Advanced Higher Coursework Assessment Task

Advanced Higher Computing Science

Project

Assessment task

This document provides information for teachers and lecturers about the coursework component of this course in terms of the skills, knowledge and understanding that are assessed. It **must** be read in conjunction with the course specification.

Valid from session 2019-20 and until further notice.

This edition: June 2019 (version 1.0)

Contents

Introduction

This document contains instructions for teachers and lecturers, marking instructions and instructions for candidates for the Advanced Higher Computing Science project. You must read it in conjunction with the course specification.

This project has 80 marks out of a total of 160 marks available for the course assessment.

This is one of two course assessment components. The other component is a question paper.

Instructions for teachers and lecturers

Time

There is no time limit for the project. Candidates should start at an appropriate point in the course.

Supervision, control and authentication

The project is conducted under some supervision and control.

Candidates can complete part of the work outwith the learning and teaching setting, therefore you must exercise professional responsibility to ensure that evidence submitted by a candidate is their own work.

You should put in place ways to authenticate candidate evidence, for example:

- regular checkpoint or progress meetings with candidates
- checklists which record activity and progress

Group work approaches can be helpful to simulate real-life situations, share tasks and promote team-working skills; however, you can only use these to prepare candidates for assessment. Group work is not allowed once formal work on assessment has started.

Resources

This is an open-book assessment. Candidates can access any appropriate resources.

Candidates are required to design and code their solution, and should be aware that extensive use of resources, such as pre-written module libraries, may not allow them to demonstrate these skills and access all marks available.

Reasonable assistance

Candidates must carry out the assessment independently. However, you can provide reasonable assistance prior to, and during, the formal assessment process.

The term ‘reasonable assistance’ is used to balance the need for support with the need to avoid giving too much help. If candidates need more than what is thought to be ‘reasonable assistance’, they may not be ready for assessment.

Reasonable assistance must be limited to constructive comment and/or questioning. You must not adopt a directive role or provide specific advice on how to rephrase, improve responses or provide model answers. Helping candidates on a one-to-one basis in the context of something they have already produced, could become support for assessment and would be going beyond reasonable assistance. For example, you should not prompt candidates to revisit their initial analysis and design as the project develops, and before submitting the final evidence to SQA.

You can give advice on a generic basis, such as how to produce a project plan or how to collate evidence. Where this happens, you should give it to the whole class.

You should advise candidates on their choice of problem, to ensure that their chosen problem meets the criteria for the Advanced Higher project and is achievable. The purpose of the project is to assess the practical skills of the Advanced Higher course. If a project does not meet the criteria, it will not allow the candidate to demonstrate these skills and access all the marks available.

Candidates must work independently once the formal assessment process has started, with input limited to constructive comment and/or questioning.

You can support candidates with the following aspects of their projects:

- printing, collating and labelling their evidence, to ensure it is in the format specified by SQA
- ensuring they have all the materials and equipment they need to complete their project
- ensuring they understand the conditions of assessment, and any administrative arrangements around submitting and storing evidence
- technical support

Once projects are completed and submitted, they must not be returned to candidates for further work.

Research

A small number of marks for research are included in the implementation stage. As candidates implement their solution, they are expected to implement some code that goes beyond the requirements of the Advanced Higher course.

Note: candidates are not expected to spend a significant amount of time on this part of their project; however, you should not discourage them from carrying out in-depth research.

Evidence

All candidate evidence (whether created manually or electronically) must be submitted to SQA in paper-based format.

The evidence checklist details all evidence to be gathered. You should encourage candidates to use it to ensure they submit all evidence to SQA.

You should advise candidates that evidence, especially code, must be clear and legible. This is particularly important when pasting screenshots into a document.

There is no need for evidence to be printed single sided or in colour.

Marking instructions

In line with SQA's normal practice, the following marking instructions for the Advanced Higher Computing Science project are addressed to the marker. They will also be helpful for those preparing candidates for course assessment.

Candidates' evidence is submitted to SQA for external marking.

