Support Materials

SQA Advanced Unit: Assessment Exemplar

Software Development: Object Oriented Programming — HP2L 48

SCQF level 8

Publication date: August 2017
Publication code: HP2L 48/AEX001/AQ

Published by the Scottish Qualifications Authority
The Optima Building, 58 Robertson Street, Glasgow, G2 8DQ
Lowden, 24 Wester Shawfair, Dalkeith, Midlothian, EH22 1FD

www.sqa.org.uk

Assessment tasks

The following information indicates the evidence which should be retained for external verification.

Assessment task 1: Project

Outcomes covered: 1–3

Evidence to be retained: All candidate evidence and assessor checklists

Other related Units

This Unit has been validated as part of the SQA Advanced Certificates/SQA Advanced Diplomas in Computing at level 8 within the Scottish Credit and Qualifications Framework (SCQF).[1] Centres are required to develop the assessment instrument in accordance with the validated Unit specification. Related Units are:

Unit code	SCQF level	Title
HP2M 48	8	Systems Development: Object Oriented Analysis and Design

Core Skills

The Unit specification will detail the Core Skills covered within the Unit.

Where Core Skills have been embedded in a Unit specification and an assessor wishes to use an alternative method of assessment, s/he must ensure that the assessment generates the necessary evidence as specified by the Evidence Requirements in the Unit specification. It is recommended that the centre seek prior verification for the alternative method to ensure that the Core Skill is still covered.

There are no Core Skills embedded in this Unit.

[1] The SCQF provides the national common framework for describing all relevant programmes of learning qualifications in Scotland. The level a qualification is assigned within the framework is an indication of how hard it is to achieve. There are 12 levels, from level 1 for National 1 through to level 12 for doctorates. For further information on the level and credit rating see the Unit specification. For further information on SCQF go to www.scqf.org.uk.

Conditions of assessment

There is one assessment task for this Unit. The task is a holistic assessment that integrates assessments for Outcomes 1, 2 and 3 and takes the form of a project based on a given brief.

This assessment is open-book and will take a period of approximately 30 hours to complete. This assessment is designed for individual work.

It is recommended that the project has a minimum of two deadlines: one for the program implementation and one for testing the implemented solution, to stagger the assessment load and allow for feedback on the implementation before the final testing is completed.

Candidate's evidence should be marked after each stage and an opportunity for remediation given, if required, using the same brief.

In the event that the final submission is not completed or is not completed to the standard required, the candidate may be re-assessed using a different brief.

Assessors should ensure themselves of the authenticity of the candidate's evidence.

Assessment task 1

Outcomes covered 1–3

Assessment task instructions

Project instructions

There is one assessment for this Unit. This is an open-book project covering all Outcomes. The project is broken down into two stages. Stage 1 is the program implementation and Stage 2 is the testing of the completed program.

You are required to implement a computer game based on the design documentation. This documentation will either have been produced by you for the Unit *Systems Development: Object Oriented Analysis and Design* (HP2M 48), or will be provided for you. All the Evidence Requirements which you must achieve are detailed after the client brief.

This project will be carried out under supervised and unsupervised conditions, ie you may work on this in your own time. The assessor will check the authenticity of any work you have done unsupervised. This may involve methods such as interviews, demonstrations, checking files, etc and may be carried out at random and pre-arranged times.

The assessor will specify the various deadline periods for the project. It is up to you to determine your own deadlines within these. You may decide to work on multiple tasks at the same time but you should try to fully complete and achieve one stage before completing the next. Applying this method of working is good preparation for the SQA Advanced Diploma Graded Unit.

You should read all the Evidence Requirements for each stage and clarify any points with the assessor before you commence the project.

You are required to read the following brief and then complete the stages detailed below:

Client brief

Retro Games Ltd has commissioned you to design and develop a simplified version of Sokoban. Sokoban is a popular game by which a player pushes **crates** around a **map** to get them all in the right location. Sokoban is a single player game. The game is played on a 2 dimensional grid, but the rooms are not usually of regular shape. The edges of the room are indicated by a wall, and the player and boxes cannot get through the wall. There is a warehouse keeper, who the player must control in order to move the crates from their starting positions onto the diamonds. The **diamonds** are the end points for the crates. You can only push a crate when you are to one side of it and its opposite side is clear, which makes the task somewhat tricky for more complicated maps.

