2020 Digital Technologies Subject Assessment Advice

Overview

Subject assessment advice, based on the 2020 assessment cycle, gives an overview of how students performed in their school and external assessments in relation to the learning requirements, assessment design criteria, and performance standards set out in the relevant subject outline. They provide information and advice regarding the assessment types, the application of the performance standards in school and external assessments, and the quality of student performance.

Teachers should refer to the subject outline for specifications on content and learning requirements, and to the subject operational information for operational matters and key dates.

School Assessment

Assessment Type 1: Project Skills

Students produce four project skills tasks in which they examine approaches to identifying, deconstructing, and solving problems of interest by applying:

* computational thinking skills, including abstraction
* data analysis skills
* design and programming skills
* iterative project-development techniques.

The problems chosen should be of interest to the students.

At least one of the tasks should involve students working collaboratively, with each student providing individual evidence of their role in and contribution to the collaborative task.

At least one of the tasks should focus on students analysing simple and complex data sets related to a problem of interest to identify patterns and/or trends, draw conclusions, and make predictions.

At least one of the tasks should involve an assessment of programming skills.

In at least one of the tasks, students research and discuss the ethical implications of data use and/or digital solutions for individuals, groups, societies, and/or the environment.

The tasks should be presented in multimodal form. Together, the four tasks should be the equivalent in multimodal form of a maximum of 20 minutes.

Tasks may include, for example:

* a screen-capture validation, highlighting innovative features
* a multimedia presentation of a solution to a problem, analysis of data, or prediction of trends
* an annotated prototype solution or proof of concept
* an informed debate about the ethics of data storage and use.

For this assessment type, students provide evidence of their learning primarily in relation to the following assessment design criteria:

* computational thinking
* development and evaluation
* research and ethics.

The more successful responses commonly:

* stuck to maximum time limit of 20 minutes
* addressed all criteria in the multi modal video evidence, explicitly addressing the performance standards
* used key terms from the performance standards
* submitted four videos (clearly identified with Skills Task 1, Skills Task 2 etc) or one combined file (example: PowerPoint with four video files embedded)
* clearly presented iterations, development, modification and testing including showing or discussing the use of external feedback to inform the decisions for the features and improvements of the next iteration
* showed computational thinking and problem breakdown not just the solutions
* clearly demonstrated computational thinking in flowcharts and pseudocode
* explained both the ethics and research (not just the research) — students voice/opinion came through in the files submitted
* sourced their research findings from a variety of sources, including primary sources, substantiating their key finds and referencing their work appropriately
* when analysing data, provided a hypothesis and pairing the interpretation of data with research to discuss reasons as to why or why not the hypothesis was supported
* task design allowed multiple opportunities for students to plan and write original code. Students were given an opportunity to write code at a complex level should their skill set allow
* for collaborative tasks, students included discussions about how they worked and contributed towards the work within their group and provided documentation in forms such as: meeting minutes, file sharing directory screenshots, team timelines, program version commits (GitHub)
* used complex structures in their code (such as nested loops and the manipulation of arrays) and explicitly discussed their use of these constructs in their multimodal presentations
* addressed innovation explicitly in their multimodal presentations by pointing out any innovative features and explaining why it was innovative.

The less successful responses commonly:

* showed research, but no ethics, i.e. therefore, students were only able to meet part of the criteria, not all. Some student responses talked about the positives and negatives of technology, but not the ethical considerations
* researched too many topics superficially, not one topic in depth, using only a few secondary sources in their research, did not substantiate their findings and/or did not reference their work appropriately
* duplicated documents that were shared across several students, each submission of work needs to be that of individuals own contribution and evidence
* did not clarify their contribution to collaborative tasks
* provided a very brief overview of each students’ role in a project (usually in the form of a table) without providing any further explanation or convincing evidence of their own contribution
* did not demonstrate the validation of testing of code
* code was not readable, diagrams were unclear. Student’s voice needs to be clear and audible
* work submitted was non-multimodal evidence
* missing tasks, students should have four tasks provided
* lacked design abstraction and did not identify core concepts
* neglected to show evidence or research, such as in-line citations and bibliography
* task design limited student achievement at highest grade bands
* tasks included scaffolded code provided by teacher so each student had very similar work/program backbone which is then not the students own work and cannot be assessed as such
* task design did not allow adequate room for students to develop their own computational thinking/algorithm design
* limited task design that gave students no opportunity to innovate
* minimal to no evidence provided of own role in group tasks
* overused external libraries as main features of the program which minimised the demonstration of Computational Thinking performance standards
* web programs built using just CSS and HTML without JavaScript or a server-side processing do not satisfy the programming requirements nor the Computational Thinking requirements
* tasks spanned too many performance standards and did not spread them out appropriately between the four assessments
* attempted to force Data Analysis into every task taking away from other Performance Standards
* reproduced a tutorial, without adding their own innovative features
* evaluated only the benefits of their solution, without considering its limitations (such as whether people would be able to use the solution)
* programming skills/concepts demonstrated were very basic — example, students only used simple IF statements.

