SQA Advanced Unit Specification: general information

Unit title: Software Development: Programming Foundations

Unit code: HP2P 47

Superclass: CB

Publication date: August 2017

Source: Scottish Qualifications Authority

Version: 01

Unit purpose

This Unit introduces candidates to generic fundamental programming constructs which are required as a base for software development. The Unit should also expand and consolidate the skills learned in the SQA Advanced Unit *Developing Software: Introduction* (HP1R 47) by introducing the importance of programming/scripting within computing. The Unit will allow candidates to understand the importance of good design and good programming practices within programming. The candidates should consolidate basic programming skills and introduce more complex programming program structures. The candidates should be able demonstrate understanding of the concepts of modularity, parameter passing and objects.

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

1 Identify different scenarios to apply programming constructs.
2 Make appropriate use of basic programming constructs.
3 Use advanced programming constructs appropriately.

Recommended prior knowledge and skills

Access to this Unit will be at the discretion of the centre, however it would be beneficial if the candidate already possessed good written communication, critical thinking and analytical skills, either through workplace experience or training at an appropriate level. It would also be beneficial if candidates had some prior experience of the basic programming constructs which could be evidenced by the SQA Advanced Unit *Developing Software: Introduction* (HP1R 47).

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.*

Core Skills

Opportunities to develop aspects of Core Skills are highlighted in the Support Notes of this Unit specification.

There is no automatic certification of Core Skills or Core Skill components in this Unit.

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.

Unit specification: statement of standards

Unit title: Software Development: Programming Foundations

Unit code: HP2P 47

The sections of the Unit stating the Outcomes, Knowledge and/or Skills, and Evidence Requirements are mandatory.

Please refer to the *Knowledge and/or Skills* for the Unit and *Evidence Requirements for the Unit* after the Outcomes.

Outcome 1

Identify different scenarios to apply programming constructs.

Knowledge and/or Skills

- Overview of the evolution and present uses of scripting/programming
- Main constructs in programming (sequence, selection, iteration)
- Comparisons/contrasts between Object Oriented (OO) programming and non OO programming
- Importance of Good Programming Practice

Outcome 2

Make appropriate use of basic programming constructs.

Knowledge and/or Skills

- Declaring/initialising variables
- Implementing selection constructs (If, case)
- Implement iteration constructs (fixed, while)
- Use of operators (Arithmetic, Logical, Boolean)

Outcome 3

Use advanced programming constructs appropriately.

Knowledge and/or Skills

- Implementing and controlling data structures
- Use of modular code (sub routines/functions/objects)
- Modular coding and argument passing
- Using good Programming practice

Evidence Requirements for the Unit

Candidates will need to provide evidence to demonstrate their knowledge and skills by Coding/implementing a number of small programs from given scenarios. These programs should cover a representative sample of the knowledge points from Outcomes 2 and 3.

The candidates should demonstrate understanding of the basic concepts outlined in Outcome 2 and the more complex concepts included in Outcome 3. The programs should also demonstrate the concept of good programming practice outlined in Outcome 1.

Candidates should use their commenting to demonstrate knowledge of the different constructs they are coding. One of these programs should also have a short report attached outlining the reasons for choosing a particular software language for a particular scenario.

As an alternative to traditional assessment methods (eg paper-based), candidates can provide a digital record of evidence to demonstrate Knowledge and/or Skills. Suggested approaches are outlined in the Support Notes, Guidance on the assessment of this Unit.

Unit specification: support notes

Unit title: Software Development: Programming Foundations

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

This Unit has been designed as an introduction to programming and is not specific to any particular programming language.

The Unit is specifically targeted at candidates undertaking an SQA Advanced Certificate in Computing. It is meant to deliver essential programming skills to candidates and provide a foundation for further studies into the broad topic of software development.

There is an opportunity to deliver the Unit using either an Object Oriented or a non-Object Oriented programming language. Although all candidates should be made aware of the importance of objects in many of the major programming languages, the programming constructs they will use are common to both Object Oriented and non-Object Oriented programming languages.

This Unit can also be used as a stand-alone Unit for candidates with an interest in developing basic programming skills.

