AQA

A LEVEL
Computer Science

Non-Exam Assessment (NEA)
Report on the Examination

7517
June 2018

Version: 1.0

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

Introduction

Schools and colleges that entered students in 2018 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 Booklet which is available on AQA's website. Exemplar and standardisation material are available on the e-AQA website secure key material area. It is still apparent that some centres did not have a real grasp of the new NEA even though material has been made available online and through 'Focus on the NEA' and face to face 'teacher standardisation' meetings that were run during the academic year.

A key part of the NEA is consideration as to whether a project is of 'A-level standard'. There were a collection of centres that submitted work marked as 'A-level standard' that, when moderated, were clearly not of this standard. To assess a project as being of 'A-level standard' the key place to look is in the analysis section. The key requirements of a user and the objectives set by the student should show clear potential for the technical solution to include skills that involve the complexity required for A-level even if a student does not meet these objectives in their final code. It was clear that some students had not set objectives that were challenging enough to be considered at the correct standard and this then affects the marking of the whole project as written in the specification. Students should be encouraged to make sure that the objectives they initially set for their project are challenging. If the student does not meet these objectives they should not be removed from the analysis section but rather be considered during the evaluation part of the project.

AQA provide NEA advisors who can be contacted over the academic year and are happy to discuss issues such as 'A-level standard'. Centres should be aware, however, that many ideas can start off with the potential of being at the correct standard but this should be finally assessed when marking the submitted analysis section.

Administration

Many centres completed all the administration requirements fully and submitted excellent assessment comments with the projects which is very helpful to the moderation process. Comments that describe how a student has met assessment points and provide page numbers to support this are really useful. A comment such as 'shows a lot of Group A skills' does not really provide the detail that a moderator requires and it would be helpful to identify items such as 'complex use of OOP with good use of inheritance (page X) and composition (page Y)'.

Some centres used their own assessment forms whilst others used the AQA Project Log. It is more straightforward for a moderator to agree with centre marks if the administration has been completed correctly and centres are reminded that they are required to provide evidence to support their marking when submitting the sample.

Centres are advised that the sample should be sent by 1st class post as required in the instructions to examinations officers from AQA. Sending by a courier requiring a signature can delay the sample being received by the moderator.

Samples are packaged up in a variety of ways but the simplest is to hole punch the top left corner and use a treasury tag to attach the CRF, Project Log and documentation together. The use of bulky ring binders or individual wallets is not required and having loose sheets does not help the moderation process. It was also noted that some centres submitted electronic samples on USB or optical media. This is not currently an accepted method of submission and moderators are instructed not to open electronic submissions but to request a hard copy sample from the centre.

The most frequent errors with the administration were:

- failure to submit a Centre Declaration Sheet with the sample sent to the moderator
- failure to add up section marks correctly on the Student Record Form (CRF)
- failure to check that the total mark on the CRF is entered correctly into the electronic submissions system
- failure to ensure student authentication signatures are on the CRFs
- failure to submit any supporting evidence of the assessment apart from the individual marks on the CRF

General

There continues to be a wide variety of projects submitted from centres, showing a good deal of initiative and exploration from students. One project that stands out for initiative was a computer controlled laser seagull scarer. This project used an Arduino to control an ultra-violet laser and a Raspberry Pi to track any seagulls with the Pi passing coordinates to the Arduino to 'blast' the seagull and scare it off.

Again this year we saw centres that had students submitting very similar work and this needs to be considered when students present their proposal at the start of the project. The specification states that "There is an expectation that within a centre, the problems chosen by students to solve or investigate will be sufficiently different to avoid the work of one student informing the work of another because they are working on the same problem or investigation." Centres should also be careful about providing table structures for groups of students to use as this then looks like provided resources rather than student work.

A lot of students continue to attempt projects based around a quiz element. The majority still end up as simple multiple-choice quizzes which are then often marked generously by centres. The NEA Guidance booklet provides guidance as to how a quiz can be made suitable for an A-level project.

