BLUFFER'S GUIDE TO NEA TASKS

The new Computer Science GCSEs in England and Wales all have an assessed component requiring students to spend 20 hours on a practical programming project. Alan O'Donohoe, leader of exa.foundation, looks at the implications for teachers

What is an NEA?

NEA stands for Non-Examination Assessment. This requirement was introduced to replace previous coursework components. It is a practical component that is formally assessed as part of the GCSE, with a weighting of 20 percent of the final GCSE grade.

Why do the NEAs exist?

A large amount of Computer Science involves practical problem solving. While the new GCSEs include an exam that tests students' knowledge and understanding of programming, computational thinking, and algorithms, the NEAs allow students to submit some coded solutions to problems. There have been some historical problems with coursework, and the NEAs are intended to provide a more robust mechanism to ensure a level playing field for all schools. As the NEAs are relatively new, there are many procedural changes to assessment to implement. Schools may take a while to adjust to the new processes.

PLEASE NOTE:

The author neither works for, nor represents, any examination bodies. As such, the information in this guide is intended only as practical advice for a wide audience, not as an official guide to one GCSE NEA. Since examination bodies vary in their rules for the NEA, teachers should familiarise themselves with the specific rules governing their choice of qualification.

KEY:

[!] Indicates areas with a significant degree of difference between different exam awarding bodies. Please check the details for the body that you are using.

What will the students be assessed on?

Students are required to solve a sequence of programming challenges. These typically include a requirement to analyse each problem, design algorithms, then develop, test and evaluate their solutions to the challenges. Teachers will need to familiarise themselves with the specific requirements of the relevant specification.

Lack of familiarity with the requirements has, for example, led to some teachers stressing about two-dimensional arrays when the requirement of their specification is only for one-dimensional arrays, or worrying about teaching SQL when the requirements are only to write to and read from a file. While there is absolutely nothing wrong with students using these methods, the tasks have been designed to require only the skills listed for the NEA.

When are students expected to complete this?

Students are allowed 20 hours of supervised time to complete the NEA. The teachers can choose when to start and end the 20 hours, and even stop before the 20 hours if students have finished. However, the NEA tasks are not released until 1 September, and students' completed work must be marked and sent for moderation by 31 March. **[!]**

Schools can choose when to schedule the NEA sessions within that window, but NEAs can only be attempted during the terminal year of assessment, i.e. the same year as the exam. There is no requirement for schools to log every lesson allocated for the NEAs, but it would be sensible to do so. Teachers should plan ahead to consider what to do with students who complete their projects in less than the allocated time. If a student 'gives up' after just three hours, this could cause problems for the remaining 17 hours.

It would be wise to start the NEA as soon in the Autumn Term as possible, and complete it as late as possible. This would allow teachers the flexibility to plan for refresher lessons during the 20 hours, and allow for other unplanned circumstances such as teacher or student absence.

Splitting the 20 hours up over an extended period would also go a long way towards keeping students' motivation levels high. Their attention is likely to be sharper in the non-NEA sessions where teachers provide refresher programming lessons, since the students could then apply this to their NEA solutions.

How can teachers best prepare their students for this beforehand?

Throughout Year 10, plan for lots of practice with lots of stepped paired-programming challenges designed to slowly build confidence and competence. While it's tempting for teachers to focus purely on teaching systems theory in Year 10, doing so would seriously disadvantage students with the NEAs. Some teachers choose to split their timetabled lessons 50/50 between systems and programming; I chose a 20/80 split, with more time allocated for the programming element. In Year 11, a refresher course at the beginning of the autumn term will put students back onto a firm footing before starting the NEAs.

It would be wise to introduce students to the restrictions imposed by the NEA as early as possible in Year 10, so that it's less of a shock for them when they start the NEA. This would allow any Resource Banks to be developed throughout Year 10.

■ Students should be prepared before the NEA begins - teachers cannot give specific help or feedback during the NEA task

■ Teachers are free to run after-school programming sessions as long as they are not teaching to the NEA task

Should teachers stop teaching programming as of 1 September?

No. However, it would be best to ensure that students have been properly prepared before then, as teaching to the tasks is strictly prohibited. If the NEA sessions are scheduled to be spread out across the autumn and spring terms, this would allow more opportunities for refresher lessons.

Can teachers offer after-school programming sessions?