General marking principles

Always apply these general principles. Use them in conjunction with the detailed marking instructions, which identify the key features required in candidates' responses.

a Always use positive marking. This means candidates accumulate marks for the demonstration of relevant skills, knowledge and understanding; marks are not deducted for errors or omissions.

b If a candidate response does not seem to be covered by either the principles or detailed instructions, and you are uncertain how to assess it, you must seek guidance from your team leader.

c Assess 'completeness' of evidence according to each project. Complete evidence:
- meets all requirements
- relates to the problem
- meets the quality and technical accuracy of Advanced Higher

d Award 0 marks where evidence is:
- not provided
- not related to the problem
- not appropriate to Advanced Higher

e Where bands refer to minor, significant, or major errors and/or omissions, these terms do not indicate the volume required, but the importance of the errors and/or omissions in the context of the project.

f Select the band that most closely describes the evidence provided. Where a range of marks is available for a band, you should determine:
- if the evidence is a closer match to the band above, and if so, award the highest available mark from the range
- if the evidence is a closer match to the band below, and if so, award the lowest available mark from the range

Detailed marking instructions

Analysis of the problem (10 marks)			
Evidence requirements	**Marks**	**Marking instructions**	
Description of the problem, including: ♦ an outline of the problem, identifying Advanced Higher concepts and integration ♦ the scope, boundaries and any constraints	**2**	**2 marks**	Complete and detailed description of the problem that meets all of the evidence requirements, including integration.
		1 mark	Description of the problem, that meets some of the evidence requirements.
UML use case diagram, showing integration, and defining: ♦ actors ♦ use cases ♦ relationships	**2**	**2 marks**	Complete, detailed and integrated use case diagram that meets all of the evidence requirements, including integration.
		1 mark	Use case diagram that meets some of the evidence requirements.
Requirements specification, including: ♦ end-user requirements ♦ functional requirements	**4**	**3-4 marks**	Complete or almost complete and detailed requirements specification that meets all end-user and functional requirements for a fully-working, integrated solution.
		1-2 marks	Requirements specification, with some missing information for a fully-working, integrated solution.
Project plan for each stage, including: ♦ identified tasks ♦ resources required ♦ estimate of timings	**2**	**2 marks**	Complete and detailed project plan that meets all of the evidence requirements.
		1 mark	Project plan that meets some of the evidence requirements.

Design of the solution (20 marks)			
Evidence requirements	**Marks**	**Marking instructions**	
Design of Advanced Higher concepts	**6**	**5–6 marks**	Complete or almost complete and detailed design.
		3–4 marks	Partially complete and detailed design, with some errors and/or omissions.
		1–2 marks	Incomplete design, with a number of significant errors and/or omissions.
Design of integration	**4**	**3–4 marks**	Complete or almost complete and detailed design showing integration.
		1–2 marks	Partially complete design.
Overall design matches the requirements specification	**5**	**4–5 marks**	Design matches all or almost all of the requirements specification.
		2–3 marks	Design matches some of the requirements specification.
		1 mark	Minimal match to the requirements specification.
User-interface design shows inputs, processes and outputs, and matches the end-user and functional requirements	**5**	**4–5 marks**	Complete or almost complete and detailed user-interface design, showing validated inputs and outputs, matching the end-user and functional requirements, and indicating the underlying processes.
		2–3 marks	Partially complete user-interface design, with some errors and/or omissions.
		1 mark	Minimal user-interface design.

Implementation (30 marks)		
Evidence requirements	**Marks**	**Marking instructions**
Implemented Advanced Higher concepts, that match the design	**12**	**11-12 marks** Complete or almost complete and fully-working implementation that matches the design. **9-10 marks** Partially complete and working implementation that closely matches the design, but has some minor errors and/or omissions. **7-8 marks** Partially complete and working implementation that matches the design, with some significant errors and/or omissions. **5-6 marks** Partially complete implementation that matches some aspects of the design, and has a number of significant errors and/or omissions **3-4 marks** Incomplete implementation, with limited match to the design due to major errors and/or omissions. **1-2 marks** Minimal implementation that does not match the design.
Implemented integration, that matches the design	**6**	**5-6 marks** Complete or almost complete and fully-working integration that matches the design. **3-4 marks** Partially complete and working integration that matches some aspects of the design, but with some significant errors and/or omissions. **1-2 marks** Incomplete implementation, with limited match to the design and a number of significant errors and/or omissions.