Assessment Type 2: Collaborative Projects

Students apply their learning about iterative project development to create a digital solution through a collaborative project. Each student presents individual evidence of their contribution to the project.

Students create:

* A digital solution (product, prototype, or proof of concept). When submitted should be no more than 1 GB and must be presented in a digital or multimodal form.
* An explanation of the solution with an individual evaluation of each student’s own role in developing the solution, and of the effectiveness of the collaborative working process. The explanation and evaluation should be oral and multimodal and be recorded.

The teacher assesses each student’s individual contribution to the digital solution, explanation, and evaluation.

For this assessment type, students provide evidence of their learning primarily in relation to the following assessment design criteria:

* computational thinking
* development and evaluation.

The more successful responses commonly:

* clearly showed deconstruction of problem and computational thinking before solution was developed
* emphasised key parts of code rather than speaking through what each type of variable was and what it held
* demonstrated iterative development and testing
* had completed prototypes/products which were shown to be working
* had open ended tasks which allowed the students to define their own problem
* had a clear outline of what the individual role students had in the collaborative project
* provided evidence of collaborating with, and presenting to, a target audience
* videos and evidence presented by individual student
* individual student videos clearly identify and demonstrate effectiveness and own role in the collaborative project
* students included discussions about how they worked with their group and provided documentation in forms such as: meeting minutes, file sharing directory screenshots, team timelines, program version commits (GitHub)
* well organised groups were able to allocate a specific part of the program to each individual student which enabled them to meet all required performance standards
* addressed all assessed performance standards
* differentiated between explanation of code and demonstration of the program
* had a clear evaluation section
* students were able to clearly refer to a client they were producing a product for in their presentation.

The less successful responses commonly:

* showed limited computational thinking
* collaboration roles of students not clear in the evidence provided
* no supporting evidence of collaboration
* did not clearly identify student’s own roles and contributions and did not stick to the prescribed time limit stated in the subject outline of maximum 5 minutes
* multiple students in the same video evidence provided. Evidence should be individualised
* not contributing in any way to creating code (computational thinking or actual software programming), hence student could not show evidence of CT4: Application of skills and processes to develop solutions to problems of interest and DE3: Evaluation of the effectiveness of a digital solution or prototype
* had a pre-defined problem as a part of the task which limited the scope for students to define and solve issues – no client conversation and/or group discussion demonstrated to identify scope and issues to then refine a problem
* did not mention, or provide evidence of, having worked with a target audience
* provided very little, if any, evidence of Computational Thinking Skills; the focus was purely on Development and Evaluation.

Overall comments

* Students should not speed up presentations to fit within time limits or talk so fast they are difficult to understand. Evidence needs to be clear and audible.
* Some LAPs and tasks were missing required specific features of the assessment design criteria (e.g. DE4: Explanation, with supporting evidence, ow own role in and contribution to projects, should be assessed in at least one of the four Project Skills Tasks, but was absent from both the LAP and tasks).
* Assessments that only use HTML and CSS without a programming language (GPL) such as JaveScript/Python are not appropriate. HTML and CSS are scripting languages, not programming languages.
* Drag and drop programming languages such as Scratch should not be used at this level.
* When submitting student’s files, it is important to ensure that students have used the recommended free text in the approved online submission formula to identify what each file is and how it links to specific tasks. Required submission file saving convention: 123456R-2XXX20-ATX-**free text**
* 123456R-2DGT20-AT1 — ***Task 1*** ***Collaboration Task***
* 123456R-2DGT20-AT1 — ***Task 2*** ***Analysing Simple and Complex Data Sets***
* 123456R-2DGT20-AT1 — ***Task 3*** ***Programming Skills***
* 123456R-2DGT20-AT1 — ***Task 4*** ***Research and Ethics.***

External Assessment

Assessment Type 3: Individual Digital Solution

Students apply iterative project techniques to independently identify, deconstruct, and solve a problem of interest by creating and evaluating a digital solution or prototype.

The problem should be chosen by, and be of interest to, the student. The problem should be manageable and have sufficient complexity to enable the student to achieve at the highest level. The problem may take as a starting point an aspect of a problem or solution created for Assessment Type 1 or Assessment Type 2 but must not repeat work already submitted for assessment.

In creating their digital solution or prototype, students should be mindful of any ethical considerations.

The solution or prototype should include:

* original source code and/or adapted code displaying selection, repetition, and sequencing, accompanied by design comments
* algorithm design
* graphical user interface and/or instructions for use.

The solution or prototype must be supported by a designer’s statement that discusses:

* the effectiveness of the solution or prototype
* a feature or features that could be considered innovative in solving the problem.

The individual digital solution and supporting documentation (code, design comments, graphical user interface and/or instructions for use) should be submitted in multimodal form. The digital solution or prototype should be no more than 1 GB.

Please note: In 2021 there has been an update to the subject assessment advice where there is now an additional time limit of 15 minutes maximum allowed to the multimodal evidence.

The designer’s statement should be a maximum of 3 minutes if oral, 500 words if written, or the equivalent if multimodal.

