SQA Advanced Unit Specification

General information for centres

Unit title: Software Development: Data Structures and Performance (SCQF level 8)

Unit code: HT90 48

Superclass:	CB
Publication date:	May 2017
Source:	Scottish Qualifications Authority
Version:	01

Unit purpose

This is a specialist unit, intended to build on learners' existing or currently developing knowledge of data structures.

In a vocational context, data structures hold a key role in data storage and effective data manipulation. This unit offers an opportunity to compare data structures not only in the context of holding data, but also how different implementations offer different levels of performance. This unit aims to develop understanding of a range of commonly used approaches.

The study of the trade-off between aspects of implementation, such as the Time/Space dichotomy and an understanding of time complexity notation, such as the 'Big O' notation equips learners for both the workplace and further study.

The unit presents a number of important data structures and the algorithms that operate on them. There is also the opportunity to develop skills in matching the type of algorithm and its application to a given problem.

One of the key skills in developing software application is the ability to match a given problem space with a known solution to that particular type of problem and by developing a solution from these known and documented data structures and algorithms this skill can be enhanced.

For those who are planning on progressing to further courses of study this unit provides a solid set of knowledge and skills.

Outcomes

On successful completion of the unit the learner will be able to:

1. analyse time and space complexity of algorithms.
2. choose algorithms and data structures for a given problem.
3. implement a solution integrating multiple data structures.

Credit points and level

1 SQA Advanced Unit credit at SCQF level 8: (8 SCQF credit points at SCQF level 8)

Recommended entry to the unit

Access to this unit will be at the discretion of the centre, however it is recommended that learners should have prior experience of appropriate high-level languages and systems development. This may be demonstrated by possession of the SQA Advanced Certificate in Computing core units.

Core Skills

Opportunities to develop aspects of Core Skills are highlighted in the support notes for 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.

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**.

SQA Advanced Unit Specification: Statement of standards

Unit title: Software Development: Data Structures and Performance (SCQF level 8)

Acceptable performance in this unit will be the satisfactory achievement of the standards set out in this part of the unit specification. All sections of the statement of standards are mandatory and cannot be altered without reference to SQA.

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. Learners 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

Analyse time and space complexity of algorithms.

Knowledge and/or skills

- Time Complexity
- Space Complexity
- The Time/Space dichotomy
- ‘Big O’ notation

Outcome 2

Choose algorithms and data structures for a given problem.

Knowledge and/or skills

- Linear structures — search, delete and update
- Tables
- String processing
- Stacks and queues
- Recursion
- Sorting and searching
- Linked Structures — Lists
- Linked Structures — Trees
- Heaps
- Hash Tables

Outcome 3

Implement a solution integrating multiple data structures.

Knowledge and/or skills

- Data structures
- Interfaces to data structures
- Algorithms to operate on data structures
- Combining data structures to implement a small scale system

Evidence requirements for this unit

Learners will need to provide evidence to demonstrate their knowledge and/or skills across all outcomes.

Outcome 1

This outcome must be assessed by accurately analysing the operation of two different algorithms and associated data structures.

The learner will be given two different algorithms/data structures to analyse and must provide sufficient documentation to show:

- operation of the algorithm.
- data structure used.
- time complexity.
- space complexity.

As long as the data structures are dissimilar, two similar algorithms may be used, such as sorting and searching by using a search tree and sorting and searching using a heap.

Outcomes 2 and 3

These two outcomes can be assessed by implementing a small application using at least two different data structures. Each data structure and the algorithms operating on it must be accurately documented and tested. The application itself must also be accurately documented and tested.

Evidence may be captured, stored and presented in a range of media (including audio and video) and formats (analogue and digital).

All the assessments for this unit can be conducted under open book conditions. Assessors must satisfy themselves of the authenticity of learners' submissions. The *Guide to Assessment* provides further advice on methods of authentication.

SQA Advanced Unit Support Notes

Unit title: Software Development: Data Structures and Performance (SCQF level 8)

Unit support notes are offered as guidance and 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 aims to provide an expanded range of data structures and the algorithms that operate on them. These are typical building blocks for a wide range of programming tasks and also contribute usefully to those who are planning to progress to advanced courses.

