# Government of South Australia LogoSACE Board Logo2023 Digital Technologies Subject Assessment Advice

Overview

Subject assessment advice, based on the 2023 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 2024, 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 that only approved video files (See ‘Accepted File Formats on the SACE Website’) or PPT files are uploaded, i.e. no Zip files containing code or digital solutions
* avoiding large PowerPoints with embedded videos; screen recording the presentation and submitting a single video file is preferred
* avoid uploading unnecessary content which is not part of the presentation or designers statement, such as PDFs containing large files of code as the student should have already recorded a multimodal presentation
* 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 solutions 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
* ensure student use an appropriate object-oriented, general purpose programming language, which is crucial for demonstrating complex coding concepts.

*The more successful responses commonly:*

* clearly showed the student’s computational thinking through evidence (e.g. flowcharts, pseudocode, class diagrams) to a digital problem (e.g. making a game or app)
* included an iterative project development task that included testing and feedback from clients and/or users with modifications based on feedback
* involved programming elements that has been supported with abstraction and computational thinking in the planning
* clearly demonstrated and explained the code produced
* for the collaborative task, it was clearly evident what role that student took, and their contribution to the task was explained with clear evidence of how they collaborated
* utilised datasets that allowed students to show high level skills (i.e. datasets of a considerable size and complexity, used datasets for more than just filtering and graphs)
* evaluated the effectiveness of the digital solution including both the negatives and positives and did not just evaluate their performance in the task
* innovative features were highlighted and discussed.

*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 most website tasks with minimal object-oriented programming language 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 and/or users
* focused on design without providing enough evidence of programming skills
* lacked code explanation
* did not use an appropriate object-oriented programming language (e.g. used Databases with VB or Construct3) so was not able to show understanding of complex coding
* digital solution was not complex
* no innovative features were mentioned.

Assessment Type 2: Collaborative Project

The collaborative project must Identify a client and outline the problem, with a showcase of the iterative work undertaken. The presentation should evaluate the group’s work and clearly showcase the individual student contribution to the overall project.

The AT2 Collaborative Project requires two multimodal submissions per student.

The first, with a total time of 5 minutes per student, should be a live recorded presentation of the student, to a client or users, where they explain the digital solution (product, prototype, or proof of concept), their project evaluation and their role in and contribution to the project, supported by evidence.

The second multimodal submission, with a max 1GB recording file, or 2 to 3 minutes long, should be a short walk-through video of the code, and a demonstration of the working final solution.

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 a display screen
* ensuring students have a client for their problem.

*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 an object oriented, general-purpose programming language (GPL) such as C#, Java, Python
* the problem was big enough to be broken down into enough parts for each student to do and show detailed evidence of the whole range of performance standards assessed in the AT2 task
* Innovative features were highlighted.

*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
* the problem was provided by the teacher, not necessarily of interest to the students, and hence did not employ a real client’s situation
* no innovative features were highlighted.

# External Assessment

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.

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 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, object-oriented programming language (C#, Python etc) and demonstrated complex coding concepts, including complex algorithms, nested arrays and multiple linked functions
* used multiple methods to showcase computational thinking including class diagrams, flowcharts and pseudocode
* clearly demonstrated iterative development, highlighting the process of testing, fixing errors and responding to client feedback (if used) in the development process
* referred back to original scope of the project within the presentation
* flowcharts were of top-level abstraction (e.g. showing how the program worked overall) as well as showing the flow of specific functions that would need to be coded
* 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 evaluation of the features of the program, referring to the original problem identified
* clearly highlighted and discussed innovative features
* addressed all performance standards
* digital solutions were complex
* evaluations showed lots of reflecting on the success of all the features, outlining how it meets clients needs, and considering future features and versions of the software.

*The less successful responses commonly:*

* digital solution was not complex and did not allow students to demonstrate in-depth understanding of coding concepts
* focused mainly on databases, HTML and CSS and little on a programming language
* did not highlight or mention any innovative features, or if they were, there was no explanation as to why they were innovative
* flowcharts and pseudocode, while evident, were not complex and showed only the main concepts, not how functions were to be created and coded
* little evidence of iterative development. Showed only some bug fixes or a time plan, no evidence of modification based on feedback or testing
* outlined difficulties with learning as opposed to logical issues within development that needed to be solved
* only defined what computational thinking is, and the different iterative models available, which does not contribute to any performance standards
* over-emphasised login systems as the main feature of a product rather than the more complex features of the main program
* did not show any computational thinking
* focused mostly on HTML and CSS with little application code present in the digital solution
* coding was very simplistic with simple loops, conditional statements and functions
* read code line by line, rather than demonstrating their understanding of the code
* did not show the final product working.

General

The Subject Adjustments must be followed correctly: if the option of only undertaking 3 AT1 tasks has been employed, then the maximum combined duration of the AT1 tasks is 15 minutes maximum. However, if the usual 4 AT1 tasks are employed, then the maximum of 20 combined minutes remains.