Implementation (30 marks) continued			
Evidence requirements	**Marks**	**Marking instructions**	
Implemented user interface, that matches the design	**3**	**3 marks**	Complete or almost complete and fully-working user interface that matches the design.
		2 marks	Partially complete and working user interface that matches some aspects of the design, but with some significant errors and/or omissions.
		1 mark	Incomplete user interface, with limited match to the design and a number of significant errors and/or omissions.
Description of new skills and/or knowledge researched and developed	**4**	**3-4 marks**	Complete, or almost complete, and detailed description of research and application of new skills and/or knowledge, that goes beyond what is required for the Advanced Higher course, developed during the implementation stage.
		1-2 marks	Partially complete description of research and application of new skills and/or knowledge developed during the implementation stage.
Log of ongoing testing, including: ♦ a description of issues resolved ♦ references used to resolve these issues	**5**	**4-5 marks**	Complete, or almost complete, and detailed log of ongoing testing, describing issues resolved, and evidencing solutions and references throughout the implementation stage.
		2-3 marks	Partially complete log of ongoing testing.
		1 mark	Incomplete log of ongoing testing.

Testing the solution (15 marks)			
Evidence requirements	**Marks**	**Marking instructions**	
A comprehensive plan for carrying out final testing of the fully implemented solution, including: ♦ all requirements ♦ description of tests ♦ persona and test cases	**6**	**5-6 marks** **3-4 marks** **1-2 marks**	Complete and detailed test plan that meets all evidence requirements. Partially complete test plan that meets some evidence requirements. Incomplete test plan that meets minimal evidence requirements.
Evidence of requirements testing	**6**	**5-6 marks** **3-4 marks** **1-2 marks**	Complete evidence of requirements testing that matches the test plan. Partially complete evidence of requirements testing. Incomplete evidence of requirements testing.
Description of the results of the test cases	**3**	**3 marks** **2 marks** **1 mark**	Complete and detailed description of the results of the test cases that matches the test plan. Partially complete description of the results of the test cases. Incomplete description of the results of the test cases.

Evaluation of the solution (5 marks)			
Evidence requirements	**Marks**	**Marking instructions**	
Evaluation of the solution in terms of fitness for purpose, by discussing: ♦ how closely the solution matches the requirements specification ♦ the results of testing	**3**	**3 marks**	Complete and detailed evaluation of the solution's fitness for purpose that meets all of the evidence requirements.
		2 marks	Partially complete evaluation of the solution that meets some of the evidence requirements.
		1 mark	Incomplete evaluation of the solution that meets minimal evidence requirements.
Evaluation of the solution in terms of: ♦ maintainability ♦ robustness	**2**	**2 marks**	Complete and detailed evaluation of the solution's maintainability and robustness.
		1 mark	Partially complete evaluation of the solution's maintainability and robustness.

Instructions for candidates

This assessment applies to the project for Advanced Higher Computing Science.

This project has 80 marks out of a total of 160 marks available for the course assessment.

It assesses the following skills, knowledge and understanding:

- applying computational thinking to solve a complex computing problem
- analysing a complex problem within a computing science context
- designing, developing, implementing, testing, and evaluating a digital solution to a complex problem
- demonstrating advanced skills in computer programming
- communicating understanding of complex concepts related to computing science, clearly and concisely, using appropriate terminology

Your teacher or lecturer will let you know if there are any specific conditions for doing this assessment.

For this project, you have to identify a computing science problem, agreed with your teacher or lecturer. You need to develop a solution to the problem, from analysis through to evaluation. You gain marks for the following stages of the project:

- analysis of the problem (10 marks)
- design of the solution (20 marks)
- implementation (30 marks)
- testing the solution (15 marks)
- evaluation of the solution (5 marks)

In this document, there is guidance on:

- how much support and assistance your teacher or lecturer can give you
- what evidence you need to collect
- choosing a suitable problem for your project
- what you need to do at each stage of the project

Support and guidance from your teacher or lecturer

You must complete this project independently; however, your teacher or lecturer can provide you with guidance to help develop your thinking as you progress. This could be:

- general support in class on broad areas, such as project planning
- constructive questioning with you on an individual basis
- constructive comments to help you find a solution

Your teacher or lecturer cannot tell you specifically how to proceed with your project, how to rephrase or improve responses, or provide you with model answers.

Evidence to be gathered

You need to gather evidence for each stage of the project. Evidence can include program listings, screenshots, web page source files, data files or similar, as appropriate. Your evidence must be printed and submitted to SQA for marking.

You should ensure that you:

- include all your evidence, by completing the checklist provided
- clearly label your evidence
- submit your evidence in a logical order, with appropriate headings for each stage

You will probably work on your project for several months, and during that time, you will produce many types of evidence. You do not need to provide evidence of your progress, but you may find it useful to keep track of your progress and organise the evidence you produce. This could be in the form of a diary.