Students presenting database projects, of which there are many, can end up scoring highly when these are managed well. Students who just allow data to be added to a table and then provide a few ways of selecting data from this table, or tables, will struggle to score high marks as there is now a greater focus on the processing of the data in the marking criteria.

Students using well known frameworks, for example Django, Pygame or Unity, must make sure that they do not just follow a tutorial to complete their NEA project. It is obvious that certain students do just follow tutorials and this is not referenced by the student or the centre. Whilst it might be appropriate for a student to make use of an available library or algorithm it is not the focus of the NEA to just copy an available tutorial or to re-implement a project from a site such as GitHub. Students who do make use of external code, be this a well-known library or just a small project on GitHub, should make this very clear in their design section and when printing off the final solution. There is ample opportunity for a student to discuss what external code might provide for the project, how it will be implemented into their code and provide examples of data coming in and out in their design section. It was clear that sometimes centres were awarding technical skill and completeness marks for sections of the project that were not written by the student themselves. There were also occasional examples of the whole technical solution coming from a YouTube tutorial, online courses and GitHub repositories and these were referred to the irregularities and malpractice department of AQA. Centres are reminded that they should be authenticating that the work completed by students is their own.

Analysis

A lot of centres are generous in awarding marks for the analysis section. The top level is looking for a 'fully or nearly fully scoped analysis, presented in a way that a third party can understand'. A

lot of students are not providing enough clear detail so that the reader can understand what the problem is about and the objectives that are to be met. The modelling part of the analysis section can often tend to be missing but this provides the opportunity for a student to, for example, provide the storyline of a game to be developed with some initial sketches as to what levels might look like. It was common for a project based around the concept of a game to not really provide enough detail as to how the game would play or be structured.

It was pleasing to see centres respond to the comment last year that interviews of potential users were very superficial and not probing enough. One centre is to be commended for setting up the interview responses in a very structured manner with each question asked followed by a summary of the responses and then any specific objectives/requirements to be drawn from this question and response.

The quality of the objectives set by students varies considerably. A good idea is to just look at the objectives on their own and consider whether from this list you get a real feel as to what the project is about and some of the complexity that will be required in the solution. Students sometimes introduced complex ideas in the research part of their analysis but then failed to mention anything about these in their objectives.

The analysis section is critical for the project and centres are to be encouraged to seek advice from their NEA advisor whilst analysis sections are being written.

Documented Design

The documented design is an important section for the student to provide evidence of their understanding as to how they will implement their project. It was common, however, to see poorly structured design sections marked generously by centres.

It is most beneficial if the design section starts with an overview. Whilst this might contain a variety of diagrams students should take time to explain what these diagrams are showing. The aim of the overview section is to show how the student has broken down their solution. This is also a good opportunity to discuss any framework that is being used and to discuss how this affects their design.

Whilst it might be appropriate for a student to give a brief introduction to a particular data structure it is considered more important to spend time discussing how that structure will be used. So, for example, the theory of linked lists might be relevant but the student must then indicate how, where and even why a linked list is going to be used in their solution. It is even better if they provide some example data for their project and show how this might be manipulated by algorithms utilising this data structure.

Students should be encouraged not to focus on providing the pseudo-code and descriptions of well-known algorithms such as merge sort but instead focus on explaining how these might be used by the project. It would be better for a student to focus on the parts of the project that will require designing. So, for example, a moderated quiz project design might have contained a section on merge sort but not have covered anything about how an answer to a question would be assessed to be correct or not. It can be assumed that the reader of the design section understands merge sort but would need to know the details such as question generation, display, assessment and tracking.

Completeness

It is pleasing to see centres consider more carefully how they award marks for this section. There was a greater usage of the spread of marks and also more detailed comments in project logs as to how these marks had been awarded. A few centres continued to award full marks for the majority

of projects sampled and this was clearly not appropriate when you consider the variety of completeness in the projects.