There is absolutely no problem with this as long as teachers provide general support and advice with the programming requirements, and do not teach to the NEA. After 1 September, students are prevented from bringing in any electronic, written or printed resources from this additional support to their NEA sessions.

What if teachers spot something after the NEAs are live that they omitted from their teaching?

Again, as long as any teaching post 1 September does not include support or solutions to the live NEAs, there is absolutely no problem with this. For example, if after the NEAs are published the teacher feels that their students need to use a particular data structure that they have not taught, the teacher could plan a general lesson on data structures and include the required structure among several that are not required. However, the teacher should be very careful not to drop hints or suggest which data structure to use as this is classed as providing solutions.

What if the teacher has a very limited amount of time available to prepare students?

In much the same way as learning any other new skill (driving a car, drawing or learning the violin), it is possible to learn certain things in a reduced amount of time. However, to really learn problem solving, algorithms, computational thinking, and programming to a level sufficient for the NEAs takes hours and hours of repetitive practice. Anything short of this is unlikely to prepare students properly. In my experience, teachers and students who have not had sufficient preparation tend to look for shortcuts out of desperation, e.g. searching for solutions from others, or looking for the answers to an important test or exam. ▶

Can teachers find solutions to the live NEAs?

While this may be possible, it is a very dangerous path to go down and I would caution teachers against it. Teachers are prohibited from providing their students with solutions to the live NEAs. If the teacher subsequently discovers what they consider is a better solution to a problem than the one their students would consider using, it would be very tempting to show these solutions to the students.

It is worth stating that there can be many different approaches to solving NEA-type problems. Once you have seen one solution it's impossible to 'un-see' it, and hard to develop any alternative solutions. My advice would be to avoid looking at solutions to NEAs, and caution students against it as well. Also, some solutions shared in the past have contained errors. If you learn from these or share them with your students, you're not just committing malpractice, but also proliferating the errors.

How can teachers support students during the NEAs?

In the actual NEA sessions, support from teachers is strictly limited. Outside the NEA sessions, teachers can give generic advice and support, providing it does not refer to the NEA tasks or include solutions. Students may also do their own research outside the NEA sessions, but most exam boards prevent students from bringing evidence of this research into the NEA sessions – students may only use what they can remember. Teachers might decide to plan some non-NEA sessions during the 20-hour period for refresher lessons, e.g. file-handling practice, data structures, etc.
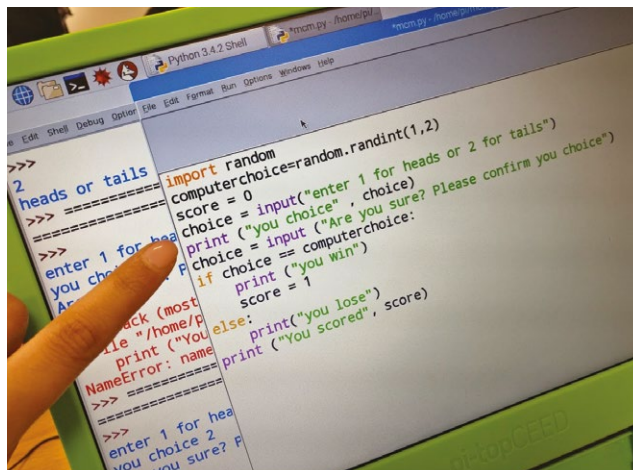
■ The resource bank can contain examples of the students' own solutions to problems set during Year 10

students can be provided with a pre-prepared resource bank. After 31 August, nothing may be added to the resource bank, but resources may be removed. Eduqas/WJEC and Edexcel allow students to be provided with examples of syntax, but no coded solutions or examples of NEAs. For AQA, students are allowed full access to the internet during the NEA. Personally, I think that this is a distraction – I think that students should be prepared before the NEAs. If a student needs to search for help during an NEA, it shows poor preparation.

> ❝ THROUGHOUT YEAR 10, PLAN FOR LOTS OF PRACTICE WITH LOTS OF STEPPED PAIRED-PROGRAMMING CHALLENGES DESIGNED TO SLOWLY BUILD CONFIDENCE AND COMPETENCE

If a student asks how to do something 'generally', but it relates to the NEA, what should the teacher do?

As long as the teacher is only giving general advice and support, there is no problem with this. It would be worth clarifying this with students, and explaining that there are strict rules teachers have to follow, regardless of student difficulties.