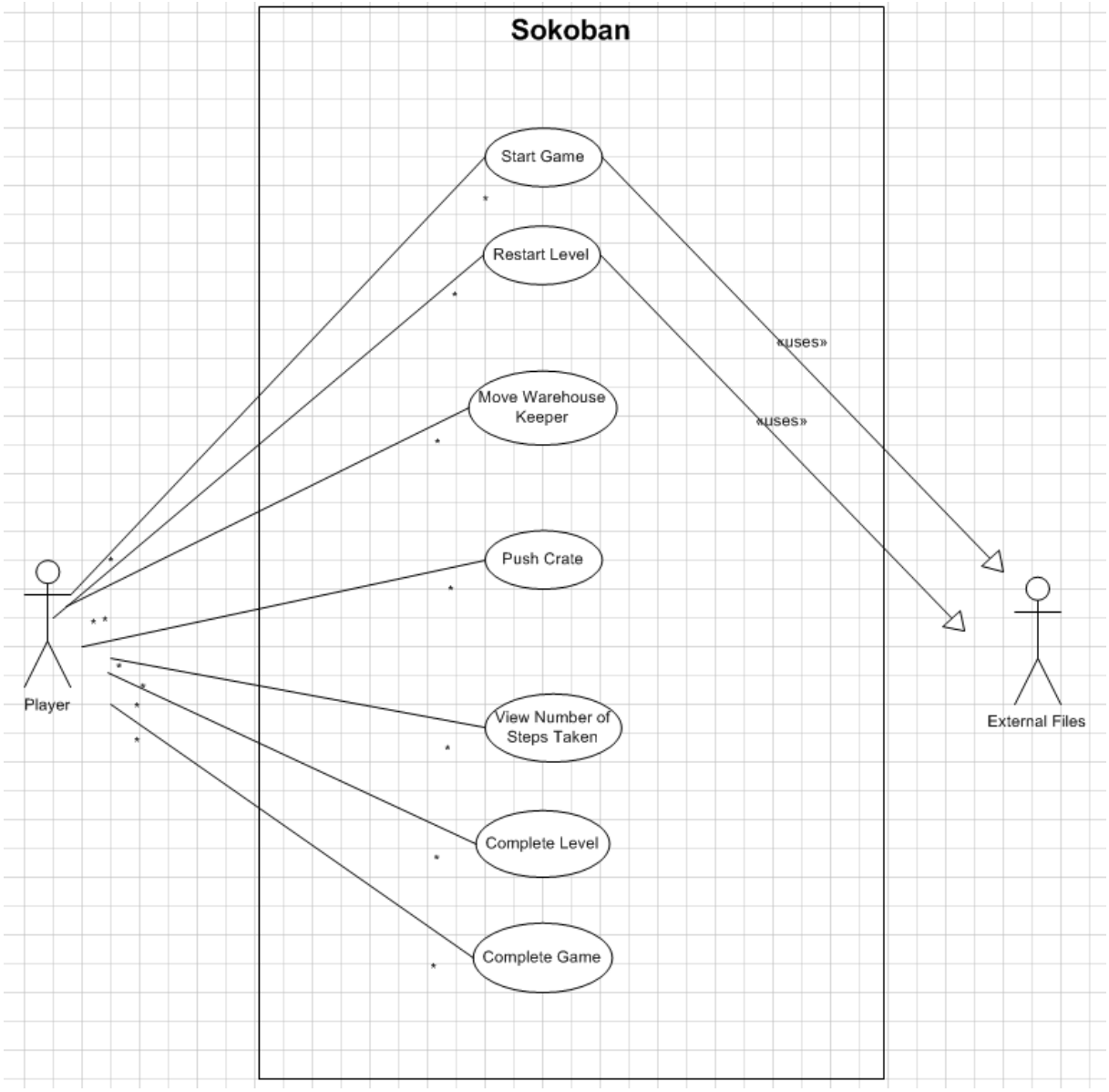
There are **walls** all around the map, and also in the middle in various configurations. Crates cannot be pushed through walls. Once a crate is up against a wall you can only push it along the wall, as you need to get behind a crate in order to push it. Once a crate is in a corner it is impossible to move it again. The warehouse keeper is unable to climb over crates, and is only strong enough to move one crate at a time. Crates can only be pushed, not pulled.