Outcome 1 discusses the complexity of algorithm and provides a basis for deciding which implementation is most suitable for a given scenario, for example a less efficient algorithm may be appropriate for a small scale application, whereas a more efficient but more difficult to code may be required for a larger scale implementation.

Outcome 2 further develops concepts of the data structures introduced in Outcome 1 and the techniques required to implement instances of them.

Outcome 3 combines the structures developed in Outcome 2 to produce a small solution, illustrating the technique of building solutions by combining pre-existing structures.

Guidance on approaches to delivery of this unit

It may be possible to deliver this unit in conjunction with HP2K 48 *Software Development: Data Structures* or similar units.

Outcome 1 requires the analysis of algorithms and those that are associated with the data structures contained in Outcome 2 are suitable for this purpose. Therefore, it is suggested that these two outcomes can be taught in parallel with this.

Delivery for Outcome 3 is likely to be most effective when Outcomes 1 and 2 have already been covered.

Outcome 1 is probably best delivered with an amount of tutor led activities to ensure shared understanding of key concepts. It may prove useful to move investigations of different algorithms to individual research exercises.

Outcome 2 may benefit from some tutor led activities, but the bulk of the topics can be assigned as research opportunities, either as groups or on an individual basis.

Outcome 3 is a practical implementation outcome and will most likely be a student centred activity.

Guidance on approaches to assessment of this unit

The assessment for Outcome 1 requires the analysis of algorithms. This can be undertaken by producing a document outlining the operation of the algorithm and concluding with a note of the time complexity of the algorithm (In ‘Big O’ notation) and a note on the space requirements of the algorithm. Some data structures may exhibit different time complexities dependant on circumstance. Examples include adding to a binary tree — adding to a balanced tree will exhibit a different time complexity than adding to a degenerate tree. Such structures should be treated as one study, giving both best case and worst case performance. Two such studies are required.

Outcome 2 requires that learners implement two different data structures, and Outcome 3 requires that the learner implement a small application using two data structures. It would be sensible to choose data structures for Outcome 2 that form the basis for those required for Outcome 3. (There exists the possibility to use the algorithms analysed in Outcome 1 through the unit, but not at the expense of coverage of other algorithms).

Assessments for these outcomes should be conducted under open book conditions.

Opportunities for 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 learner evidence and that conditions of assessment as specified in the evidence requirements are met, regardless of the mode of gathering evidence. The most up-to-date guidance on the use of e-assessment to support SQA’s qualifications is available at **www.sqa.org.uk/e-assessment**.

This unit was developed with the intention that it be suitable for using e-assessment throughout. This does not preclude using traditional methods.

Opportunities for developing Core and other essential skills

Knowledge of a wide selection of data structures and the algorithms that operate on them is a fundamental component of the software developer’s skill set and a computational thinking mindset developed while undertaking the unit will be broadly transferrable. Operating on these structures may well involve utilisation and development of Core Skills such as *Numeracy, Communications, ICT* and *Problem Solving*. Opportunities exist, especially in formative sections, to develop team-working skills.

History of changes

Version	Description of change	Date

© Scottish Qualifications Authority 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.

For further information, please 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 learners

Unit title: Software Development: Data Structures and Performance (SCQF level 8)

This section will help you decide whether this is the unit for you by explaining what the unit is about, what you should know or be able to do before you start, what you will need to do during the unit and opportunities for further learning and employment.

This unit develops skills in analysing data structures used to build larger applications and the performance of algorithms used to create and maintain them. You will learn how a number of commonly used data structures are built and used.

You will learn how to compute the performance of these algorithms, both in terms of the speed that they operate at and the amount of space that they consume. You will also be shown how the speed of an algorithm is often bought at the cost of the amount of space required for it to run.

The data structures and algorithms presented in this unit are widely used in the production of computer-based solutions.

You will probably have some experience in writing computer programs before starting on this unit.

The assessment for the unit requires you to analyse two different data structures and report on their speed and space requirements (Outcome 1), implement two data structures (Outcome 2) and build a small application using two data structures (Outcome 3).

This unit provides a strong foundation in data structures for those progressing to further study.