H446 Component 3

ALLERTON GRANGE SCHOOL A LEVEL COMPUTER SCIENCE NEA

2024-2025

TABLE OF CONTENTS

Contents

Overview of requirements

INTRODUCTION

The A Level Computer Science Non-Examined Assessment is a substantial piece of work that is worth 20% of the overall qualification. The programming project assesses a variety of different skills including the development and demonstration of computational thought processes. Beginning at the end of Year 12 and concluding near the end of Year 13, a significant number of months are to used to work on this valuable independent project.

Many students who have completed the NEA have commented that undertaking the project greatly improved their project management and software development skills. It is hoped that you find the experience just as useful and enjoy it as a positive experience that enables you to gain important UMS points.

KEY FEATURES OF THE COURSEWORK

Your project will consist of the following:

- Report
- Source code files
- Video files (evidencing development and testing)

You are expected to analyse, design, develop, test, evaluate and document a program written in a suitable programming language. In order to achieve this, you will use an agile development methodology. This means that once the problem has been identified and success criteria devised, you will work in multiple cycles of designing, implementing and testing different components of your problem. Using an iterative approach when working on each component, you must document these cycles to show the first, second and any additional revisions that have taken when working on a part of your solution – evidencing the improvements and refinements that have taken place throughout.

Your report will look like a development diary with lots of commentary and screenshots of what has taken place. This can be further supported by video files that show your working solution.

Your teacher and supporting materials will provide clear guidance on what sections to include in your report.

Choosing your topic

One of the considerations when embarking on a programming project with is ensuring you choose a project with sufficient complexity to be of A Level standard; this is not easy to define, but it must be more complex than typical GCSE work.

As a minimum, it must include:

- a graphical user interface
- complex algorithms

Consider the complexity of the algorithms you study on the A Level Computer Science course as a benchmark. These include linear search, binary search, bubble sort, insertion sort, merge sort, quick sort, Dijkstra and A* algorithm. You are unlikely to need many of these, but they give you an idea of the level of complexity required.

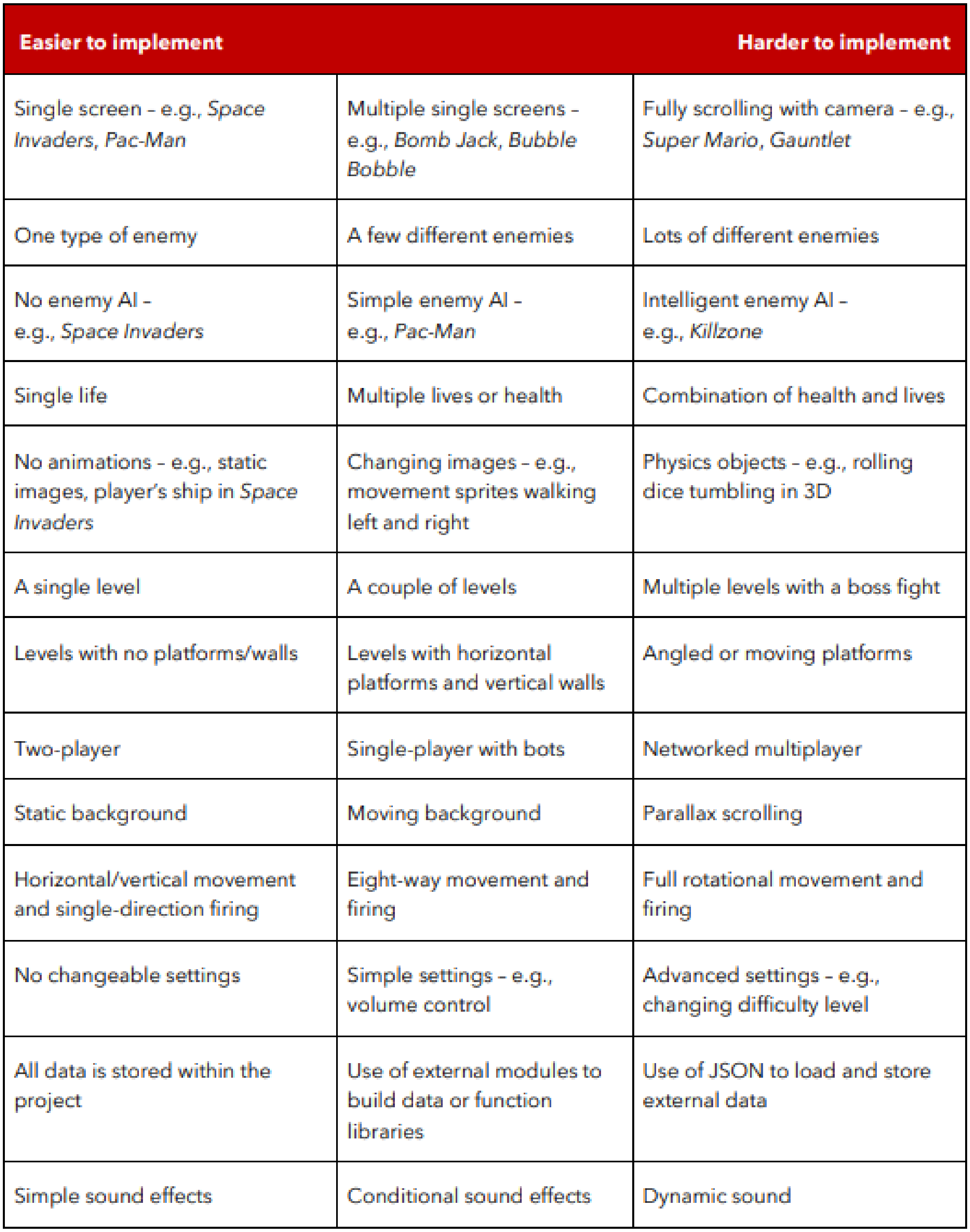
Examples of what could determine whether a game is complex enough:

