Questions matter

A-level
Computer Science

7517/C NEA

Report on the Examination

7517
June 2024

Version: 1.0

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

General Feedback

The NEA continues to develop with most centres now well aware of the requirements of this component. It is useful to start this report with a reminder of the main sources of information and resources regarding the NEA as a minority of centre and teachers are unaware of these:

- TOLS – teacher online standardisation – each year new NEAs are uploaded and continue to extend this bank of standardised projects. These are a useful resource for any new teacher and/or centre and for use as exemplars with students and during internal standardisation.
- NEA adviser – each centre has an NEA adviser who will respond by email to any questions regarding the NEA.
- Project Log – this is the recommended document to complete for each NEA when providing a rationale for marking.

In 2024 centres continued to submit a wide variety of projects using a wide variety of environments. Whilst it is appropriate for students to have access to this wide choice of environments it can lead to issues during supervision and marking if a centre does not qualify the knowledge and skills of the student.

A minority of centres allowed students to work through tutorials for these environments yet awarded high marks for skills and completeness. Many students did refer to these tutorials/resources but often the marker did not comment upon this in their rationale for the skills marks. For example, awarding high marks for skills such as path-following when this was clearly code just taken from a tutorial. For the students that do place references into their work, skills should be awarded for their implementation – the NEA adviser is of help in these circumstances. In a few cases students did not refer to the tutorial used and were referred to the irregularities process when this was not picked up by the centre.

Many students go through an impressive process of setting up a project, implementing it and then documenting in a way that makes it clear that the student understands their work. To support their work, it was pleasing to see more students making use of videos with commentaries.

This year saw the move to online submission and whilst the announcement of this provided little time for centres it appears to have worked well. For the moderation process it is to be noted that PDF submissions are to be preferred over formats such as Word documents. It is easy to save/export a Word file in PDF format and the moderators benefit from this.

Analysis

The comments for 2024 follow on from the 2023 report and we continue to see some common issues with this section.

Scoping – students do not take the time to fully scope out their project leaving questions in the mind of the reader. To 'fully scope' details such as the number of students, the number of products, the topics to be covered, the formulas to be used, the current number of customers, the number of lessons per day – all of these could be part of scoping out different kinds of projects.

Initial modelling – for many centres this may be missing or hard to locate. Having a section titled 'Initial modelling' at the end of the analysis section helps clearly provide evidence for this area. Initial modelling can mean many things, and these do depend on the project context.

Some centres still encourage all students to complete one kind of diagram across any project and this does not always work well. For example, whilst DFDs can be useful for projects dealing with the processing of data they do not work well for projects such as simulations and games. Centres are encouraged to give students a list of ideas that they might use for initial modelling and suggestions as to when each type of diagram might be appropriate. For example, good initial modelling for simulations involved sketches of the proposed output(s) and linking this to the formulas/processing required to generate the output.

Documented Design

This section continues to provide clear differences across the ability of students but also, sometimes, how centres encourage students to complete.

The design is the section from which a reader should be able to build a solution. There are students who do not design the main processing parts of their project and this will limit the marks available. For example, a quiz system where the student does not detail the actual mechanics of asking a question, marking a question and the processing of the results.

An area for centres to consider further is the work students place into explaining how output is produced and positioned, for example an SUVAT simulation where a student might just supply one sketch of the interface and not present any further information. How are the values for any formulae turned into data for drawing the projectile? What happens in the next 'frame' and is this explained? To get output onto the screen can require many steps and this is an area where students could provide more detail.

For data projects many students sketch out a layout but do not include any example data. Whilst this is not obligatory it can be beneficial to show some data on the sketch. The student also can explain how the data is collected and then processed before displaying on the screen, eg how are multiple rows represented? what happens when data fills an area of the screen – does it automatically show a scrollbar or move to a paged implementation? what query was used to generate the data for this sketch?

Skills and Completeness

These sections continue to be assessed well by centres. As mentioned earlier centres must be mindful of checking and distinguishing between borrowed' code and code written by the student. Whilst students are to be encouraged to make use of/extend/integrate ideas there is a distinction to be made and this becomes more problematic if a large amount of code is just copied across from another solution/project.

Whilst many centres assess completeness appropriately a minority do not fully consider what might be expected of a project at A-level standard. This does not tend to happen for those projects that clearly stretch am able student but for slightly simpler projects. Maze projects continue to be popular but clearly vary in complexity and therefore should receive a variety of completeness marks. Reference to TOLS and use of the NEA adviser can be useful resources to help measure completeness.

For a number of centres, it appears that students are asked to complete a 'technical solution' section, and this is not really part of this specification. In these 'technical solution' sections a student explains sections of their code (line by line) or sometimes completes a diary-like log. For this specification the

student does need to provide a full copy of their code but there is no need to explain each part. A page giving references to areas where skills are demonstrated is useful for the marker/moderator.

Some centres may feel that this is an extension to the 'documented design' section but for the majority just listing a section of code and then explaining it (line by line) does not provide any further credit to the design marks. Occasionally these sections did have sketch work / example data / run throughs, and these did allow credit to be awarded back to the design section.
If these sections are being provided as evidence for the design section centres are encouraged to change how this is completed otherwise it is a section that is not required.

Testing

Many centres have now moved to video testing for evidence, and this is clearly the preferred manner. Screenshot evidence has many limitations for the kinds of projects being produced and, it can be argued, takes more time for the student to complete. A hybrid of video evidence and then some screenshot evidence of items such as unit-testing and console output can be useful in some circumstances.

Students who provide a commentary over the video greatly help a viewer.

Students should be encouraged to concentrate on the main processes of their project and in most cases, this is **not** user registrations and login. A video of 10 minutes allocating two-thirds of the time to registration/login/incorrect passwords and then only demonstrating the core components of the project in the remaining time has the allocation of time the wrong way round .

Students should also be encouraged to test any processing and not just rely on any output as proving that a system works, eg

- testing an AI chess system should consider whether a move is appropriate rather than just the fact that it made a move – indeed there are websites providing positions with which AI chess systems struggle
- a Rubik's cube solver can be tested against an online solver to check that a sequence of moves matches up
- a calculator can be tested against an online calculator.

This 'manual checking' that an output/result from the solution is correct/appropriate is an important part of the testing and tends to be done well by stronger students but missed by many others.

Evaluation

Evaluations continue to be completed and assessed well. Students are encouraged to start with an 'overall evaluation'. This gives a student an ideal opportunity to reflect on the system as a whole and take time to highlight both the strengths but also the weaknesses in the submitted solution.

User feedback varies in quality and an effective way is for the reviewer to have read the analysis stage and have a copy of the objectives to provide their feedback.
Honesty is to be encouraged in the evaluation section, and this can provide a basis for suggestions for improvements and extensions.

Mark Ranges and Award of Grades

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