

## **SQA Advanced Unit Specification**

### **General information for centres**

**Unit title:** Software Development: Object Oriented Programming-  
Advanced Concepts (SCQF level 8)

**Unit code:** HT93 48

**Superclass:** CB

**Publication date:** August 2017

**Source:** Scottish Qualifications Authority

**Version:** 01

### **Unit purpose**

This unit is designed to develop knowledge of a range of concepts, principles and techniques of object oriented software development which are beyond the introductory level. Learners will develop further their problem solving and object oriented technical skills. Learners will then be required to demonstrate their proficiency in these skills through the creation of object oriented software solutions to problems.

This unit will be suitable for learners who already have a broad understanding of the concepts, principles, and techniques of object oriented software development.

### **Outcomes**

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

- 1 investigate a range of advanced object oriented programming techniques.
- 2 implement a solution from an object-oriented design using specified advanced object-oriented techniques.
- 3 test the completed product using automated testing techniques.

### **Credit points and level**

1 SQA Credit at SCQF level 8: (8 SCQF credit points at SCQF level 8)

## **SQA Advanced Unit Specification**

### **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 a broad understanding of the concepts, principles, and techniques of object oriented software development. This may be demonstrated by possession of the SQA Advanced Unit HP2L 48 *Software Development: Object Oriented Programming*.

Alternatively, learners may have considerable practical work experience and a portfolio of programs which demonstrate their competence in object oriented software development.

### **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](http://www.sqa.org.uk/assessmentarrangements).

### SQA Advanced Unit Specification: Statement of standards

**Unit title:** Software Development: Object Oriented Programming-Advanced Concepts (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

Investigate a range of advanced object oriented programming techniques.

##### Knowledge and/or skills

- ◆ Interfaces
- ◆ Generics
- ◆ Exceptions
- ◆ Basic I/O
- ◆ Object Oriented Automated Testing
- ◆ Graphical user interfaces

#### Outcome 2

Implement a solution from an object oriented design using specified advanced object oriented techniques.

##### Knowledge and/or skills

- ◆ Creation and use of interfaces
- ◆ Implementation of code using generics
- ◆ Implementation of standard exception handling within a given object oriented programming language
- ◆ Implementation of user-defined exceptions
- ◆ Implementation of basic file I/O
- ◆ Implementation of basic graphical user interfaces

#### Outcome 3

Test the completed product using automated testing techniques.

##### Knowledge and/or skills

- ◆ Test plan using a defined strategy
- ◆ Test documentation
- ◆ Results of test runs
- ◆ Evaluation of test results

## SQA Advanced Unit Specification

### Evidence requirements for this unit

Learners will need to provide evidence to demonstrate their knowledge and/or skills across all outcomes by showing that they can investigate advanced object oriented programming techniques, apply them appropriately to a design, and test the finished product using appropriate automated testing techniques. This must be of sufficient complexity to cover the knowledge and skills for each outcome of this unit.

**Outcome 1** is knowledge based and requires that learners demonstrate their cognitive competence.

Evidence of this may be sampled across the knowledge domain which must encompass all of the knowledge statements in Outcome 1. If a traditional test is used to assess the learner's knowledge and understanding, this test should be timed and completed in a single assessment occasion ('sitting') and an appropriate pass mark should be set. Where reassessment is required, it should contain a different sample from that previously used.

### Outcomes 2 and 3

A learner's response can be judged to be satisfactory where the evidence produced shows the learner has successfully applied appropriate advanced object oriented programming techniques.

Learners will be required to implement a given object oriented design, including:

- ◆ Create and use Interfaces
- ◆ Develop code using Generics
- ◆ Implement standard Exception Handling within a given object oriented programming language
- ◆ Implement user-defined Exceptions
- ◆ Implement basic file I/O
- ◆ Implement basic Graphical User Interfaces

Learners will need to provide evidence to demonstrate that they can implement a test plan using a defined strategy, apply appropriate automated testing techniques and produce completed test documentation recording both the expected results of the test data and the actual results. The test data should be sufficient to adequately test the implemented solution in scope and range.

The learner will be expected to record and evaluate the results of the test runs. Where there are discrepancies between the expected results and the actual results, the coding must be amended and corrected accordingly.

Assessment of these outcomes should be conducted under open-book conditions. Assessors must assure themselves of the authenticity of each learner's submission.

### SQA Advanced Unit: Support Notes

**Unit title:** Software Development: Object Oriented Programming-Advanced Concepts (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

The purpose of this unit is to introduce the learner to a number of important object oriented programming concepts that are beyond the introductory object oriented concepts. The learner should already be familiar with the following object oriented concepts:

- ◆ Objects and classes
- ◆ Abstraction, encapsulation and information-hiding
- ◆ Inheritance
- ◆ Polymorphism
- ◆ Method overriding and overloading
- ◆ Association
- ◆ Aggregation and collection

Learners will acquire knowledge of advanced object oriented software development concepts and principles. Learners will then be required to implement advanced object oriented programming skills through the creation of object oriented solutions to problems.

