SQA Advanced Unit specification: general information

Unit title: Systems Development: Object Oriented Analysis and Design

Unit code: HP2M 48

Superclass: CB

Publication date: August 2017

Source: Scottish Qualifications Authority

Version: 01

Unit purpose

This Unit is designed to enable candidates to develop a knowledge of the theoretical concepts, underlying principles, scope and role of systems analysis and design undertaken within an object oriented environment. The Unit develops candidates' practical systems development skills and introduces candidates to a variety of requirements gathering and modelling techniques used in object oriented systems analysis and design, using UML or similar modelling notation. The emphasis will be on static conceptual modelling (class diagrams) and evolving behavioural models (use case models and sequence diagram modelling). It is recommended that this Unit is delivered in tandem with the Unit *Software Development: Object Oriented Programming* (HP2L 48) to give candidates an insight into the full development lifecycle.

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

1   Describe the object oriented paradigm.
2   Produce a static model of a system.
3   Produce a dynamic model of a system.

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 the candidate had prior experience of the software development process.

This may be evidenced by the possession of one or more relevant SQA Advanced Units such as: *Software Development: Systems Foundations* (HR8K 47) *Systems Development: Introduction* (HR8M 47) *and Software Development: Developing Small Scale Standalone Applications* (HP2N 47).

Credit points and level

2 SQA Credits at SCQF level 8: (16 SCQF credit points at SCQF level 8*)

*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.

The most appropriate approach to delivery is to integrate this Unit with a Unit that requires the candidate to carry out development of a computer system that utilises object oriented technologies. It is recommended that this Unit be delivered in conjunction with *Software Development: Object Oriented Programming* (HP2L 48).

Unit specification: statement of standards

Unit title: Systems Development: Object Oriented Analysis and Design

Unit code: HP2M 48

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

Describe the object oriented paradigm.

Knowledge and/or Skills

♦   Object oriented concepts
♦   Object oriented design models and techniques
♦   Object oriented system life cycle

Outcome 2

Produce a static model of a system.

Knowledge and/or Skills

♦   Construction of a class diagram
♦   Identification of attributes
♦   Identification of operations
♦   Visibility of attributes and operations (private, public, protected)
♦   Specification of appropriate association, aggregation and inheritance relationships between classes

Outcome 3

Produce a dynamic model of a system.

Knowledge and/or Skills

♦ Domain modelling:
  — Use case diagrams
  — Use case descriptions
♦ Dynamic modelling:
  — Sequence diagrams
  — Activity diagrams
  — Statechart diagrams
♦ Requirements validation

Evidence Requirements for the Unit

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.

Closed-book Assessment

Candidates will need to provide evidence to demonstrate their knowledge by showing that they can identify and critically analyse object oriented concepts, models, techniques and life cycle stages of object oriented design. The questions should cover the following areas:

♦ Objects and classes
♦ Attributes and operations
♦ Abstraction, encapsulation and data hiding
♦ Inheritance
♦ Polymorphism
♦ Association
♦ Aggregation
♦ Collaboration
♦ Coupling
♦ Cohesion
♦ The purpose of use case, class, and sequence diagrams
♦ The system life cycle for an object oriented systems development method

This assessment should be conducted under supervised closed-book conditions and the questions presented must change on each assessment occasion.

Open-book Assessment

Candidates will need to provide evidence to demonstrate their Knowledge and/or Skills by showing that they can:

♦ analyse a problem statement and identify requirements
♦ identify classes, attributes and operations using either Class, Responsibility, Collaboration (CRC) cards or natural language analysis
♦ produce a static model of a system by producing a class diagram that includes
  — visibility of attributes and operations (private, public, protected):
  — specification of appropriate association, aggregation and inheritance relationships between classes
♦ construct a use case diagram which models use cases, actors, and associations between actors and use cases
♦ construct a minimum of four use case scenarios which include pre and post conditions, trigger event and the best case scenario flow of events. Alternative or exceptional behaviour must be included in one or more use case scenarios
♦ validate requirements using use case scenarios and user interface walkthroughs
♦ construct a sequence diagram which shows the flow of messages between three or more objects, for one use case
♦ construct one other interaction diagram. This could be one of any of the following: activity, collaboration, statechart, component. The diagram must be appropriate for the scenario

This assessment is open-book. Assessors should assure themselves of the authenticity of each candidate's submission.

Unit specification: support notes

Unit title: Systems Development: Object Oriented Analysis
and Design

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 80 hours.