(i) a simple game of battleships where the computer opponent chooses a random square to fire is too simplistic. An algorithm that decides the next position to fire on after a hit is scored and is dependent on the shape of ships already shot down has that added layer of complexity that would elevate the project to A Level standard.

(ii) A game of Pong with two paddles and a ball is too simplistic because many game engines will do this for you – leaving you with little code to write. However, extending this to a breakout clone with bricks and power-ups adds the required complexity. Extending this further into a four-player version where players defend their castle in the four corners of the screen would be considered A Level standard.

On the other hand, you want to avoid your project becoming too complex so that the undertaking of it becomes overwhelming. Remember that AAA games are made by large teams of experienced programmers and take many years. You are on your own and have a limited amount of time. **Often it is possible to come up with a simple idea that takes your interest and then apply a series of extensions to make a sufficiently complex solution.**

The next page shows some ideas to help you to determine the scope of your project.



Easier to implement		Harder to implement
Single screen – e.g., *Space Invaders*, *Pac-Man*	Multiple single screens – e.g., *Bomb Jack*, *Bubble Bobble*	Fully scrolling with camera – e.g., *Super Mario*, *Gauntlet*
One type of enemy	A few different enemies	Lots of different enemies
No enemy AI – e.g., *Space Invaders*	Simple enemy AI – e.g., *Pac-Man*	Intelligent enemy AI – e.g., *Killzone*
Single life	Multiple lives or health	Combination of health and lives
No animations – e.g., static images, player's ship in *Space Invaders*	Changing images – e.g., movement sprites walking left and right	Physics objects – e.g., rolling dice tumbling in 3D
A single level	A couple of levels	Multiple levels with a boss fight
Levels with no platforms/walls	Levels with horizontal platforms and vertical walls	Angled or moving platforms
Two-player	Single-player with bots	Networked multiplayer
Static background	Moving background	Parallax scrolling
Horizontal/vertical movement and single-direction firing	Eight-way movement and firing	Full rotational movement and firing
No changeable settings	Simple settings – e.g., volume control	Advanced settings – e.g., changing difficulty level
All data is stored within the project	Use of external modules to build data or function libraries	Use of JSON to load and store external data
Simple sound effects	Conditional sound effects	Dynamic sound

PROPOSAL DOCUMENT

Before you break-up for summer at the end of year 12, you must submit an A Level Computer Science NEA Proposal Form. This will allow you to share your initial ideas with your teacher to have it approved. It must include the following:

- A title
- A brief overview of your idea.
- Identify your client (end-user)
- An outline of what you want to achieve as an outcome.
- Initial ideas for how it will be complex enough for Level

SUPPORT RESOURCES

See the suggested list of resources below:

- Allerton Grange School A Level Computer Science SharePoint
- OCR A Level Comp Sci NEA mark scheme
- Documenting Defold Programming Projects E-Book (Craig N Dave)
- Documenting programming Non-Games Projects E-Book (Craig N Dave)
- Tackling A Level Projects in Computer Science Physical Book (PG Online)
- Defold Game Tutorials (Craig N Dave)
- Creating Video Games Using PyGame Physical Book (Mike Gold)
- https://www.pygame.org/wiki/tutorials
- YouTube tutorials for Pygame

Guidance

The Joint Council for Qualifications sets rigorous expectations on the completion of internally assessed work and clear guidelines for the role of the teachers assisting you.

