment can be very useful.

Making decisions and the use of repetition should be introduced once candidates are comfortable with the basics. Formative assessment can play an important part in determining the correct time to introduce the complexities of simple decision making (if statements) which can then be developed to introduce more complexity (case).

The concepts of arrays are not included within this unit but these may be introduced if appropriate but must not be assessed. Most development environments include a range of predefined functions and expect a value to be passed in the form of a parameter. Use of these should be encouraged. The concept of passing data between functions without the need for declaring global or local variables will be discussed. This is best demonstrated in simple code and developed once confidence grows.

SQA Advanced Unit Specification

The use of completion exercises can be used successfully to build confidence. In addition coding that requires to be commented and tested are also useful tools for building confidence while encouraging, by demonstration, good practice in coding.

Test strategies and techniques should be covered. Much of this will be theoretical but candidates should be aware of when they may be used. The use of stub testing is a particularly useful technique in a learning situation.

Guidance on the assessment of this unit

It is suggested that a single program may be developed which will assess all requirements of the unit. Code may be produced which includes selection which will result in the two different tasks being carried out as a result of the selection. In addition a loop may also be included in this code (eg one of the tasks may involve a loop which will be executed a number of times).

Where possible the choice of environment should reflect the expected SQA Advanced Diploma that candidates will follow. Technical support candidates may develop a utility using a batch file or shell script that might offer a menu which carries out a range of tasks which interact with the system. Networking candidates may develop a similar script which maps network drives or reports on network activity. Software development candidates may produce a web script or a very small program that interacts with the system.

The use of functions must be included within the code. These may be included in the same script mentioned above or another script may be developed to demonstrate these.

The test log completed by the candidate may also be supplemented by the screen shots of execution.

Techniques to improve readability of code are required, such as indentation and detailed comments.

Centres should be encouraged to allow candidates to submit assessment evidence for this unit electronically.

It is recommended that a single instrument be used to assess this unit as a whole. Alternately the assessment for this unit may be administered in three separate parts and completion exercises could be used for these as follows:

Assessment guidelines

Outcome 1

Outcome 1 may be assessed by a separate instrument of assessment, the first of which carries out selection and decision making. The second would involve iteration and the third would involve functions and parameter passing. For each part candidates will be presented with a design which includes variables, detailed pseudo code and test plans. The test plans would require to be completed and the code fully commented. This may provide the candidate with valuable feedback on which they can further develop their skills. It should be noted, however, that the use of formative assessment can also assist in candidate feedback and development.

Outcome 2

If three separate instruments of assessment are used candidates would need to meet the requirements of Outcome 2 for each of the separate instruments of assessment. This would require three small technical guides to be produced.

SQA Advanced Unit Specification

Online and distance learning

This unit may be suitable for distance or online learning but centres should be aware that candidates undertaking this unit may require significant support.

Opportunities for the use of e-assessment

E-assessment may be appropriate for some assessments in this unit. By e-assessment we mean assessment which is supported by Information and Communication Technology (ICT), such as e-testing or the use of e-portfolios or social software. Centres which wish to use e-assessment must ensure that the national standard is applied to all candidate evidence and that conditions of assessment as specified in the evidence requirements are met, regardless of the mode of gathering evidence. Further advice is available in *SQA Guidelines on Online Assessment for Further Education (AA1641, March 2003)*.

Opportunities for developing Core Skills

This unit is based on development of software. Candidates will solve problems in the context of programming. Candidates will:

- use a computer system to develop software.
- apply critical thinking to develop solutions.

As candidates are working through this unit they will be developing aspects of the Core Skills in *Problem Solving* and *Information and Communication Technology (ICT)*.

This unit has the *Critical Thinking* component of *Problem Solving* embedded in it. This means that when candidates achieve the unit, their Core Skills profile will also be updated to show they have achieved *Critical Thinking* at SCQF level 6.

Equality and inclusion

This unit specification has been designed to ensure that there are no unnecessary barriers to learning or assessment. The individual needs of learners should be taken into account when planning learning experiences, selecting assessment methods or considering alternative evidence.

Further advice can be found on our website www.sqa.org.uk/assessmentarrangements.

History of changes

Version	Description of change	Date

© Copyright SQA 2012, 2017

This publication may be reproduced in whole or in part for educational purposes provided that no profit is derived from reproduction and that, if reproduced in part, the source is acknowledged.

SQA acknowledges the valuable contribution that Scotland's colleges have made to the development of SQA Advanced Qualifications.

FURTHER INFORMATION: Call SQA's Customer Contact Centre on 44 (0) 141 500 5030 or 0345 279 1000. Alternatively, complete our [Centre Feedback Form](#).

General information for candidates

Unit title: Developing Software: Introduction

This unit is designed to enable you to develop basic software development skills. The design and implementation of the constructs of programming (variables, sequence, selection, iteration, functions and parameter passing) will be covered in the context of a development environment. Test plans, test cases and program documentation will also be introduced.

This introduction will provide a basis for further study in software development using a range of programming languages.

On completion of the unit you should be able to:

- 1 implement and test code to carry out tasks following a given design.
- 2 prepare technical documentation in line with good practice.

Throughout this unit you will learn tools and techniques for basic software development using a development environment determined by your centre. The choice of language will be at the discretion of your centre.

You will learn how to code simple tasks and how this code interacts with the system. In addition you will learn to troubleshoot your code so that it runs error free and produces the desired results. This will involve rigorous testing and you will be introduced to the importance of testing and techniques that can be used.

You will be given designs to follow and this should be used to help you code your solutions.

This unit is assessed by practical assessment. All theoretical aspect of development will be demonstrated within a practical context.

On successful completion of this unit you will also gain the Core Skill component of Critical Thinking at SCQF level 6.