You should store your evidence methodically in a folder, ring binder and/or electronic storage system. Each stage of the project provides more detail on the evidence required. For the design and implementation stages, the evidence will depend on the problem you are solving.

Although there is no page limit or maximum word count for your evidence, marks are awarded for the quality of your work, not the quantity.

Choosing a suitable problem

You must choose a suitable problem for your project. You may already have an idea or you can explore ideas with other candidates and/or your teacher or lecturer. You can also get ideas from online resources, industry news, television, local business partners or STEM ambassadors. A successful project is likely to be about something you are interested in.

There are ideas for your project in this document — you can choose or adapt one of these, or use an idea of your own.

To fully implement your solution, you need to include a small amount of code based on skills and knowledge not covered in the Advanced Higher course. You should research this yourself.

It is possible to complete some projects within your centre, but you could consider a project that requires collaboration with a university, college or local industry. Your teacher or lecturer can advise you about this.

You must discuss your project idea with your teacher or lecturer. This will ensure that it meets the criteria set out below and is achievable within the constraints of time, expertise and resources available.

Project criteria

Your project must:

- be based on **one** of the following areas of the course:
  - software design and development
  - database design and development
  - web design and development
- include at least **two** concepts from this area of the course
- integrate with **one** of the other two areas of the course
- validate all inputs

You can review detailed requirements and examples of suitable projects for each possible combination on the following pages.

Software design and development (SDD) project

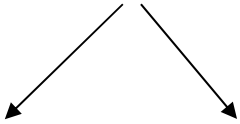
Object-oriented programming with an array of objects	**or**	Procedural programming with a 2D array or an array of records

with

One standard algorithm (using the data structure from above), from the following:

- binary search
- insertion sort
- bubble sort

Your SDD project should integrate with either:



Database design and development Create a database with at least one table **and** open and/or close the database connection to execute an SQL query and (if required), format the results	**or**	**Web design and development** Web user interface to receive input and/or display formatted output

Outline of the problem A computer game requires a variety of regular polygons to appear in the top, left, right and bottom of the screen. Players respond to each shape by pressing keys corresponding to the correct number of sides and the position of the shapes. The reaction time of players is calculated and stored.	**Outline of the problem** The results of a survey are stored in a file. A simple web page, with embedded program code, is required to allow a user to search for data within the file and display the results of the search.

Analyse the problem, create a design and then implement a procedural program that:

- reads previous players' times from a database table, into parallel arrays
- prepares a 2D array storing randomised shapes and co-ordinates
- displays each shape in the 2D array and calculates the total time taken by each player to press the correct keys
- adds each player's time to the parallel arrays, and bubble sorts by time
- displays the top 10 times
- inserts the players' times into a table

Test the program with a variety of sample data.

Evaluate the solution.

Analyse the problem, create a design and then implement object-oriented code that:

- reads the survey data from a file and stores it within an array of objects
- uses a binary search to find user input within the array
- outputs the search results within the web page, formatted using HTML table elements and Inline CSS

Test the program with a variety of survey data and search scenarios.

Evaluate the solution.

Database design and development (DDD) project

Create a database with a minimum of four related tables, using SQL

with

SQL queries (using the tables above), that incorporate any two of the following:

- subquery
- one logical operator (NOT, BETWEEN, ANY, EXISTS)
- query across at least three tables

Your DDD project should integrate with either:

Software design and development

Programming interface to receive input and/or display formatted output

or

Web design and development

Web user interface to receive input and/or display formatted output

Outline of the problem

A relational database is needed to store personal details, meter readings and previous bills of electricity customers.

Analyse the problem, design and implement a suitable database using only SQL statements.

Design and implement additional SQL statements to maintain the database, for example to:

- insert, update and remove customers
- generate bills
- update electricity costs

Outline of the problem

A relational database is needed to store the personal details and health data (for example steps and average heart rate) of members of a gym.

Analyse the problem, design and implement a suitable database using only SQL statements.

Design and implement additional SQL statements to maintain the database, for example to:

- insert, update and remove members
- create statistical output for members

Design and implement a small program to search for a customer, and insert a new meter reading for that customer. Test the solution with sample data. Evaluate the solution.	Design and implement a simple web page to allow members to input their step count and average heart rate. Test the solution with sample data. Evaluate the solution.