WHAT GUIDANCE IS YOUR TEACHER ALLOWED TO OFFER?

- Support you in choosing an appropriate project without giving you a specific idea. For example, your teacher could advise and help you choose from a list of your own ideas.
- Provide you with the mark scheme.
- Review your work and provide oral and written feedback at a general level.
- Allow you to redraft your work in response to this feedback.

WHAT GUIDANCE IS YOUR TEACHER NOT ALLOWED TO OFFER?

- Provide you with writing frames or paragraph/section headings.
- Provide detailed, specific advice on how to improve drafts to meet the assessment criteria.
- Give detailed feedback on errors and omissions.
- Intervene to improve the presentation or content of your work.
- Provisionally mark your work and then allow you to revise it.

Malpractice

If a student commits malpractice it means that, they have failed to follow the rules of an examination or assessment. ‘Candidate malpractice’ means malpractice by a candidate in connection with any examination or assessment, including the preparation and authentication of any controlled assessments, coursework or non-examination assessments, the presentation of any practical work, the compilation of portfolios of assessment evidence and the writing of any examination paper.

Malpractice includes:

Plagiarism – copying another student’s work or copying from any other source e.g. books, articles, websites or AI. This is cheating and is malpractice.

Collusion – sharing your work with another student to benefit the competition of the coursework. Sharing your work for someone else to copy is still cheating and is deemed as malpractice. The consequence could mean that your coursework is jeopardised.

Students who are suspected of plagiarism or collusion will be investigated and this shall be sent off to the examination board by the examination officer. The examination board, will notify the Head teacher of their final decision. If the examination board finds the student guilty of malpractice because they have colluded, the student may not be given any marks for their coursework.

The AGS malpractice policy, which has been drawn from the JCQ (Joint Council for Qualifications) can be found here:

Malpractice Policy (Exams) (2).pdf

AI AND ASSESSMENTS

AI stands for artificial intelligence and using it is like having a computer that thinks.

AI tools like ChatGPT or Snapchat, My AI can write text, make art and create music by learning from data from the internet.

Using AI to create your coursework and say it is your own work is cheating and is considered as ‘malpractice.’ Candidate’s work which is suspected of using AI, will undergo an investigation through the examination board. This could result in the candidate’s coursework being invalid and will not count towards their final GCSE or A-Level coursework.

AGS A LEVEL COMPUTER SCIENCE NEA

Do not use any AI to complete your coursework. It is cheating and could jepordise your grade.

The exam boards use sophisticated ‘plagarism’ software called Turnit In to identify any cheating. As coursework is submitted to the examination board, any plagiarism will be identified and consequences will occur.

For the full AI policy created by JCQ (Joint Council for Qualifications) please following the link below:

JCQ guidance - AI-Use-in-Assessments_Feb24_v3.pdf

SUBJECT SPECIFIC GUIDANCE

For A Level Computer Science, it is possible to use AI generated art for your sprites and other assets. However, you must fully reference what you have searched for and asked to be created as outlined in the JCQ guidance. In addition, you may use libraries that have been created by third-parties, but these must be fully referenced in your report and no credit will be given for the algorithms that they include.

Referencing guide

It is essential that you credit authors for their research material and ideas otherwise you could be accused of plagiarism. It is a very serious academic offence to pretend that someone else's work has been created by you. This applies even if you copy just a few sentences. Learning to cite references correctly will help to ensure that you do not commit plagiarism by accident.

BASIC REFERENCING TERMS

- **Reference -** details of any item (e.g. book, chapter, video, web page, article) used as a source which enables that source to be found by someone else.
- **Bibliography -** a list of references at the end of a document e.g. essay, thesis, journal article.
- **Citation -** brief details about a reference given in the text of a document e.g. (author: date)

REFERENCING A BOOK

First name, Surname, *Full Title of the Book* (Place of Publication, Date of Publication) p. [to signify the page you got the information from or pp. to signify the pages you got the information from].

