# VCE Computing: Software Development: Administrative information for School-based Assessment in 2019

## School-assessed Task

The School-assessed Task (SAT) contributes 30 per cent to the study score.

Teachers will provide to the Victorian Curriculum and Assessment Authority (VCAA) a score against each criterion that represents an assessment of the student's level of performance for Unit 3 Outcome 2 and Unit 4 Outcome 1. The recorded scores must be based on the teacher's assessment of the student's performance according to the criteria on pages 5−12. This assessment is subject to the VCAA's statistical moderation process.

The 2019 VCE Computing: Software Development assessment sheet on page 15 is to be used by teachers to record the SAT score. Student scores for each outcome will be submitted separately via VASS. The score for Unit 3 Outcome 2 must be submitted no later than 14 June. The score for Unit 4 Outcome 1 must be submitted no later than 6 November.

The performance descriptors for the assessment criteria are published annually on the Computing: Software Development study page of the VCAA website. Details of authentication requirements and administrative arrangements for School-assessed Tasks are published in the *VCE and VCAL Administrative Handbook 2019*.

The School-assessed Task has five components. They relate to:

- Unit 3 Outcome 2 (three components)
- Unit 4 Outcome 1 (two components).

# Unit 3

## Analysis and design

## Outcome 2

On completion of this unit the student should be able to analyse and document a need or opportunity, generate alternative design ideas, represent the preferred solution design and formulate a project plan for creating the solution.

### Nature of task

- an analysis that defines the requirements, constraints and scope of a solution in the form of a software requirements specification
- a folio of two to three alternative design ideas and the detailed design specifications of the preferred design
- a project plan (Gantt chart) indicating times, resources and tasks.

### Scope of task

#### Analysis: Software requirements specification

Students must identify a need or opportunity for a software solution created through the use of a programming language. It needs to be a real-world problem (need) or opportunity, not a problem determined by the teacher. Students should engage in the activities of the Analysis stage of the problem-solving methodology and document their findings in the form of a software requirements specification (SRS). The SRS should be assembled as a formal document that addresses the purpose and environment (audience and technical characteristics) of the software solution and details of the solution requirements, including constraints and scope. Typically the analysis tools (use case diagram, context diagram, data flow diagram) would form an appendix to the SRS. There is no prescribed template for the SRS, hence flexibility exists in its presentation.

Students acquire data in order to carry out their analysis, and the extent of data (breadth/depth) collected by students is dependent on the nature of the problem or opportunity. When describing the technical environment, students can couch this in terms of the digital system(s) that support the solution.

Teachers should approve each student's need or opportunity before they conduct an in-depth analysis. The performance descriptors at the highest level for criteria 2 and 3 should be used to help make this decision.

#### Folio

The folio will comprise design ideas, student-generated evaluation criteria and the detailed specifications of the preferred design.

Students should generate two or three alternative design ideas for their solution. They should take a design-thinking approach when generating different ideas for further development (see Study Design Glossary, page 11). Where appropriate, the design ideas should consider both appearance and functionality. These ideas do not have to be detailed – they represent broad ideas of key appearance features and functions of possible solutions. Students apply their criteria to select the

design idea that will be developed into a detailed design for the solution, using a set of tools, methods and techniques. The design should be able to meet the user requirements stated in the SRS. The evidence from this task is assessed through criterion 4.

### Project plan

Students prepare a project plan, in the form of a Gantt chart, for the full implementation of the problem-solving methodology stages (analysis, design, development and evaluation) and activities, (See Study Design, page 14) hence the plan incorporates the requirements of Unit 4 Outcome 1. Students should only be assessed on their ability to document the key concepts and processes relevant to their project; they must not be assessed on their software skills. Students should take account of any assessment dates and other deadlines provided by the teacher; however, all other scheduling must be determined by the students. As the project progresses in Unit 3, students should be recording any adjustments to their original plan.

Students are strongly urged to prepare their plan near the beginning of their project so they can incorporate all stages of the problem-solving methodology. The evidence from this task is assessed through criterion 1.

# Unit 4

## Software solutions

### Outcome 1

On completion of this unit the student should be able to apply stages of the problem-solving methodology to create a solution using a programming language that fulfils identified requirements and assess the effectiveness of the project plan in monitoring progress.