In addition to the following advanced object oriented programming concepts:

- ◆ Interfaces
- ◆ Generics
- ◆ Exceptions
- ◆ Basic I/O
- ◆ Object Oriented Automated Testing

The unit also introduces the learner to:

- ◆ use basic concepts of implementing graphical user interfaces in an object oriented context.
- ◆ carry out automated testing in an object oriented context; for example Junit, which is a unit-testing framework for the Java programming language.

This unit forms part of the SQA Advanced Diploma in Computing: Software Development and should be delivered within the context of the group award. It would be suitable for learners who are proposing to follow a career in software development. It is not suitable as a stand-alone unit, but could be taken after successful completion of HP2L 48 *Software Development: Object Oriented Programming*, or fit into the second year of the SQA Advanced Diploma in Computing: Software Development.

### Guidance on approaches to delivery of this unit

Outcome 1 introduces learners to the following advanced object oriented concepts:

- ◆ Interfaces
- ◆ Generics
- ◆ Exceptions
- ◆ Basic I/O
- ◆ Object Oriented Automated Testing
- ◆ Graphical User Interfaces

Outcome 2 requires the learner to implement all of the concepts taught in Outcome 1, except for Object Oriented Automated Testing which is dealt with in Outcome 3.

Outcome 3 requires the learner to implement automated testing techniques, and would probably be best taught after Outcomes 1 and 2.

Outcome 1 could be treated as a theoretical outcome and taught first, but would probably be best taught together with Outcomes 2 and 3, thereby intertwining theory and practice as much as possible.

The implementation language chosen is not specified. At the time of writing, the most appropriate languages to use would be Java or C#. Both these languages are widely used in industry and have a similar, C-type syntax. Java is a good teaching language with a large object library, and the creation of user interfaces is supported through AWT and Swing. C# allows for the creation of user interfaces through the System.Windows.Forms namespace and classes of the .NET Framework.

It is suggested that delivery could be a mix of traditional classroom activities and online delivery.

It may be possible to deliver this unit either in conjunction with or subsequent to HP2L 48 *Software Development: Object Oriented Programming*.

### Guidance on approaches to assessment of this unit

Evidence can be generated using different types of assessment. The following are suggestions only. There may be other methods that would be more suitable to learners.

Centres are reminded that prior verification of centre-devised assessments would help to ensure that the national standard is being met. Where learners experience a range of assessment methods, this helps them to develop different skills that should be transferable to work or further and higher education.

The timing of the assessment for Outcome 1 should be at the discretion of the centre assessor. It is recommended that this is assessed towards the end of the unit, since it is intended that the learner should have practical experience of the concepts being assessed.

## SQA Advanced Unit Specification

It is recommended that Outcome 1 be assessed using a set of constructed-response questions. There should be at least one question for each of the following topics in the knowledge and skills statement of Outcome 1:

- ◆ Interfaces
- ◆ Generics
- ◆ Exceptions
- ◆ Basic I/O
- ◆ Object-Oriented Automated Testing
- ◆ Graphical User Interfaces

In order to achieve this outcome, learners must clearly explain all of the above concepts.

Assessment should be undertaken under open-book conditions.

At least half of the questions should change for each assessment opportunity.

It is recommended that Outcomes 2 and 3 be assessed holistically using a single case study. Assessors must assure themselves of the authenticity of each submission.

### 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](http://www.sqa.org.uk/e-assessment).

### Opportunities for developing Core and other essential 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.

Learners will naturally develop the Core Skill of *Problem Solving* at SCQF level 6 through designing an object oriented solution for a variety of problems that are presented as part of the teaching process, and also as part of assessment. The learners could be required to write a short report detailing their problem solving process, ie how they identified the best programming structures to use for the major program functionality, in order to fulfil this requirement.

## 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: Object Oriented Programming-Advanced Concepts (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 is designed to cover some of the more advanced concepts and skills required for a career in object oriented software development.

It is a non-introductory unit and assumes prior knowledge and proficiency in the fundamental concepts, principles, and techniques of object oriented software development. HP2L 48 *Software Development: Object Oriented Programming* is a recommended pre-requisite for this unit.

In this unit you will acquire knowledge of a range of more advanced concepts, principles, and techniques of object oriented software development necessary to enable you to design and develop object oriented software.

This will involve the following areas of learning:

- ◆ Using the features of an object oriented programming language, and in particular the following concepts and techniques, you will implement a software solution based on a given design. Your understanding and grasp of object oriented concepts and programming techniques will be reinforced throughout with practical exercises.
  - Interfaces
  - Generics
  - Exceptions
  - Basic I/O
  - Object Oriented Automated Testing
  - Graphical User Interfaces
  
- ◆ Using automated testing techniques and tools, you will test your software to ensure it works correctly and meets the user requirements. You will be required to amend any errors in your code in order to achieve a robust, reliable and efficient working program.

You may be assessed by a test for the knowledge in Outcome 1 and a case study for Outcomes 2 and 3.

You will naturally develop the Core Skill of *Problem Solving* at SCQF level 6 through designing an object oriented solution for a variety of problems that are presented as part of the teaching process, and also as part of assessment.