Example:
Matthew Taylor, The Association Game: A History of British Football (Harlow, 2008), pp. 26-29.
For a book, the second and subsequent times you mention it: here, we use a shortened form to save words: Surname, First Part of Title, page number(s). Do not use ibid. or op. cit. They really do not help the reader at all, and they do not show off the breadth of your reading either.

REFERENCING A WEBSITE

Website author (Year published / last updated) *Title of the Internet site.* Available at: URL (Accessed Day Month Year)

Example:

International tourism partnership (2004) *International tourism partnership*. Available at: http://www.internationaltourismpartnership.org/ (Accessed 8 February 2009).

If the website author is not available, simply use the website name e.g. Website name, Year.

If the date of the website is not available, substitute the date with the words 'no date' e.g. Website author (no date).

REFERENCING A YOUTUBE VIDEO

Name of person posting video (Year video posted) Title of film or programme. Available at: URL (Accessed Day Month Year).

Example:

APintTurtle (2008) Zig & Zag - Christmas crises. Available at: http://youtu.be/yCv4iyPqZKQ (Accessed 12 December 2014).

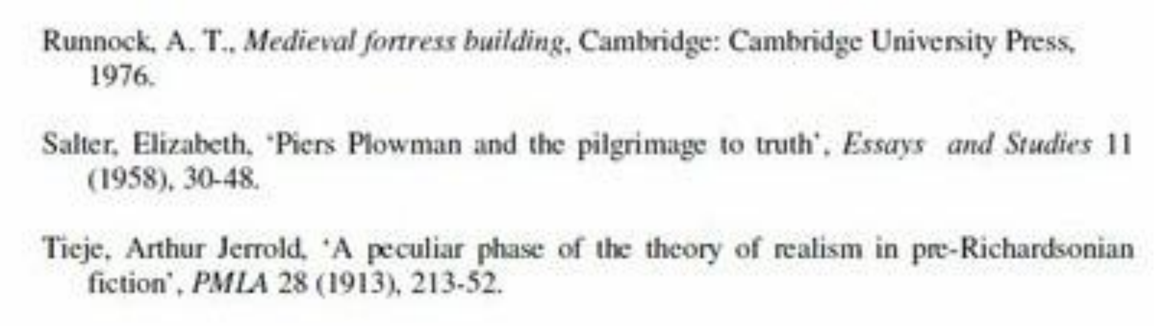
BIBLIOGRAPHY

A bibliography should generally contain all the sources cited in the text and notes and any other important titles that you have consulted or used in preparing the submission.

The form of entries in the bibliography is similar to that for the full reference, except that the author's surname is placed before the initials of the first name.

The bibliography does not give references to specific page numbers where information can be found.

1. A bibliography should have what is called a 'hanging indent', that is, the first line is flush with the left margin, but subsequent lines are indented three or four spaces. (This is as shown in the examples below.)
2. Examples for books and articles in bibliography.



Runnock, A. T., *Medieval fortress building*, Cambridge: Cambridge University Press, 1976.

Salter, Elizabeth, 'Piers Plowman and the pilgrimage to truth', *Essays and Studies* 11 (1958), 30-48.

Tieje, Arthur Jerrold, 'A peculiar phase of the theory of realism in pre-Richardsonian fiction', *PMLA* 28 (1913), 213-52.

Presentation of your coursework

Your report must have the following features:

- A front cover page that includes: your name, your candidate number, centre name, centre number, project title, A Level Computer Science, date.
- An accurate table of contents page.
- A header that includes your full name, candidate number and centre number.
- A footer that includes page numbers.
- Headings and sub-headings that stand out by using different formatting to the main body of text
- The main body of text in pitch size 12 and black.
- Screenshots that are large enough to see and have a clear resolution.
- Screenshots that are cropped in an appropriate way.
- Diagrams that use the appropriate symbols.
- Referencing where relevant (using the guidance in this booklet)
- Bibliography at the end of the report (using the guidance in this booklet)
- Written work should be free of grammatical, spelling and punctuation errors. It is highly recommended that at least two people read your coursework as it is easy to overlook mistakes when you have been so heavily involved in writing it.