Web design and development (WDD) project

Complete website that includes:

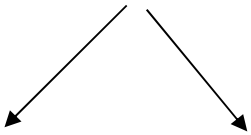
- form elements (action, method, and name)
- external CSS
- multiple layouts using a media query
- use of session variables

with

Server-side processing (PHP) used to:

- assign variables
- process form data

Your WDD project should integrate with either:



Database design and development

Create a database with at least one table

and

open or close the database connection to execute an SQL query and (if required), format the results

or

Software design and development

Embedded programming using one Advanced Higher level data structure or standard algorithm

Outline of the problem

Members of a swimming club need to be able to register for the club's annual swimming competition by completing an HTML form on a website.

Analyse the problem, design and implement a screen width responsive website for the swimming club using a media query.

Outline of the problem

Online revision quizzes are needed to prepare learner drivers for their theory test.

Analyse the problem, design and implement a website for the quizzes that stores the quiz questions and answers, within a 2D array.

Design and implement a single-table database, along with the required PHP code to validate and store the form details. Test the solution with sample data. Evaluate the solution.	When a user submits a quiz answer, a function is called to check the answer and display a result. Design and implement the required code to store and check users' answers. Test the solution with sample quiz data. Evaluate the solution.

Choosing a suitable problem — checklist

Use the following checklist to help you decide on the idea for your project.

a Will the solution to your problem involve implementing two Advanced Higher concepts from one area of the course? **yes** ☐ **no** ☐

These are:

1 _____

2 _____

b Will the solution to your problem involve integration with one of the other two areas of the course? **yes** ☐ **no** ☐

Integrates with _____

c Will the solution to your problem validate all inputs? **yes** ☐ **no** ☐

d Will you be able to complete the project in the time available? **yes** ☐ **no** ☐

e Can you overcome all potential barriers to carrying out your project, for example permissions, logistics, and access to necessary hardware and software? **yes** ☐ **no** ☐

f Does your teacher or lecturer agree with your answers above? **yes** ☐ **no** ☐

Your answers to questions a–c will help you outline your problem in the analysis stage.

If you answer 'no' to any of the above questions, you will need to reconsider your project idea.

Guidance for each stage

Developing a solution

As you work through your project, you must follow these five stages of development:

- analysis
- design
- implementation
- testing
- evaluation

You can follow these stages using an iterative approach or using an agile methodology — where you break the project down into several small iterations of design, implement and test.

Whatever approach you use, each stage of development should continue from the previous stage. For example, you should create your design from the requirements identified at the analysis stage; you should implement a solution from the design you created; and so on.

The following pages detail the requirements for each stage. You gain marks based on the evidence you submit for each of these stages.

If you need to go back and revisit a previous stage (for example to add detail to analysis or improve a design), you should ensure that you submit only the final version as evidence.

Analysis of the problem 10 marks

Before you begin designing and developing a solution, you must analyse the problem that you are going to solve, to ensure that you fully understand every aspect of it.

Description of the problem (2 marks)

Describe your problem. Your description should include:

- an outline of the problem, identifying the Advanced Higher concepts and integration (see the examples earlier in this document)
- the scope and boundaries of the problem, and any constraints you identify

UML use case diagram (2 marks)

Draw a UML use case diagram for your problem. Your diagram should define the following:

- actors
- use cases
- relationships

Your diagram should include the integration you intend to implement, for example a web project connecting to a database.

Requirements specification (4 marks)

Produce a requirements specification. Your requirements specification should list:

- end-user requirements
- functional requirements

Your requirements specification should consider any input validation that may be required for your problem.

Project plan (2 marks)

Create a project plan for the four remaining stages of your project.

Your project plan should include:

- the tasks you complete in each stage
- any resources you need to implement your solution
- an estimate of how long each stage will take

Guidance for producing a project plan

Tasks could be:

- user-interface design
- refinement of coding
- ongoing testing

Resources can include access to development tools and end users. Some of these could be available at any time, while others may only be available at certain times. You need to plan to ensure that your project is not held up waiting for resources.

Your timings should allow for holidays, or other events that affect how much time you can spend on your project.

You should review and update your project plan as you work through each stage. **You must submit your final version of the plan as evidence.**

Design of the solution 20 marks

You must now design your solution based on your requirements specification.