The game will require at least five levels. Each level should be harder to solve than the previous one, either by having more crates or obstacles, or tighter corridors, or a more complex starting arrangement of crates. The program should record how many moves a player takes to solve a level, and output this information visually.

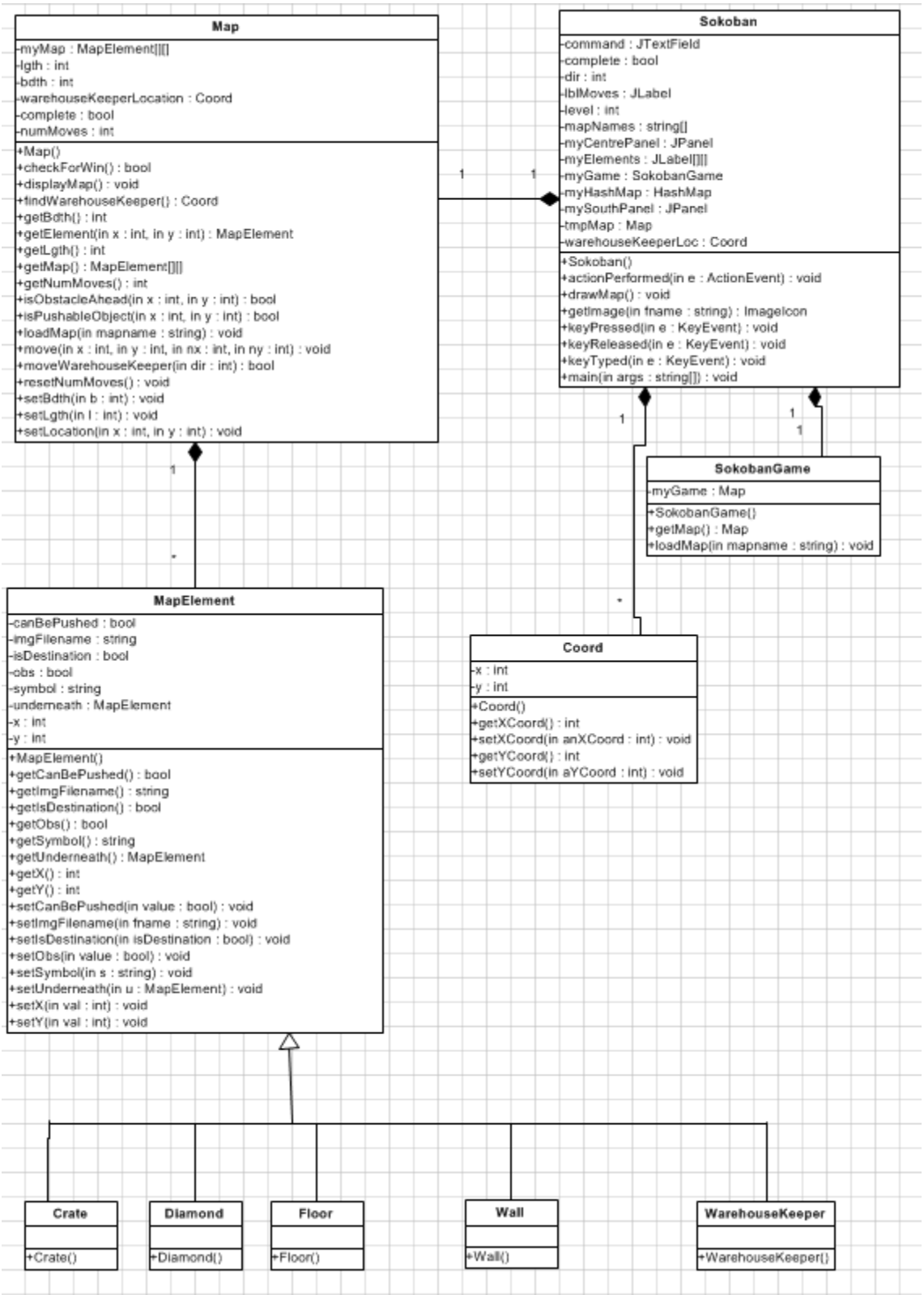
Design documentation

If the candidate has successfully completed *Systems Development: Object Oriented Analysis and Design* (HP2M 48) they may use the design documentation they produced for that Unit as the basis of their program implementation. Alternatively, candidates may be given these diagrams as the basis of their program design.