Your coded solution must have the following features:

- Use of commenting at the start of any source code to state the file name and a brief summary of the purpose of the code.
- Sensible identifiers throughout
- Commenting to show the dividing and purpose of different sections
- Commenting to show the purpose of certain lines where relevant
- All assets appropriately named
- All files collated in a zip folder

You will complete your work incrementally, so it is wise to save versions of your work at regular intervals. This way, you can revert to a previous version if you encounter a critical error that can't be fixed. Additionally, this approach allows you to develop two potential solutions and determine which one works best. Therefore, having good version control is crucial when creating both your report and coded solution.

Completion of Coursework

DEADLINES

The following are the interim deadlines for your NEA:

Analysis: Monday 30th September 2024

Design and Development: Monday 17th February 2025

Testing After Development: Monday 10th March 2025

Evaluation: Monday 31st March 2025

The following is the final deadline for your NEA:

Final Deadline: Friday 4th March 2025

SUBMISSION OF COURSEWORK

For interim deadlines your work must be stored on the Files section of your private channel on Microsoft Teams.

For the final deadline your work must be uploaded the assignment that is set on Microsoft Teams.

MODERATION

Internal moderation of your coursework takes place amongst teachers of the Computing department at Allerton Grange. Your ‘provisional’ for the NEA project will then be shared. This is subject to approval by OCR during external moderation for which you will find out the outcome on results day in August.

APPENDICES

Appendix 1: Mark Scheme

Programming project (Component 03 or 04) marking criteria – 70 marks

AO 2.2 Analysis (maximum 10 marks)			
1–2 marks	**3–5 marks**	**6–8 marks**	**9–10 marks**
The candidate will have:			
• Identified some features that make the problem solvable by computational methods. • Identified suitable stakeholders for the project and described them and some of their requirements. • Identified some appropriate features to incorporate into their solution. • Identified some features of the proposed computational solution. • Identified some limitations of the proposed solution. • Identified some requirements for the solution. • Identified some success criteria for the proposed solution.	• Described the features that make the problem solvable by computational methods. • Identified suitable stakeholders for the project and described how they will make use of the proposed solution. • Researched the problem looking at existing solutions to similar problems identifying some appropriate features to incorporate into their solution. • Identified the essential features of the proposed computational solution. • Identified and described some limitations of the proposed solution. • Identified most requirements for the solution. • Identified some measurable success criteria for the proposed solution.	• Described the features that make the problem solvable by computational methods and why it is amenable to a computational approach. • Identified suitable stakeholders for the project and described them and how they will make use of the proposed solution and why it is appropriate to their needs. • Researched the problem in depth looking at existing solutions to similar problems identifying and describing suitable approaches based on this research. • Identified and described the essential features of the proposed computational solution. • Identified and explained any limitations of the proposed solution. • Specified the requirements for the solution including (as appropriate) any hardware and software requirements. • Identified measurable success criteria for the proposed solution.	• Described and justified the features that make the problem solvable by computational methods, explaining why it is amenable to a computational approach. • Identified suitable stakeholders for the project and described them explaining how they will make use of the proposed solution and why it is appropriate to their needs. • Researched the problem in depth looking at existing solutions to similar problems, identifying and justifying suitable approaches based on this research. • Identified the essential features of the proposed computational solution explaining these choices. • Identified and explained with justification any limitations of the proposed solution. • Specified and justified the requirements for the solution including (as appropriate) any hardware and software requirements. • Identified and justified measurable success criteria for the proposed solution.

AO 3.1 Design (maximum 15 marks)			
1–4 marks	**5–8 marks**	**9–12 marks**	**13–15 marks**
The candidate will have:			
• Described elements of the solution using algorithms. • Described some usability features to be included in the solution. • Identified the key variables / data structures / classes (as appropriate to the proposed solution). • Identified some test data to be used during the iterative or post development phase of the process. •	• Broken the problem down systematically into a series of smaller problems suitable for computational solutions describing the process. • Defined the structure of the solution to be developed. • Described the solution fully using appropriate and accurate algorithms. • Described the usability features to be included in the solution. • Identified the key variables / data structures / classes (as appropriate to the proposed solution) and any necessary validation. • Identified the test data to be used during the iterative development of the solution. • Identified any further data to be used in the post development phase.	• Broken the problem down systematically into a series of smaller problems suitable for computational solutions explaining the process. • Defined in detail the structure of the solution to be developed. • Described the solution fully using appropriate and accurate algorithms explaining how these algorithms form a complete solution to the problem. • Described, explaining choices made, the usability features to be included in the solution. • Identified and justified the key variables / data structures / classes (as appropriate to the proposed solution) explaining any necessary validation. • Identified and justified the test data to be used during the iterative development of the solution. • Identified and justified any further data to be used in the post development phase.	• Broken the problem down systematically into a series of smaller problems suitable for computational solutions, explaining and justifying the process. • Defined in detail the structure of the solution to be developed. • Described the solution fully using appropriate and accurate algorithms justifying how these algorithms form a complete solution to the problem. • Described, justifying choices made, the usability features to be included in the solution. • Identified and justified the key variables / data structures / classes (as appropriate to the proposed solution) justifying and explaining any necessary validation. • Identified and justified the test data to be used during the iterative development of the solution. • Identified and justified any further data to be used in the post development phase.