A project that only meets objectives and skills that are around high GCSE level should only be entering level 2 at best. Whilst the student might be meeting their objectives consideration must also be given to the requirements of the actual project and potential users.

There are projects on e-AQA where the student has met their objectives but the completeness mark has remained in level 2 and centres are encouraged to look at these. A common situation for this to happen in is with quiz systems and 'teaching' systems. If, for example, a student introduces the idea of A-level students using a program to understand Djikstra's algorithm as part of their introduction to the analysis stage then this must be considered when marking the completeness. The student might have met all of their objectives in terms of working out Djikstra's algorithm and displaying the results but if the system does not help a user understand the algorithm then it can't be marked as fully complete.

Techniques Used

Identifying techniques used continues to be performed well by most centres. It would be beneficial for students to provide a 'summary' of the skills they feel are evidenced in their implementation by placing a cover page before the code. This can simply be a list of the skills alongside a page reference as to where this can be found in the code.

Centres need to be careful in awarding credit for 'complex OOP' when there is little evidence of this either in the design section or the final code provided. Where an environment forces a student to use OOP techniques this does not really provide all of what is required to gain credit for 'complex OOP'.

It was evident, however, that there is still a continuing move towards using OOP techniques in the coded solutions.

When identifying skills and awarding marks, consideration should be made of the coding completed by the student and its effectiveness. The calling of an MD5 or SHA library function is not to be confused with writing a hashing algorithm. In a similar fashion the usage of a merge sort, whilst mentioned in the Group A table, is not perhaps as complex as a student writing their own code to perform something novel for their project and merge sorts were often be implemented by all students at a centre without consideration as to the appropriateness for a particular project.

It was surprising to see quite a few students submit evidence of their code using very small font size that could not be read. It is the assumption that the centre marked from the live code rather than this documentation as this would have been impossible. Students should make sure that the code listing is printed in a font size that is appropriate and taking screenshots is not a suitable way compared to copying the code across and then formatting in a sensible way.

Centres where the marker annotated the code listings are to be commended in helping provide evidence as to the skills being demonstrated by the student.

Testing

It is pleasing to see more centres submit evidence of testing using video. For certain projects this can really help provide evidence that objectives have been met and actually be less time consuming for the student compared to providing a large number of screenshots. Students should be encouraged not to focus on testing too much of the UI of their project and aspects such as login and registration. When a project has a main focus on some sort of processing this should be where the majority of the testing is focused.

It is also pleasing to see students begin to realise the important of performing some manual checks for the processing of their solution. So when a depth first search, for example, is performed this is shown to be correct by also manually performing it. Whilst it is not appropriate to manually check every action of the solution it is appropriate to perform this once or twice to show that their code is performing as expected.

In a similar fashion with database projects it is pleasing to see students realise that it is appropriate to also show the contents of the tables whilst performing some tests so that it can be evidenced that the program and the database are interacting correctly.

There are students who focus more on testing the specific objectives independently and perhaps forget to test the system as a whole. So, for a quiz, an important test would be to work through a whole test to show the functionality of the system ending with a score and perhaps an entry into relevant database tables and updating of tracking data. Students who submit video testing can often show a whole system test across the video and then use timestamping to indicate where specific tests have also been performed.

Evaluation
Students with good objectives set in their analysis section could go on to score well in the evaluation section. They should, however, be careful to paint a fair evaluation of their project. If certain objectives have not been met very well then it is more beneficial to reflect on this rather than gloss over it. In the same way, user feedback should also be honest as it was common for a project to receive glowing user feedback when it was obvious from the technical solution and testing that there were serious limitations to be commented upon. To gain the top level a student is required to consider some improvements in detail and whilst this is not to final coding detail it would be appropriate for a student to take at least one idea and consider the changes and challenges this would present if it were to be implemented.

Mark Ranges and Award of Grades

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

Converting Marks into UMS marks

Convert raw marks into Uniform Mark Scale (UMS) marks by using the link below.

UMS conversion calculator