A-LEVEL
COMPUTER SCIENCE

7517/C
Report on the NEA

7517
June 2022

Version: 1.0

Further copies of this Report are available from aqa.org.uk

Introduction

Centres that entered students in 2022 should read this report in conjunction with the specific feedback sent to them on the publication of results. The comments below highlight the observations of Senior Moderators during this year's examination and should be used with the subject specification, assessment criteria and the NEA guidance document (https://filestore.aqa.org.uk/resources/computing/AQA-7517-NEA-GUIDE.PDF) which is available on AQA's website. Exemplar and standardisation material are available on AQA Centre Services through the Teacher online standardisation (T-OLS) system where there are 10 standardised projects.

This year saw the return of full moderation of the NEA and it was pleasing to see the continuation of the wide range of projects that were last seen by moderators in 2019. It is also pleasing to see the wide range of environments used. Both of these allowed students to select a project that fits their skills and in the main this was seen by the moderation panel. Over the year there were many centres who made use of the NEA Advisor email addresses and gained answers to questions around the running of the NEA and the suitability of project proposals. Occasionally, however, it is clear that a student took on a project that was not suitable and then struggled to navigate the requirements of the NEA. The use of project proposals and a focus on getting the analysis section completed early might help certain students perform better.

Analysis

It is pleasing to see s taking time to scope out their projects so it is clear to a reader the aims and direction the project will be heading in. Many students take time to research into areas of their project highlighting strategies, libraries, APIs and other needs for their project. It is also pleasing to see those students referencing tutorials and other sources they have been using as part of this stage.

Initial modelling is still a challenging area and it is pleasing to see a more varied choice of content for this section. A DFD, for example, may not be the best way to model games whereas a sketched-out storyboard showing an idea of levels and content would help both scope and model the proposed solution.

Design documentation

Students should take time to explain the workings of their project rather than just producing copious amounts of pseudocode. One student had over 30 pages of pseudocode with no text present to explain. Contrast that to a student working hard to explain the stages required, using text alongside diagrams and example data, to turn a finite state machine (FSM) into a regular expression (RE) and it is obvious which design presents a clearer view of how a project will work.

A number of centres followed the list of suggested parts of a design as a prescribed order and ended up having, for example, an SQL section where many queries for all parts of the project are grouped together. It is felt that students perform better if they split the design into 'chunks' following their overview and then talk about how that chunk will work by using diagrams, text, example data and example SQL all together where appropriate so it is clear how that section works.

So, for example, when designing a login section for a project the student might have a section in the design documentation labelled 'Login' and then have a sketch of the UI for this part of the project, talk through the login process and structure out the algorithm required with the SQL queries necessary to implement the login process.

Some centres this year had designs that followed more of a diary format with students explaining a step and then the next step with explanations of how they fixed bugs. For this specification the design section is not meant to be in diary form but rather a summary of how the project works.

Completeness and Technical Skills

It was pleasing to see centres make use of a variety of ways of assessing both completeness and the technical skills. Asking students to include a cover sheet in front of a copy of their code is one way that centres provided evidence of the skills. On the cover sheet students clearly identify skills used in their code, for example 'Inheritance', and then provide a page number to refer to. Other centres performed this by adding an extra sheet to the project log identifying the skills alongside page number references.

Some centres, unfortunately, still only provide comments on the project log such as 'Many skills from Group A present' which is not really providing the required justification / evidence for any mark awarded.

A few centres had students who did not submit full copies of their code and this made moderation difficult. It is a requirement for a student to submit a full copy of their code.

Students providing code in 'dark mode' (where the code might be in a barely readable colour against a black background) and/or via screenshots where this is not sized appropriately again make moderating difficult. Students should be encouraged to look at their documentation printouts and make sure that everything is clear and readable.

Testing

The move to video testing is encouraging to see as it provides many benefits over the traditional use of screenshots. A group of students are providing commentaries over their video, and this really helps get a feel for the project and also how the student is attempting to test it. It is still important, however, to have a test plan in the documentation providing an overview of the kind of testing and how this links to the project objectives.

It was pleasing to see students testing functions / classes / modules in their design sections and referring to this in the testing. Also to be commended are those students who verified their project's output by other means: by sketching a maze to make sure the shortest path was correct, calculating by hand a projectile's trajectory using SUVAT, or checking on a chess board to see if an AI move was appropriate. One student even took to finding out chess moves that AI engines find problematic and tested against these.