0 marks = no response or no response worthy of credit.

AO 3.2 Developing the coded solution (maximum 25 marks)			
Iterative development of a coded solution (maximum 15 marks)			
1–4 marks	**5–8 marks**	**9–12 marks**	**13–15 marks**
The candidate will have:			
• Provided evidence of some iterative development for a coded solution. • Solution may be linear. • Code may be inefficient. • Code may not be annotated appropriately. • Variable names may be inappropriate. • There will be little or no evidence of validation. • There will be little evidence of review during the development.	• Provided evidence for most stages of the iterative development process for a coded solution describing what they did at each stage. • Solution will have some structure. • Code will be briefly annotated to explain key components. • Some variable and/or structure names will be largely appropriate. • There will be evidence of some basic validation. • There will be evidence that the development was reviewed at some stage during the process.	• Provided evidence of each stage of the iterative development process for a coded solution relating this to the break down of the problem from the analysis stage and explaining what they did at each stage. • Provided evidence of some prototype versions of their solution. • The solution will be modular in nature. • Code will be annotated to explain all key components. • Most variables and structures will be appropriately named. • There will be evidence of validation for most key elements of the solution. • The development will show review at most key stages in the process.	• Provided evidence of each stage of the iterative development process for a coded solution relating this to the break down of the problem from the analysis stage and explaining what they did and justifying why. • Provided evidence of prototype versions of their solution for each stage of the process. • The solution will be well structured and modular in nature. • Code will be annotated to aid future maintenance of the system. • All variables and structures will be appropriately named. • There will be evidence of validation for all key elements of the solution. • The development will show review at all key stages in the process.
Testing to inform development (maximum 10 marks)			
1–2 marks	**3–5 marks**	**6–8 marks**	**9–10 marks**
The candidate will have:			
• Provided some evidence of testing during the iterative development process.	• Provided some evidence of testing during the iterative development process. • Provided evidence of some failed tests and the remedial actions taken.	• Provided evidence of testing at most stages of the iterative development process. • Provided evidence of some failed tests and the remedial actions taken with some explanation of the actions taken.	• Provided evidence of testing at each stage of the iterative development process. • Provided evidence of any failed tests and the remedial actions taken with full justification for any actions taken.