Use case diagram



Class diagram



Outcomes 1 and 2

Stage 1 — Implementation

You are required to implement the program in an object oriented manner. The solution must demonstrate each of the following:

Design requirements:

- abstraction, encapsulation and information hiding used where appropriate
- inheritance used if appropriate to the solution
- polymorphism used if appropriate to the solution
- all class-wide variables are private to prevent content coupling
- class-wide variables are kept to a minimum to ensure a minimum of common coupling
- data coupling is used (parameter passing) in preference to content or common coupling
- program does not contain a lot of unnecessary data coupling
- classes are highly cohesive

Implementation requirements:

- a working solution which meets the requirements of the given brief
- variables are correctly declared and initialised
- arithmetic and/or logical operators are used correctly
- a range of control structures are implemented correctly
- at least two data structures are implemented correctly
- the program contains a minimum of four classes, which contain attributes, methods and a constructor method
- a minimum of three objects are created from the classes, with appropriate initial attribute values set through the constructor methods
- the program contains at least one overloaded method (this may be a constructor method)
- classes are linked appropriately through association, aggregation or inheritance relationships
- parameters are passed correctly both within and between objects
- appropriate access types are defined for methods, attributes and classes
- use of pre-defined classes and/or methods from the standard object library
- the program appropriately handles errors with exceptions or pre-validation
- the program code is commented appropriately throughout

Outcome 3

Stage 2 — Testing

After completing Stage 1 you are required to develop a test plan and test the completed program. The test log should identify any areas where the program fails, and detail any fixes and retests required.

Assessment checklist

Unit assessment: candidate's assessment record

HP2L 48 — Software Development: Object Oriented Programming

Class | Candidate's name

Group | Candidate's ID

Outcome 1

Assessment task 1

Record of Performance

Evidence Requirements

Candidates will need to provide evidence to demonstrate their Knowledge and/or Skills by showing that they can investigate object oriented programming techniques and apply them appropriately to a design.

- Object oriented concepts and terms.
- Object oriented programming techniques.
- Objects and classes.
- Attributes and methods.
- Parameter passing.
- Abstraction, encapsulation and information hiding.
- Inheritance.
- Polymorphism.
- Association.
- Aggregation and collection.
- Coupling and cohesion.

Satisfactory/Unsatisfactory | Comments

The comment column can be used to highlight any re-assessment that may be needed.

Overall comments

Assessor's signature _____ **Date** _____

Assessment checklist

Unit assessment: candidate's assessment record

HP2L 48 — Software Development: Object Oriented Programming

Class | Candidate's name

Group | Candidate's ID

Outcome 2

Assessment task 1

Record of Performance

Evidence Requirements

Candidates will need to provide evidence to demonstrate their Knowledge and/or Skills by showing that they can investigate object oriented programming techniques and apply them appropriately to a design.

- Declaring and initialising variables.
- Using operators.
- Implementing control structures.
- Defining data structures.
- Accessing and manipulating data structures.
- Using parameter passing.
- Creating classes.
- Creating instances of classes.
- Creating relationships between classes.
- Creating constructor methods.
- Use of exceptions.
- Use of standard object libraries.
- Documenting code.

Satisfactory/Unsatisfactory | Comments

The comment column can be used to highlight any re-assessment that may be needed.

Overall comments

Assessor's signature _____ **Date** _____

Assessment checklist

Unit assessment: candidate's assessment record

HP2L 48 — Software Development: Object Oriented Programming

Class | Candidate's name

Group | Candidate's ID

Outcome 3

Assessment task 1

Record of Performance

Evidence Requirements

Candidates will need to provide evidence to demonstrate their Knowledge and/or Skills by showing that they can investigate object oriented programming techniques and apply them appropriately to a design.

- Implementing a test plan using a defined strategy.
- Maintaining test documentation.
- Evaluating results of test runs.
- Amending code as necessary.

Satisfactory/Unsatisfactory	Comments

The comment column can be used to highlight any re-assessment that may be needed.

Overall comments

Assessor's signature _____ **Date** _____