Outcome 1: Contextualises the concepts of programming/scripting. Candidates should be given a brief overview of how programming has evolved. They should be introduced to the three key elements within programming of sequence, selection and iteration. Candidates should be made aware of the use of objects within programming as part of modular coding. Candidates should explore the different methods used to maintain good programming practice.

Outcomes 2: Gives candidates the opportunity to develop basic programming skills through a series of small programs.

Declaring and initialising variables

Candidates should understand the use of variables in applications and be able to declare and initialise them appropriately dependant on the language used and whether the language uses loose typing. They should understand the concept of simple data types and strings.

Using arithmetic, comparison and boolean operators

Candidates should be able to implement each of the operators appropriately.

Implementing control constructs

Candidates should be able to implement both selection and iteration constructs as appropriate to the programming environment used for delivery. They should be able to implement both fixed and while loops.

Outcome 3: Gives candidates the opportunity to develop advanced programming skills through a series of small programs.

Using data structures appropriately

Candidates should understand the concept of data structures and be able to implement them as appropriate to the programming environment used for delivery. The most common structure used is an array but other structures may be used dependant on the language used.

Using modular code

Depending on the programming environment selected candidates will be expected to be able to implement user defined code blocks, procedures, sub routines, methods or functions as appropriate.

Passing arguments using modular coding

They should be able to pass arguments to the code 'blocks' implemented. If an Object Oriented language is used then it may be deemed appropriate to demonstrate object building however this is at the discretion of the lecturer.

Good Programming Practice

Candidates should demonstrate standardised internal documentation. Candidates should also demonstrate knowledge of standardised naming conventions, indentation and use of white space.

Guidance on the delivery of this Unit

This Unit has been designed to allow centres flexibility in terms of the chosen development environment used for delivery. It has been written is such a way that the centre could deliver the Unit to candidates following a number of different paths within Computing but who need the basic structures utilised within programming.

Outcome 1

Given that the first Outcome is more theoretical in nature it could be used to provide candidates a solid basis of why programming has evolved as it has and the current uses of programming within a number of different strands within computing. If there is a specific path the candidates are embarking on then emphasis may be placed on why programming is important in their particular field. It is also highly recommended that this could be incorporated into the report as part of the open-book assessment.

Outcomes 2 and 3

Outcome 2 and 3 focus on programming constructs and since there is no documentation, test evidence required the vast majority of the time spent would involve the candidates writing code to particular designs. The centre also has an opportunity to deliver the Unit with two programming languages and highlight the differences/similarities between these languages.

Guidance on the assessment of this Unit

The open-book assessment for this Unit could be undertaken using an e-portfolio.

Assessment Guidelines

Open-book Assessment

Evidence for this Unit could be demonstrated by the candidates creating programs from a series of different scenarios. These programs should be documented appropriately demonstrating good programming practices and commented to show an understanding of the constructs used. The evidence could be submitted either by paper copy or e-assessment.

Online and Distance Learning

It would be perfectly feasible to develop a range of blended learning material to deliver this Unit by online means. Support for distance learners could be provided by both synchronous and asynchronous communication technologies. Care would need to be taken to ensure the authenticity of assessments undertaken by distance learners.

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 e-checklists. 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), SQA Guidelines one-assessment for Schools (BD2625, June 2005)*.

Opportunities for developing Core Skills

Although there is no automatic certification of Core Skills or Core Skill components in this Unit, there are opportunities for developing *Information and Communication Technology (ICT)* and *Problem Solving* Core Skills throughout the Unit.

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 to Unit

Version	Description of change	Date

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: Software Development: Programming Foundations

This Unit introduces you to the generic fundamental programming constructs which are required as a base for software development. The Unit should also expand and consolidate the skills you have learned in completing the SQA Advanced Unit *Developing Software: Introduction* (HP1R 47).

As you work through the Unit you will be introduced the importance of programming/scripting within computing. You will learn to understand the importance of good design and good programming practices within programming.

You should then be able to consolidate basic programming skills and introduce more complex programming program structures.

You should also be able to demonstrate an understanding of the concepts of modularity, parameter passing and objects.

On completion of the Unit you should be able to:

1 Identify different scenarios to apply programming constructs
2 Make appropriate use of basic programming constructs.
3 Use advanced programming constructs appropriately.