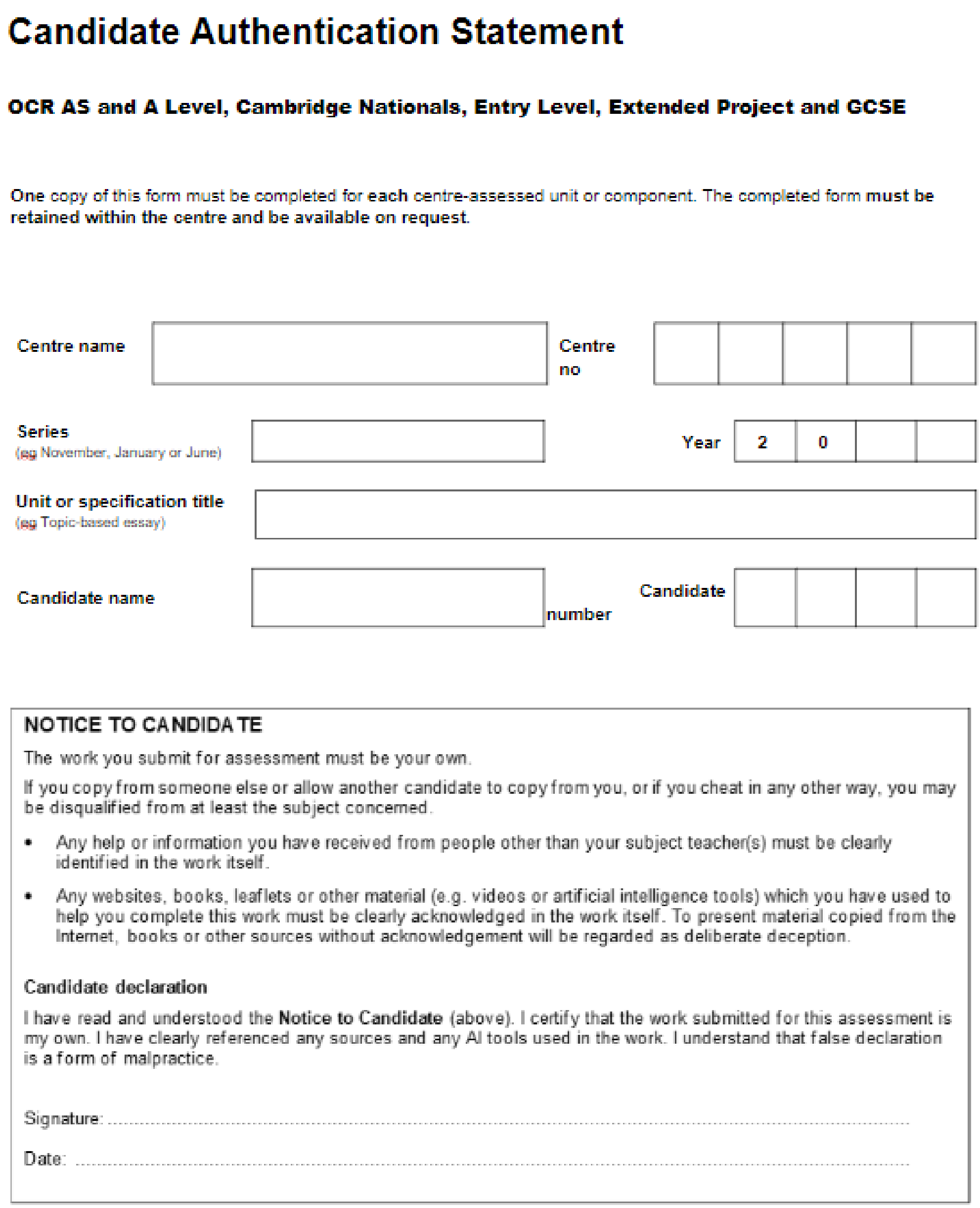
AO 3.3 Evaluation (maximum 20 marks)			
Testing to inform evaluation (maximum 5 marks)			
1 mark	**2 marks**	**3–4 marks**	**5 marks**
The candidate will have:			
• Provided evidence of some post development testing.	• Provided evidence of final product testing for function.	• Provided annotated evidence of post development testing for function. • Provided annotated evidence for usability testing.	• Provided annotated evidence of post development testing for function and robustness. • Provided annotated evidence for usability testing.
Evaluation of solution (maximum 15 marks)			
1–4 marks	**5–8 marks**	**9–12 marks**	**13–15 marks**
The candidate will have:			
• Commented on the success or failure of the solution with some reference to test data. • The information is basic and communicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.	• Cross referenced some of the test evidence with the success criteria and commented on the success or otherwise of the solution. • Provided evidence of usability features. • Identified some limitations on the solution. • The information has some relevance and is presented with limited structure. The information is supported by limited evidence.	• Used the test evidence to cross reference with the success criteria to evaluate the solution identifying whether the criteria have been met, partially met or unmet. • Provided comments on how any partially or not met criteria could be addressed in further development. • Provided evidence of the usability features. • Considered maintenance issues and limitations of the solution. • There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.	• Used the test evidence to cross reference with the success criteria to evaluate the solution explain how the evidence shows that the criteria has been fully, partially or not met in each case. • Provided comments on how any partially or unmet criteria could be addressed in further development. • Provided evidence of the usability features justifying their success, partial success or failure as effective usability features. • Provided comments on how any issues with partially or unmet usability features could be addressed in further development. • Considered maintenance issues and limitations of the solution. • Described how the program could be developed to deal with limitations and potential improvements / changes. • There is a well developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.

0 marks = no response or no response worthy of credit.