What feedback can the teacher provide to students?

The simple answer is 'absolutely none'. There are strict rules limiting teachers from providing feedback to students. It is likely that you'll still want to provide students with some regular assessment feedback during the period of the NEA, so why not set a much lighter parallel project on a completely different theme, unrelated to the NEA? This would also serve as a useful teaching aid, but it would need to be carefully planned to avoid similarities to the NEA.

What resources are students allowed to use in the NEAs?

This is where the widest variations exist between examining bodies. For some GCSE qualifications, for example the OCR GCSE, no internet access is allowed at all during the NEA sessions, but

For some teachers, this limit on resources has been a major factor in choosing awards. If you're joining a school where the resource bank has not yet been prepared, you could probably use one from another school – providing it was prepared before 31 August. [!]

How should students reference resources (when allowed)?

This is quite simple. If students use something in their NEA that they have copied from a resource, they should reference it. If they already know it, they don't need to.

How will students manage without internet access?

Some exam bodies restrict use of the internet. [!] Having conducted controlled assessments with internet access being allowed, I'm not convinced it helps the students during a formal assessment. It can lead to laziness and the additional risk that solutions may be shared. Not having the internet available for the NEAs is a challenge, but it forces a different way of thinking and discipline – using the resources you have available to address the problems you're being asked to solve.

To prepare students for working on NEAs without internet access and using the resources provided, set some practice tasks before the NEAs to help them get used to the restrictions and develop good habits. This will also reduce any potential stress for students working under these controls.

What should an OCR resource bank include?

Teachers following the OCR GCSE course can put resources that students might find useful into a resource bank, up to 31 August. This includes wall displays, electronic resources, and paper-based resources, but not templates or writing frames. Where allowed, it is recommended that teachers provide students with the following reference materials:

- the students' own solutions to previous challenges set throughout Year 10 to remind students how they solved previous problems

- some carefully selected examples of previous students' projects e.g. from previous years

- reference materials reminding students of the syntax of the chosen programming language

Which resources should I buy for my students?

Thankfully there is a great variety of free downloadable resources available to support students (where allowed), so in theory you shouldn't need to buy anything. However, some publishers have released paid content which schools can purchase. Teachers who have purchased these have said they found them really helpful. Ask other teachers which resources they've found most useful.

How should we store the students' work?

All student work must be stored securely to prevent students from accessing their work outside the NEA sessions. Some schools have created exam accounts with restricted access to the internet, USB ports and publicly shared drives. It would be a good idea to get students into the habit of using these before the NEA starts.

Who should I ask for advice?

Make sure you ask the right people the right kind of questions. While you should always check with exam board representatives and documentation for official rules and regulations, asking other teachers as well will help you to find out how they are approaching the NEAs and preparing their students. However, you should be careful how much trust you place on advice from others who do not represent exam board officials – just like the advice in this article!

Isn't this a lot of hassle for just 20 percent of a student's final grade?

I think it's dangerous to assume that programming is only worth 20 percent, when computational thinking, algorithms and programming combined are worth 60 percent in total. In my teaching, I've managed to reduce the time required to teach theory down to about 20 percent, allowing about 80 percent of the time for computational thinking, algorithms and programming. The students were least familiar with the latter, so a 50/50 split would not have worked as well. A lot of the theory they can learn themselves without direct supervision from me.

Security

To maintain integrity, the task instructions and student work are not allowed to leave the NEA sessions. Students are not allowed to bring additional resources into the sessions. [!]

What if some schools decide not to follow the rules?

Through the Joint Council for Qualifications (JCQ), the exam boards have jointly said they will be conducting an unspecified number of centre visits this year, along with statistical analysis of the NEA results compared to exam results. Schools can expect a visit at any time, and the exam boards don't need an excuse to trigger a visit. (HW)

MORE INFORMATION

- Blog post on NEAs: **exa.im/helpwithnea**

- Teachers at Outwood Grange Academy Trust have compiled a helpful NEA resource: **helloworld.cc/2uHij4m**

- Chris Roffey has shared these Code Cards: **helloworld.cc/2vICM5U**

- Mark Clarkson's guide to Python: **helloworld.cc/2gQdGzm**

- OCR Guide to pseudocode and Python: **helloworld.cc/2vFa7BO**