Guidance on the content and context for this Unit

The aim of the Unit is to introduce candidates to the principles and practices of object oriented analysis and design. The Unit provides candidates with the skills necessary to use a variety of object oriented tools, techniques and methods to analyse a problem and model and design a system.

The Unit will also provide candidates with an understanding of the defining features of object oriented analysis and design and how an object oriented analysis and design lifecycle fits into the general subject area of systems development.

The aim of domain modelling is to represent the main concepts (and their interactions) of the real world problem irrespective of whether a software solution will be used to deal with it. In specification modelling, a software solution is assumed and an abstract model of the services to be provided by the system is modelled. Implementation modelling specifies physical design of the system. This includes both static and dynamic modelling.

This Unit has been developed with the intention of taking the candidates through all phases of analysis and design from requirements gathering through to specifying the physical design of the system under consideration. Although it is envisaged that UML or a similar modelling notation is used to model and document analysis and design, the Outcomes of the Unit are generally specified to allow the Unit to be delivered using selected techniques from combinations of object oriented analysis and design methods. The choice of techniques and modelling notation may be matched to the requirements of the stakeholders, the type of system under consideration and development environment.

It is strongly recommended that this Unit be delivered in conjunction with *Software Development: Object Oriented Programming* (HP2L 48) to reduce assessment load and to show how the analysis and design techniques and system modelling can be applied to their programming tasks.

This Unit covers some of the skills described for a pre entry/Junior technician role in the National Occupational Standards — IT and Telecoms (2009). The main areas covered correspond to discipline 4.4 Systems Analysis and discipline 4.7 Systems Design. There are also ample opportunities within the Unit to address a range of skills at both foundation and intermediate level that are described in the National Occupational Standards for IT Users v3. The most likely areas to be covered would be Using the Internet and IT Software Fundamentals.

Guidance on the delivery of this Unit

This Unit has a notional length of 80 hours. Time may be allocated as follows:

Object oriented life cycle	6 hours
Object oriented concepts	16 hours
Domain modelling	12 hours
Natural Language Analysis	4 hours
UML static modelling	16 hours
UML dynamic modelling	18 hours
Assessment	8 hours

Though Outcome 1 is largely theoretical, much of the delivery of this Unit will be through practical activities. It is highly recommended that this Unit be delivered in conjunction with *Software Development: Object Oriented Programming* (HP2L 48) to reduce assessment load and to show how the analysis and design techniques and system modelling can be applied to their programming tasks.

Outcome 1 introduces candidates to the general principles of the object oriented approach to systems analysis and design. The candidates should be introduced to the fundamental object oriented concepts of classes, objects, persistence, encapsulation, inheritance and polymorphism. The candidates should be able to identify the processes that take place in the lifecycle and identify the major inputs and deliverables that result from each process.

In Outcome 1, candidates should be introduced to the object oriented software development life cycle, and how this relates to the concepts within this Unit. Candidates could be introduced to the following important object oriented concepts:

♦   Objects and classes
♦   Attributes and operations
♦   Abstraction, encapsulation and data hiding
♦   Inheritance
♦   Polymorphism
♦   Association
♦   Aggregation
♦   Collaboration
♦   Coupling
♦   Cohesion
♦   The purpose of use case, class, and sequence diagrams
♦   The system life cycle for an object oriented systems development method

SQA Advanced Unit Specification

Outcome 2 addresses requirements gathering and the creation of the static system model. The high level specification of the client's requirements should indicate all the users/actors that interact with each case of usage, and the relationship between cases of usage. This may be done using a diagram such as UML Use Case Diagram. These can then be modelled using CRC Cards. These cards can then be used in the creation of the class (conceptual) diagram for Outcome 3. Candidates could produce a low fidelity user interface prototype that will serve as the starting point for modelling user interface requirements. GUI design is often the easiest way to introduce the concept of use cases, as use cases are about the functionality of the system from the user's point of view, and the user is likely to understand the system functionality through the options available via the user interface. Finally, the models produced should be validated using use case scenarios and a user interface walkthrough.

This could all be applied to projects they are working on for *Software Development: Object Oriented Programming* (HP2L 48) in order to provide useful examples and make this less abstract.

Outcome 2 involves creating a static model of the system. Candidates can use their CRC cards and Use Case Diagram and Scenarios to identify the classes, attributes, and operations required for the system, and the relationships between the classes. The class diagram should indicate all appropriate classes and their relationships in terms of association, aggregation and inheritance. The model should also describe the attributes and operations needed to support defined cases of usage. It is expected that access types and data types would be documented for attributes, and that access types, return types, and parameters would be documented for operations.