Appendix 2: Candidate Authentication Form

All work must be your own, authentic and free from plagiarism. You will be expected to sign a candidate declaration form stating this.

Once you have signed your declaration and submitted work to the exam board, any questions over authenticity or evidence of plagiarism may result in a malpractice investigation. If found guilty, the consequences are severe and could lead to you being withdrawn from your A Levels.



Candidate Authentication Statement

OCR AS and A Level, Cambridge Nationals, Entry Level, Extended Project and GCSE

One copy of this form must be completed for **each** centre-assessed unit or component. The completed form **must be retained within the centre and be available on request.**

Centre name		**Centre no**					
Series (eg November, January or June)		**Year**	2	0			
Unit or specification title (eg Topic-based essay)							
Candidate name		**Candidate number**					

NOTICE TO CANDIDATE

The work you submit for assessment must be your own.

If you copy from someone else or allow another candidate to copy from you, or if you cheat in any other way, you may be disqualified from at least the subject concerned.

- Any help or information you have received from people other than your subject teacher(s) must be clearly identified in the work itself.
- Any websites, books, leaflets or other material (e.g. videos or artificial intelligence tools) which you have used to help you complete this work must be clearly acknowledged in the work itself. To present material copied from the Internet, books or other sources without acknowledgement will be regarded as deliberate deception.

Candidate declaration

I have read and understood the **Notice to Candidate** (above). I certify that the work submitted for this assessment is my own. I have clearly referenced any sources and any AI tools used in the work. I understand that false declaration is a form of malpractice.

Signature: ……………………………………………………………………………………

Date: ………………………………………………………………………………………

Appendix 3: Previous candidates’ project titles

- Private social network JS / React
- Member management system C#
- hospital database (a&e patients) Python + DB
- monopoly Python + pygame
- game (asteroids / space invaders) Python + pygame
- d&d game Python
- plant sale (DB, sales) C#
- cipher decipher C#
- Parcel delivery route planning Java
- burst pipe location (NN) Python
- Escape the zoo (maze game) Python + pygame
- Python chat server - client (with plugin ability) Python
- Go Cheaper (rail fare times and prices) Python
- garage stock inventory Python
- Planet orbiter simulation C#
- History Revision Game C#
- Maths Battle (simple arithmetic game) C#
- Song playlist generation Python
- Simulating people and traffic in a city C#
- Traffic Light Control System (investigation) raspberry pi (AVR)
- Network sim hacking game C#
- Pub inventory and stock system C# / SQL
- Click and collect website PHP / MySQL / C#
- Shop system (till / stock) C# / SQL
- Pokemon Battle simulator C# / SQL
- Medi-Minder (medicine app) C#
- Hotel reservation system Python
- MAths teaching system Python
- Economics question system Python
- Quote and schedule maker Python
- bowling simulation C# unity
- kam's tyre database Python
- Warhammer - army creation and analysis Python
- multiplayer card game Python
- Equation quiz system Python
- The maze game Python
- a website for upcoming artists (lyric analysis) Python / django
- DAW for the visually impaired VB.net
- graph theory algorithms Python
- elearning tool (revision) Python / django
- sports day management system Python
- room layout planner VB.net
- HR system Python / django
- golf management syste, Python
- blackjack Python
- projectile trajectory Python
- maths quiz app Python
- tank attack C#
- role playing game unity
- easy face detection Python
- harmonise chorales VB.net
- chess with ai Java
- stealth game Python
- tower defence game VB.net
- energy monitoring application Java / android
- checkers with ai (gen alg) Python
- advanced graphing tool Python
- street assassin (game) Python
- emergency route finding engine Python
- pedestrian crossing simulation VB.net
- simulating pre-industrial propulsion changes Python

Appendix 4: Final submission checklist

Checklist	Completed (/)
Ensured that all sections of the report are present	
Reviewed the mark scheme to ensure all marking criteria have been addressed	
Proof-read report to check spelling, grammar and punctuation	
Proof-read report to check all screenshots are clearly visible	
Reviewed code to check that sensible identifiers are used throughout	
Reviewed code to check that commenting allows for maintainability	
Ensured all references to sources of information are included	
Ensures all references to assets sourced from somewhere else are included	
Checked that the contents page matches the page numbers	
Zip your source code and assets and uploaded onto Teams assignment	
Saved a Word version and PDF version of your code and uploaded onto Teams assignment	