The problem you are solving will determine the design methodologies you use. The following may be suitable:

Software design and development

- code design — top level design with data flow and refinements of Advanced Higher concepts (structure diagrams or pseudocode)
- UML class diagram — including class names, properties and data types, methods (including constructor) and arguments, and (where appropriate), public and/or private, inheritance

Database design and development

- entity relationship diagram — including (where appropriate) entity name, entity type (strong, weak), attributes, relationship participation (mandatory, optional), relationship name and cardinality
- data dictionary — using SQL attribute types
- query designs

Web design and development

- wireframes relating to media query
- low-fidelity prototypes of pages
- code design for PHP (structure diagrams or pseudocode)

Project design (15 marks)

Design your solution, using appropriate design methodologies or techniques.

Your design should meet the end-user and functional requirements identified at the analysis stage, and include the integrated part of the project.

User-interface design (5 marks)

Design the user interface for your solution using appropriate design methodologies or techniques.

Your user-interface design should include wireframes that show all inputs (with notes on validation), underlying processes and outputs. It should also meet the end-user and functional requirements identified at the analysis stage.

Implementation 30 marks

You must now implement your solution and user interface, based on your design.

The problem you are solving will determine the evidence you provide of your implemented solution, including the user interface. The following may be suitable:

Software design and development

- program code
- screenshots of program user interface

Database design and development

- SQL code
- screenshots of implemented tables

Web design and development

- PHP code
- HTML code and page content
- CSS declarations
- screenshots of pages, as viewed in a browser

Implementation (21 marks)

Implement your solution, including the user interface, ensuring it matches your completed design.

As you implement your solution, you will encounter errors or problems that you need to solve before you can continue. Take notes of errors, solutions and any reference materials you use, for example websites, forums, textbooks or learning resources. You will need to refer to these notes to produce evidence of ongoing testing (see below).

Your implementation should include some coding that goes beyond what you are taught during the Advanced Higher course. This will require you to carry out some research. You should keep a note of any references you use and where in your solution you use the skills and knowledge you learn through your research, as you will need to provide a description of this.

Research and development of new skills and/or knowledge (4 marks)

Describe:

- the new skills and/or knowledge that you researched
- how you applied these new skills and/or knowledge to your project

You should reference the resources you used to research and develop these new skills and/or knowledge.

Log of ongoing testing (5 marks)

Produce a log of the ongoing testing you carry out during implementation. Your log should include:

- what you are testing
- descriptions of issues you encounter during testing
- descriptions of how you resolve these issues
- lists of references you use to resolve each issue

You could present your log as a table, using the above bullet points as column headings.

Testing the solution 15 marks

Once you have fully implemented your design, you must carry out final testing on your solution. This testing should be systematic and comprehensive, and based on a test plan.

Final test plan (6 marks)

Create a plan of how you will carry out final testing of your fully implemented solution.

Your plan should be comprehensive, to ensure that your solution meets all the requirements identified at the analysis stage. It should include:

- all requirements
- a description of the tests you will carry out
- a persona and test cases to test the solution with an end user

Note: the tester can be another candidate, a teacher or a lecturer, who adopts the persona and carries out the test cases.

Requirements testing (6 marks)

Test your solution and provide evidence of each end-user and functional requirement test identified in your plan.

The problem you are solving will determine the evidence you require. The following may be suitable:

- screenshots of program inputs and outputs (including errors if any are generated)
- printouts of database output tables generated by SQL statements
- screenshots of form data being entered, along with subsequent results (for example table updated and errors returned)

Testing with persona and test cases (3 marks)

Test your solution using the persona and test cases identified in your plan.

Describe the results of each test case — this could be in the form of a short report or a table.

Evaluation of the solution — 5 marks

You must now evaluate your solution.

Evaluation report (5 marks)

Produce a report to evaluate your solution. This should include:

- the fitness for purpose of your solution, discussing:
  - how closely your solution matches the functional requirements
  - the results of your testing
- the maintainability and robustness of your solution

Candidate checklist

Analysis	Complete
Description of the problem	
UML use case diagram	
Requirements specification	
Project plan	
Design	
Project design	
User-interface design	
Implementation	
Implementation	
Research and development of new skills and/or knowledge	
Log of ongoing testing	
Testing	
Final test plan	
Requirements testing	
Testing with personas and test cases	
Evaluation	
Evaluation report	

Administrative information

Published: June 2019 (version 1.0)

History of changes

Version	Description of change	Date

Note: you are advised to check SQA's website to ensure you are using the most up-to-date version of this document.

Security and confidentiality

This document can be used by SQA approved centres for the assessment of National Courses and not for any other purpose.