The following specific features of the assessment design criteria for this subject are assessed in Assessment Type 3: Individual Digital Solution:

* computational thinking — CT1, CT2, CT4
* development and evaluation — DE1, DE2, DE3.

The more successful responses commonly:

Choosing the Problem

* identified a genuine problem which the digital solution solved, and this remained a recurring theme throughout the iterative process.
* students had a real interest in the problem they were trying to solve
* students engaged genuinely with the community to find stakeholders, gather feedback at multiple points in the development cycle, and reflect on how they have used this feedback to improve their solution
* consulted stakeholders and included discussion before and after the program was written to show if the problem the Digital Solution aims to solve had been addressed.

Iterative Development

* consistently applied iterative development processes, and highlighted changes through testing and modification. Showing evidence at each iteration.

Computational Thinking/Planning

* showed evidence of Computation Thinking which included complex algorithmic design and adjustments where necessary
* showed more than one method of computational thinking (e.g. designs/concept drawings, flowcharts and other diagrams for modular design, pseudocode) and explained the thinking behind these
* provided supporting documentation that showed computational thinking, planning and design.

Code/Testing/Validation/Evaluation

* had video walkthrough evidence of the solution with voice over explanation
* showed evidence of the code with comments
* demonstrated testing and validation of the code
* evaluated the digital solution as opposed to personal performance
* critically evaluated the effectiveness of their solution explaining strengths and weaknesses
* were able to identify innovative features within their solution and clearly explained how the product was innovative to them
* evaluated the overall effectiveness of the solution rather than parts, giving both positives and places for improvement
* highlighted key elements of code, with clear explanation and elaboration from the student
* gave meaningful and honest evaluation that reflected an understanding of the technical issues, social issues and usability issues
* showed complex programming constructs, including nested loops, functions, arrays and file input/output.

The less successful responses commonly:

* had voice overs which didn’t add more value (regurgitating what has already been provided in written text)
* over emphasis of the aesthetic of the digital solution rather than the computational/algorithmic design
* applied instances of a class and calling methods which were in-built, not created by the student
* evidence provided did not show work which constitutes 9 weeks, worth of development
* solutions that were websites with only HTML and CSS and no actual programming language used were not able to demonstrate programming concepts
* were structured like the old Information Technology course, showing validation and testing with limited discussion of innovations and iterative development
* limited or extremely brief evaluations. Designer’s statement was only a few sentences. Subject outline clearly states should be a maximum of 3 minutes if oral, 500 words if written or multimodal equivalent.
* although it was possible to show innovation, many students who created solutions that were databases (VBA) were not as successful, particularly with the innovation and initiative in design
* showed only the final digital solution and had no or very limited evidence of the iterations. Although timelines showed planning, many were not followed, and it was difficult to see if the iterative development had taken place
* showed no evidence of the digital solution, even in a prototype form
* evaluations that did not evaluate the ‘effectiveness of the digital solution or prototype’. Sometimes this was just recounting what they did, sometimes evaluating their walk-through video, and sometimes evaluating a part or a few parts of the system
* did not show the code at all, or included a copy of the code in a PDF or Word doc with no explanation
* discussed general purpose of things/coding conventions but did not connect this to their own solution. i.e. discussed the importance of having a flowchart but did not show one for their system
* did not discuss why the solution was innovative to them or were not able to identify innovative features in their solution
* large focus on ethics when this is not assessed in Assessment Type 3
* recreations of existing products with limited changes
* the chosen problem was not an interest to the student and/or was a problem given to them by the teacher
* did not analyse the problem or show iterative development in the development of their solution
* did not demonstrate evidence of computational thinking through adequate planning of their solution
* evaluated the process and their abilities rather than the product's user-friendliness, effectiveness, portability, security, privacy, reliability, etc.

General subject issues/advice

* On occasion, too many files were submitted many which were not necessary as all the evidence was already provided in the first video. The additional PowerPoint, PDF and other files that were also submitted were just duplicates of information what was included in the video.
* Adequate evidence below for Assessment Type 3: Individual Digital Solution (AT3):
* two files maximum (mp4 or ppt) video evidence of planning, computational thinking and design, iterative development and the code of the solution including the solution working (if not a prototype) — maximum file size no more than 1 GB and 15 minute time limit must not be exceeded
* Designers Statement — stating the effectiveness of the solution and innovative features. — maximum 3 minutes oral or 500 words total.
* The Individual Digital Solution should be an approximate 9-week project and all students should be given this amount of time
* Videos:
* do not need to be polished, but students should be prepared before filming (check diagrams etc. so they don’t have to re edit them while filming or after
* some students set their videos to music and had very fancy effects. Sometimes this came at the expense of sections where they could obtain marks
* students do not need to spend time in the video telling the marker to read the documentation, they will.
* Try to keep to a *maximum* of 15 minutes for a walk through, any more is unnecessary and too long.
* HTML and CSS are not programming languages. To meet the subject requirements there must be a programming language accompanying the creation of a website such as JavaScript, PHP etc.
* Many evaluations focused on what the student should have done rather than evaluating the effectiveness of the features they did do
* In the individual Digital Solution, students must reflect on how their solution is innovative and describe this in the designer’s statement.