2022 Digital Technologies Subject Assessment Advice

Overview

Subject assessment advice, based on the 2022 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.

Across the Assessment Types for this subject, students can present their responses in oral or multimodal form, where 6 minutes is the equivalent of 1000 words. Students should not speed-up the recording of their videos excessively in an attempt to condense more content into the maximum time limit.

From 2023, if a video is flagged by markers/moderators as impacted by speed, schools will be requested to provide a transcript and markers/moderators will be advised to mark/moderate based on the evidence in the transcript, only considering evidence up to the maximum word limit (e.g. up to 2000 words for AT3).

If the speed of the recording makes the speech incomprehensible, it affects the accuracy of transcriptions and it also impacts the ability of markers/moderators to find evidence of student achievement against the performance standards.

School Assessment

Teachers can improve the moderation process and the online process by:

* ensuring no zip files containing the program are included
* avoiding large PowerPoints with embedded videos; screen recording the presentation and submitting a single video file is preferred
* avoid uploading unnecessary PDFs (eg. large files of code where a student has already recorded a multimodal presentation)
* screen record at a high quality to avoid blurry code that is difficult to read
* not using computer generated voice overs which can be difficult to understand and does not allow expression to be heard.

Assessment Type 1: Project Skills Task

Project skills should enable students to create a solution of interest to them. As part of the collaborative task, it is critical that students can showcase their own individual contribution to the project.

Teachers can elicit more successful responses by:

* ensuring all the performance standards are assessed at least once
* providing tasks that allow students to solve problems in unique ways.

The more successful responses commonly:

* clearly showed the student’s computational thinking through evidence (e.g. flowcharts, pseudocode, class diagrams)
* included an iterative project development task that included testing and feedback from clients with modifications made based on feedback
* involved programming elements that had been supported with abstraction and computational thinking in the planning
* utilised datasets that allowed students to show high level skill (i.e. datasets of a considerable size and complexity, used datasets for more than just filtering and graphs).

The less successful responses commonly:

* included tutorials that were followed by students, rather than unique programs created by the student
* focussed on HTML/CSS in mostly website tasks with minimal general-purpose programming language (GPL) used
* did not show evidence of testing or debugging that the student would have had to do
* did not provide evidence of what changes were made based on feedback from clients, or did not provide evidence of client feedback
* focussed on design without providing enough evidence of programming skills
* lacked code explanation.

Assessment Type 2: Group Task

One multimodal submission with a total time of 5 minutes per student should be submitted. Identifying a client and outlining the problem with a showcase of the iterative work undertaken is critical. Finally, the presentation should evaluate the group's work and clearly showcase the individual student contribution to the overall project.

Teachers can elicit more successful responses by:

* ensuring students are individually filmed presenting the project. This is often best done in front of a projector or display screen.

The more successful responses commonly:

* identified a clear client and received feedback from the client throughout the design and implementation phase
* identified and developed their problem from client discussion
* demonstrated their own clear contribution to the submission, showcasing the collaborative work and highlighting their individual contributions
* clearly demonstrated their iterative project development with modifications based on feedback, as well as fixing bugs
* used a general-purpose programming language (GPL) such as C#, Java, Python.

The less successful responses commonly:

* were unclear about what the student’s contribution to the project was after the initial planning stage
* utilised a database solution which required minimal coding by the student
* did not show evidence of presenting to a client
* did not demonstrate that the solution worked, or any code.

External Assessment

Teachers can elicit more successful responses by:

* ensuring students look to solve a problem of interest. The problem identified should be clearly articulated at the beginning of the presentation and deconstructed throughout the presentation.

Assessment Type 3: Investigation

The investigation needs to focus on solving a problem of interest, with the development process clearly shown. A client (real or fictitious) is not necessary for AT3. Students should be encouraged to identify a problem of interest to solve.

Reminder, please take note of de-identifying student materials for AT3 and not including assessment marking or grades.

The more successful responses commonly:

* chose an interesting topic or goal that the student was interested in providing a solution for
* identified the tools used (IDE) and languages within the project (C#, Python etc)
* identified and consistently referenced an original problem and evaluated various stages of the development (iteration) process to address issues
* where possible used an audience to test or provide feedback
* used a general-purpose programming language (GPL) (C#, Python etc) and demonstrated complex coding features, including complex algorithms, nested arrays and multiple linked functions
* used multiple methods to showcase computational thinking including class diagrams, flowcharts and pseudo code
* clearly demonstrated iterative development, highlighting the process of fixing errors and responding to client feedback (if used) in the development process
* referred back to original scope of project within the presentation
* showed a balance between problem deconstruction and the computational logic required to solve the problem (CT1)
* utilised multimodal elements in the presentation to show code snippets and then demonstrate program responses to test different scenarios
* used diagrams to outline the program and show how the code was modularised to run the program
* demonstrated the final product with appropriate voice over, after a full development cycle process was shown
* demonstrated clear in-depth evaluations of the features of the program, referring to the original problem identified.

The less successful responses commonly:

* didn’t clearly identify a problem, which resulted in an unclear deconstruction phase to show how the problem would be addressed
* did not use a GPL
* only used simple loops, conditional statements and functions
* only used mind maps to demonstrate their computational thinking, rather than more complex pseudocode or detailed flowcharts
* included a PDF with all the programming but did not provide an explanation
* read code line by line, rather than demonstrating their understanding of the code
* focussed on databases
* did not show the process to arrive at a final product
* only showed code but no final product
* only showed development but no coding or final product
* did not identify innovative features
* used applications such as Unreal Engine or Construct3 that only allowed for drag and drop block based programming, rather than text programming.