In Outcome 2, candidates should be shown the process of getting from requirements analysis to producing a static model of the system (class diagram). Natural Language Analysis is a technique which candidates should be taught as this enables them to identify possible attributes, classes, objects, behaviours, and possible valid attribute values from any problem statement. It would also be beneficial to introduce candidates to Case-Based Reasoning. This is the technique of looking back at previous problems and their solutions and using similar cases as the basis for the design of a solution to the new problem. For example, candidates could be shown the problem of designing a noughts and crosses game, and as a class the solution to this problem could be designed. Candidates could then be asked to design a Connect 4 game, using the noughts and crosses game as a basis for their design due to the many similarities between the two games. Candidates must learn UML standards for creating a Class Diagram, and must be able to correctly model classes, attributes, operations, access and return types, and relationships (including inheritance, aggregation and association) between classes.

Outcome 3 involves creating a dynamic model of the system. This is important to show how objects will need to interact in order to perform important system processes. As with all modelling, it is important to focus on modelling that is most beneficial either for problem solving or communication, rather than steadfastly producing every possible type of diagram whether it is relevant or not. The Evidence Requirements therefore ask for use case diagrams and scenarios, and sequence diagrams. Use cases are excellent for communication with clients, and sequence diagrams help identify if there are flaws in the static model resulting in high coupling between objects. One other type of dynamic diagram should also be produced, depending on what is considered most useful for the given project. It is recommended that activity diagrams, collaboration diagrams and statechart diagrams are all covered, but only one needs to be assessed. Candidates should be shown examples of complex diagrams, but these should not be assessed if they are not listed in the Evidence Requirements for the Unit.

Outcomes 2 and 3 could be taught in order, or they could be inter-mingled and taught in parallel. Outcome 3 could be taught before Outcome 2 if preferred. This might be beneficial depending on what stage candidates are at in *Software Development: Object Oriented Programming* (HP2L 48) when you are moving on from Outcome 1.

Guidance on the assessment of this Unit

Outcome 1

The timing of Outcome 1 should be assessed at the discretion of the centre assessor. It is recommended that this is assessed last, since it is intended that the candidate should have practical experience of an approach to object oriented systems development and the techniques involved. The main focus in Outcome 1 should be on the object oriented systems approach and its defining features, boundaries and general concepts. It is recommended that Outcome 1 be assessed by a set of multiple choice questions.

Assessment must be undertaken in supervised conditions and is closed-book. Candidates may not bring to the assessment event any notes, textbooks, handouts or other material (calculators are not allowed). The questions presented must change on **each** assessment occasion.

Candidates must answer at least 60% of the questions correctly.

If a centre is presenting this assessment online the following assessment methods, where appropriate, may be selected:

♦ Multiple choice
♦ Drag and drop
♦ Multiple response
♦ Mix and match
♦ A combination of the above

There is an opportunity for a candidate to be assessed online subject to meeting the prescribed assessment conditions.

A candidate should complete this assessment within one hour.

Outcome 2

See guidance for Outcome 3.

Outcome 3

It is recommended that Outcomes 2 and 3 be assessed holistically using a single case study, and that these are linked to the project for *Software Development: Object Oriented Programming* (HP2L 48).

Online and Distance 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.

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

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.

Candidates will naturally develop the Core Skill of *Problem Solving* at SCQF level 6 through identifying requirements and designing an object oriented solution. The candidates could be required to write a short report detailing their problem solving process, ie how they identified a suitable class structure for the program given the initial problem statement, in order to fulfil this requirement.

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

General information for candidates

Unit title: Systems Development: Object Oriented Analysis and Design

This Unit is designed to enable you to develop a knowledge of the theoretical concepts, underlying principles, scope and role of systems analysis and design undertaken within an object oriented environment.

The Unit develops your practical systems development skills and will introduce you to a variety of requirements, engineering techniques and the main modelling and diagramming techniques used in object oriented systems analysis and design, using UML or similar modelling notation.

The Unit will also develop your appreciation of the boundaries, strengths and limitations of object oriented systems analysis and design so that you can select the most appropriate tools and techniques for undertaking analysis and design given a specific project context.

The study of this Unit will provide a strong foundation for anyone who will be developing object oriented software, and systems in an object oriented based environment.

On completion of this Unit you should be able to:

1   Describe the object oriented paradigm.
2   Produce a static model of a system.
3   Produce a dynamic model of a system.