Those that used screenshots and provided descriptive text alongside generally gave good evidence of testing. It was also pleasing to see console output of data structures and interactions with database tables as a way of providing supporting evidence that a system is working with more than just basic input and output.

A number of centres provided a central location from which their student videos could be accessed. Moderators were given access to shared cloud drives and even websites to showcase the projects. Students who just inserted a hyperlink but modified the link text to say ‘My Video’ caused problems for moderators as they unfortunately could not click on the printed link. Those who provided URL links to follow did mainly allow moderators to access their video but it is important to remove the hyperlink formatting and pick a font that distinguishes clearly between characters such as 0 (zero) and (capital O), l (lower case L) and I (upper case I). Students should check on another device that their link is available as those set as ‘private’ did cause problems. The use of QR codes and URL shorteners also helped moderators working with very long URLs such as those for Google drive links.

Evaluation

The evaluation section is far easier to complete if a student completes a good set of objectives in the analysis section. It was pleasing to see students giving an overall evaluation of the ‘whole system’ before considering the objectives. Those that were thoughtful, critical and reflective over their attempt at meeting the objectives generally produced better evaluations. User feedback varied from the simplistic comments that reflected on how it worked to those that also considered deeper as to how the project had met the objectives. A good idea would be for students to provide the list of objectives to a user as part of the evaluation process.

As previously, students are encouraged to be honest and critical in their evaluation sections. It is better for the student and users to find the holes and gaps in a project rather than just praising an obviously struggling implementation.

Final comments

It was clear that some centres provided templates for students to use for their documentation and this is to be discouraged. This becomes clear when students forget to remove the commentary and hints provided in the template. Whilst students can be supported and encouraged to use certain structures the providing of a framework is not allowed.

It is also clear that some centres run sample NEAs and this then becomes problematic when students continue to develop the sample as many students end up with very similar documentation and actual code. The standard is to encourage every student to complete a different project. To find, in a sample, a group of similar ‘ordering’ projects with templated login code is problematic when trying to fairly assess both skills and the documentation.

It is good to see students identifying tutorials they have used in both analysis and design section. It is important that in the skills assessment markers carefully identify what code may come from the tutorial and what the student developed. Students are still tempted to find code online and treat this as their own. Centres are encouraged to discuss code with their students and probe for their understanding of it. As in previous years a number of students were referred for irregularities as their code could be found online from various sources including GitHub repositories, website tutorials, YouTube tutorials and courses on sites such as Udemy. A student who is submitting code that is above their standard should be detected by a supervising teacher and then investigated. It was pleasing to see some centres including notes to the fact that a student had tried to submit code that was not their own and upon interview they had therefore zeroed both the completeness and technical skills marks.

A number of centres submitted a variety of simple database projects where tool-designed interface forms (using, for example data grids) allowed the adding, editing and deleting of records from individual tables. A group of these students were classed as not being of A-level standard even if they had multiple tables, usually down to very limited objectives and testing showing nothing more than working with individual tables. A student working on a database centred project who is also using tools to help implement this within an IDE such as Visual Studio needs to think very carefully about what processing is required by the system in the analysis section so that they can complete algorithms of some complexity.

Upon moderation it was clear that there are centres who are still using structures from the old specification (2009) rather than the current one. Centres are encouraged to check any resources they are using have been updated to the current specification and, if necessary, check items they are intending to ask students to produce with their NEA advisor. An example of this would be the section ‘Data volumes’ which was required in the old specification but is no longer current.

In terms of NEA administration, please remember that when sending in the sample a CDF (Centre Declaration Form) should be signed and included. For each student we then need a signed CRF (Candidate Record Form) and a rationale for the marks awarded on the AQA Project Log or similar. In terms of securing each project it is preferable to place just one hole punch in the top left of the work (CRF, Project Log, documentation) and secure with a treasury tag.

In terms of support centres should remember that they have access to their NEA advisor via email and T-OLS which is now accessible via the AQA Centre Services website. Each year more standardised projects are added to T-OLS.

Mark Ranges and Award of Grades

Grade boundaries and cumulative percentage grades are available on the Results Statistics page of the AQA Website.