### Nature of task

- a software solution that meets the software requirements specification and the results of the useability test
- an assessment of the extent to which the project plan (Gantt chart) assisted in monitoring project progress in one of the following:
  - a written report
  - an annotated visual plan.

### Scope of task

#### Software solution

Students use the detailed design produced in Unit 3 as the basis for developing a software solution. Students should apply the necessary features of the programming language published in the *VCAA Bulletin* and also published on the Computing: Software Development study page as the document labelled Software Development: Programming requirements. If the detailed design generated and assessed in Unit 3 is incomplete or contains significant errors and would prevent demonstration of the highest level of achievement on the relevant criteria, students have the opportunity to make adjustments to their design. Teachers can provide feedback on the quality of

the designs but the adjustments must be student initiated, not teacher directed. The modified design is not reassessed but this opportunity prevents negative consequential effects in the development stage.

Students must present evidence of the useability test that they designed, conducted and documented. Two or more ‘users’ other than the student must undertake the test. The useability test should address the core features of the solution and these would be reflected in the SRS. Students make any necessary adjustments to their solution.

In accordance with the problem-solving methodology (see Study Design, page 14) evaluation typically takes place after the client has implemented the solution, but this is not practically possible for this task. However, students are required to propose an evaluation strategy assuming implementation of their solution, as well as undertaking an individual evaluation of the solution (student-generated evaluation) using the criteria developed in Unit 3. The evidence from this task is assessed through criteria 5, 6, 7 and 8.

### Assessment of project plan

Students are required to explain, either through annotations or a report, how their plan assisted in monitoring the progress of the project. Students can use a variety of ways of showing adjustments on their plan. In addition to acknowledging adjustments, students must explain the usefulness of the plan in monitoring their progress through the stages of the problem-solving methodology. The evidence from this task is assessed through criterion 8.

The following rubric is used to assess student achievement on Unit 3 Outcome 2 and Unit 4 Outcome 1. It should be noted that each piece of evidence in each criterion is not equally weighted.

The criteria identify specific characteristics that are used to judge levels of performance against the outcomes. Performance descriptors describe typical evidence associated with five different levels of performance for a criterion (five levels; 10 marks).

Criteria 1 to 4 relate to Unit 3 Outcome 2.

Criteria 5 to 8 relate to Unit 4 Outcome 1.

| VCE Software Development: School-assessed Task 2019 | | | | | | |
|---|---|---|---|---|---|---|
| **Assessment criteria** | **Levels of Performance** | | | | | |
| | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **1. Understanding of project management concepts and processes** | | Includes some stages of the problem-solving methodology in the project plan and identifies limited activities.<br><br>Constructs a project plan using software that identifies limited milestones (concepts), tasks and time allocations (processes).<br><br>Omissions, errors and brevity reduce the effectiveness of the chart to adequately monitor progress. | Prepares a Gantt chart that includes most stages of the problem-solving methodology and identifies some activities.<br><br>Prepares a Gantt chart using software that identifies some student-determined milestones, few dependencies (concepts) and a range of tasks and time allocations (processes).<br><br>Some omissions and errors reduce the effectiveness of the chart to adequately monitor progress. | Prepares a Gantt chart that includes all stages and some of the activities of the problem-solving methodology for both parts of the project.<br><br>Prepares a Gantt chart using software that identifies some relevant student-determined milestones and key dependencies (concepts), and some of the key tasks, sequencing of tasks and time allocations (processes).<br><br>Generally effective chart with minimal errors to monitor progress. | Prepares a Gantt chart that documents all stages and most of the activities of the problem-solving methodology for both parts of the project.<br><br>Prepares a Gantt chart using software that identifies most of the relevant student-determined milestones, key dependencies (concepts) and key tasks, sequencing, resources and time allocation (processes).<br><br>Mainly accurate and coherent chart contributes to a project management plan to monitor progress. | Prepares a Gantt chart that documents all stages and activities of the problem-solving methodology for both parts of the project.<br><br>Prepares a Gantt chart using software that identifies all relevant project management milestones, key dependencies (concepts) and key tasks, sequencing, resources and time allocation (processes).<br><br>The accuracy, completeness and coherence of the chart contribute to an effective project management plan to monitor progress. |
| | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |

| VCE Software Development: School-assessed Task 2019 | | | | | | |
|---|---|---|---|---|---|---|
| **Assessment criteria** | **Levels of Performance** | | | | | |
| | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **2. Skills in using analytical tools to depict relationships between data, users and digital systems** | 0 ❑ | Identifies limited data for analysis and uses few appropriate collection methods and techniques.<br><br>Uses very limited features in the selected analytical tools.<br><br>Depicts limited relationships between data, users and digital systems in an analytical tool. Omissions and errors result in misinterpretations of these relationships.<br><br>Applies very few correct techniques in the analytical tools.<br><br>1 ❑ 2 ❑ | Identifies some relevant data for analysis and uses some appropriate collection methods and techniques.<br><br>Uses some features of the selected analytical tools.<br><br>Depicts some of the relationships between the data, users and digital systems in some analytical tools. Errors and inconsistencies result in some misinterpretations of relationships.<br><br>Applies some correct techniques in the analytical tools.<br><br>3 ❑ 4 ❑ | Identifies adequate and mainly relevant data for analysis and uses some appropriate collection methods and techniques.<br><br>Uses most features of selected analytical tools.<br><br>Depicts accurately most of the relationships between the data, users and digital systems in most analytical tools. Errors, omissions or inconsistencies result in minor misinterpretations of relationships.<br><br>Applies most techniques correctly in the analytical tools, however, some inconsistencies exist.<br><br>5 ❑ 6 ❑ | Identifies sufficient relevant data for analysis and uses a range of appropriate collection methods and techniques.<br><br>Uses all features of the selected analytical tools, with minor errors.<br><br>Depicts accurately all of the relationships between the data, users and digital systems in all analytical tools. Minor errors do not affect interpretation of relationships.<br><br>Applies most techniques correctly and consistently in the analytical tools.<br><br>7 ❑ 8 ❑ | Identifies a complete and relevant set of data for analysis and uses a wide range of appropriate collection methods and techniques.<br><br>Uses accurately all features of the selected analytical tools.<br><br>Depicts accurately all of the relationships between the data, users and digital systems in all analytical tools.<br><br>Applies all techniques correctly and consistently in all analytical tools.<br><br>9 ❑ 10 ❑ |

| VCE Software Development: School-assessed Task 2019 | | | | | | |
|---|---|---|---|---|---|---|
| **Assessment criteria** | **Levels of Performance** | | | | | |
| | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **3. Interpretation of data to identify and document the software requirements specification** | | Makes some observations about the data represented in the analytical tools.<br><br>Lists limited solution requirements (not classified), constraints and scope.<br><br>Outlines limited aspects of the technical environment and intended audience of the solution. A range of errors affect the interpretation of the solution requirements.<br><br>Omissions, errors and brevity of the documentation reduce its usefulness for design and development purposes. Only a partial solution could be achieved. | Draws some conclusions from the data represented in the analytical tools, however, some conclusions cannot be substantiated.<br><br>Outlines some appropriate solution requirements, with limited classification, constraints and scope.<br><br>Describes some aspects of the technical environment and intended audience of the solution. Some errors affect the interpretation of the solution requirements.<br><br>Some omissions, errors and generalities of the documentation reduce its usefulness for design and development purposes. | Draws mainly logical conclusions from data represented in the analytical tools, however, some conclusions cannot be substantiated.<br><br>Describes specifically some appropriate functional and non-functional requirements, constraints and scope.<br><br>Describes using study-specific language, some key aspects of the technical environment and intended audience of the solution. Some errors of fact exist but do not cause meaningful ambiguities.<br><br>Assembles key components of the documentation. Minor errors or omissions reduce its usefulness in fulfilling its purposes. | Draws a set of logical conclusions with most being consistent with the data represented in the analytical tools.<br><br>Describes specifically most of the appropriate functional and non-functional solution requirements, constraints and scope.<br><br>Describes using study-specific language, all key aspects of the technical environment and intended audience of the solution. Mainly accurate facts.<br><br>Assembles logically the key components of the documentation. Minor errors or omissions exist but do not reduce its usefulness in fulfilling its purposes. | Draws a complete set of conclusions that are connected to, and consistent with, the data represented in the analytical tools.<br><br>Describes specifically and accurately all the functional and non-functional solution requirements. All constraints and scope are logically connected to the solution requirements<br><br>Using study-specific language, describes precisely and accurately all relevant aspects of the technical environment and intended audience of the solution.<br><br>Assembles all the documentation clearly, accurately and logically. It is fit for purpose. |
| | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |

| VCE Software Development: School-assessed Task 2019 | | | | | | |
|---|---|---|---|---|---|---|
| **Assessment criteria** | **Levels of Performance** | | | | | |
| | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **4. Skills in generating design ideas and designing preferred solutions** | | Lists a narrow set of criteria for evaluating design ideas and solutions. Omissions reduce the capacity of the preferred design to meet requirements.<br><br>Generates two design ideas that have minor differences in their appearance or functionality.<br><br>Selects a preferred design idea for further development based on personal preference with limited acknowledgement of the solution requirements.<br><br>Expresses the preferred design using limited and incomplete methods. Minimal consideration evident of relevant design factors. Logical errors exist. | States some criteria that enable a partial evaluation of design ideas and solutions.<br><br>Generates two or three design ideas that are slight modifications of each in their representation of the solution's appearance and functionality.<br><br>Selects a preferred design idea for further development based on limited criteria relevant to the solution requirements.<br><br>Expresses the preferred design using some appropriate methods. Some consideration of design factors is evident with some errors of logic and omissions that impact the capacity of the designs to be developed. | States a range of criteria for evaluating the capacity of design ideas and solutions to meet functional and non-functional requirements.<br><br>Generates two or three design ideas that represent some feasible alternatives to the solution's functionality and appearance.<br><br>Selects a preferred design idea for further development based on some relevant criteria to the solution requirements.<br><br>Expresses the preferred design using some appropriate methods. Consideration of relevant design factors is evident with minimal errors. | Specifies a complete set of criteria for evaluating alternative design ideas and solutions to meet functional and non-functional requirements.<br><br>Generates two or three design ideas that are feasible and clearly differ in their representation of the solution's functionality and appearance.<br><br>Selects a preferred design idea for further development based on some relevant criteria related to functional and non-functional solution requirements.<br><br>Expresses preferred design using appropriate methods. Designs demonstrate consideration of most relevant design factors. Minor errors do not reduce the capacity of the designs to be developed. | Specifies a comprehensive set of criteria for evaluating alternative design ideas and the efficiency and effectiveness of the solution.<br><br>Generates two or three distinctive design ideas that are feasible and original representations of the solution's functionality and appearance.<br><br>Selects a preferred design idea for further development based on a complete set of criteria.<br><br>Expresses accurately and completely preferred design using appropriate methods. Designs are detailed and fully demonstrate consideration of relevant design factors. |
| | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |

| VCE Software Development: School-assessed Task 2019 | | | | | | |
|---|---|---|---|---|---|---|
| **Assessment criteria** | **Levels of Performance** | | | | | |
| | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **5. Skills in using a programming language to develop a software solution that meets specific needs or opportunities** | | Applies limited processing features of the language to develop a partial solution.<br><br>Writes limited internal documentation with minimal formatting.<br><br>Applies limited data validation techniques to check the reasonableness of some input data.<br><br>Limited evidence of an algorithm in the solution. | Applies some processing features of the language to develop a solution. Inaccuracies and omissions affect the operation of the solution. Applies inconsistently some coding conventions.<br><br>Writes some formatted internal documentation.<br><br>Applies some relevant data validation techniques to check the reasonableness of most input data.<br><br>Writes a solution with an algorithm with some level of complexity. | Selects and applies a range of processing features of the language to develop a solution. Some errors of correctness or completeness exist. Coding conforms to some accepted conventions.<br><br>Writes some formatted internal documentation with relevant program comments, however, inconsistencies exist.<br><br>Applies efficiently and effectively some relevant data validation techniques to check the reasonableness of input data.<br><br>Documents a solution with a mostly complex algorithm. | Correctly selects and applies a wide range of relevant processing features of the language to develop a solution. Minor errors exist. Coding conforms to most accepted conventions.<br><br>Writes internal documentation that contains relevant program comments and is formatted.<br><br>Applies efficiently and effectively most relevant data validation techniques to check the reasonableness of input data<br><br>Documents a solution with a complex algorithm. | Correctly selects and skillfully applies an extensive range of relevant processing features of the language to develop a correct solution. Coding conforms to all accepted conventions.<br><br>Writes clearly internal documentation that is comprehensive, contains relevant program comments and is well formatted.<br><br>Applies efficiently and effectively all relevant data validation techniques to check the reasonableness of input data.<br><br>Documents the efficient use of a complex algorithm in the solution. |
| | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |

| VCE Software Development: School-assessed Task 2019 | | | | | | |
|---|---|---|---|---|---|---|
| **Assessment criteria** | **Levels of Performance** | | | | | |
| | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **6. Skills in organising and managing data and files.** | 0 ❑ | Identifies limited ways of organising files or data to allow access.<br><br>Organises and manipulates limited data through the use of few data structures.<br><br>Applies few procedures or techniques to secure files or enhance access to required files.<br><br>1 ❑ 2 ❑ | Identifies some ways of organising files or data to allow efficient access.<br><br>Organises and manipulates some data through the use of some data structures. Inaccuracies exist.<br><br>Applies some general procedures and techniques to handle and manage some files. Errors expose potential security vulnerabilities and problems accessing required files.<br><br>3 ❑ 4 ❑ | Identifies a logical plan for organising files or data to allow efficient and secure access.<br><br>Organises and manipulates some data through the use of some appropriate data structures.<br><br>Applies some feasible procedures and techniques to handle and manage most files. Some errors expose potential security vulnerabilities or problems accessing some files.<br><br>5 ❑ 6 ❑ | Identifies a systematic and logical plan for organising files or data to allow efficient and secure access.<br><br>Organises and manipulates most data through the efficient and effective use of mostly appropriate data structures.<br><br>Applies a feasible set of procedures and techniques to handle and manage the security and accessibility of all files. Minor efficiency or effectiveness errors exist.<br><br>7 ❑ 8 ❑ | Identifies a comprehensive, systematic and logical plan for organising files or data to allow efficient and secure access.<br><br>Organises and manipulates all data through the efficient and effective use of appropriate data structures.<br><br>Applies a feasible and comprehensive set of procedures and techniques to efficiently and effectively handle and manage the security and accessibility of all files.<br><br>9 ❑ 10 ❑ |

| VCE Software Development: School-assessed Task 2019 | | | | | | |
|---|---|---|---|---|---|---|
| **Assessment criteria** | **Levels of Performance** | | | | | |
| | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **7. Skills and strategies for testing a software solution to meet a specific need or opportunity** | | Documents limited test data and performs limited tests. Many logic errors are undetected. | Documents generally a small range of test data and applies some suitable testing techniques. Some key logic errors are undetected. | Documents formally a range of test data and applies suitable techniques for detecting most logic errors. | Documents formally a wide range of relevant test data and applies suitable testing techniques. Minor logic errors exist but do not reduce the capacity of the solution to meet its requirements. | Documents formally and clearly a comprehensive range of relevant test data. Applies suitable techniques to detect all logic errors. |
| | | Prepares and conducts for one user, a brief and general useability test. Uses limited testing techniques. | Prepares a general useability test. Conducts the test for one user, using some suitable techniques. Meets user requirements. | Prepares a useability test that is specific to some requirements of the solution. Conducts the test for the required number of users and uses suitable testing techniques. | Prepares a useability test that is appropriate to targeting specific key requirements of the solution. Conducts the test using a range of suitable techniques and meets user requirements. | Prepares a useability test that comprehensively covers all targeted requirements of the solution. Conducts test using a set of suitable techniques and meets user requirements. |
| | | Documents informally limited evidence of the useability test. | Documents general results of the useability test. | Documents accurately most results of the useability test in an organised manner. | Classifies and documents accurately all the results of the useability test. | Documents in detail a logically classified set of accurate results from the useability test. |
| | | Implements few modifications to the solution as a result of useability testing. | Implements some modifications to the solution as a result of useability testing. | Implements most modifications to the solution as a result of useability testing. | Implements key modifications to the solution as result of useability testing. | Implements all necessary modifications to the solution as a result of useability testing. |
| | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |

| VCE Software Development: School-assessed Task 2019 | | | | | | |
|---|---|---|---|---|---|---|
| **Assessment criteria** | **Levels of Performance** | | | | | |
| | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **8. Skills in evaluating the software solution and assessing the effectiveness of the project plan in monitoring progress** | | Lists few approaches for evaluating the solution, if implemented by the user.<br><br>Student evaluation outlines generally how some requirements of the solution are met, making limited reference to the criteria.<br><br>Provides limited evidence of adjustments to the initial plan of the project.<br><br>Refers to limited measures of effectiveness when outlining some general improvements to some parts of the project as a result of using the plan. | Identifies some feasible strategies for evaluating the solution, if implemented by the user.<br><br>Student evaluation describes generally how some functional and non-functional requirements of the solution are met, making some reference to criteria.<br><br>Records some adjustments to the initial plan during the progress of most stages of the project. Some errors exist.<br><br>Describes generally some factors that influence the effectiveness of the project plan.<br><br>Describes how the plan and some of its adjustments assisted in monitoring and improving most stages of the project. | Proposes some feasible strategies for evaluating the efficiency and effectiveness of the solution, if implemented by the user.<br><br>Student evaluation explains in terms of efficiency and effectiveness how some functional and non-functional requirements of the solution are met. Makes reference to most criteria.<br><br>Records most adjustments to the initial plan during the progress of the entire project using some appropriate techniques. Minor errors exist but do not reduce the usefulness of the plan.<br><br>Explains the importance of some factors that influence the effectiveness of the project plan.<br><br>Explains generally how the plan and its adjustments assisted in monitoring the progress of the entire project. | Proposes feasible strategies for evaluating the efficiency and effectiveness of the solution, if implemented by the user.<br><br>Student evaluation explains in terms of efficiency and effectiveness how some specific features of the solution meet most functional and non-functional requirements. Considers all criteria.<br><br>Records correctly adjustments to the initial plan during the progress of the entire project using appropriate techniques.<br><br>Explains clearly the importance of relevant factors that influence the effectiveness of the project plan.<br><br>Compares and explains how the specific adjustments recorded as part of the project plan assisted in monitoring the progress of the entire project. | Proposes a coherent set of feasible strategies for evaluating the efficiency and effectiveness of the solution, if implemented by the user. Strategies align with all criteria.<br><br>Student evaluation explains in terms of efficiency and effectiveness how specific features of the solution meet all functional and non-functional requirements. Makes reference to a coherent and comprehensive set of criteria.<br><br>Records correctly and clearly all adjustments to the initial plan during the progress of the entire project. Applies a range of appropriate recording techniques.<br><br>Explains clearly and coherently the importance of relevant factors that influence the effectiveness of the project plan.<br><br>Reports clearly and comprehensively the usefulness of the initial plan and its adjustments in monitoring the progress of the entire project. |
| | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |

# Authentication of VCE Computing: Software Development School-assessed Task (SAT)

Teachers are reminded of the need to comply with the authentication requirements specified in the Assessment: School-based Assessment section of the *VCE and VCAL Administrative Handbook 2019*. This is important to ensure that 'undue assistance [is] not … provided to students while undertaking assessment tasks'.

Teachers must be aware of the following requirements for the authentication of VCE Computing: Software Development School-assessed Task.

1. The body of work created for the School-assessed Task (SAT) is based on work developed and completed in Unit 3 Outcome 2 and Unit 4 Outcome 1.
2. Teachers are required to fill out the Authentication Record Form and provide the student with feedback on their progress at each observation.
3. Undue assistance should not occur at any time during the development of the body of work, and teachers need to be vigilant. Students are required to demonstrate development of their thinking and working practices. Teachers are reminded that it is not appropriate to provide 'detailed advice on, corrections to, or actual reworking of students' work'.
4. Teachers must sight and monitor the development and documentation of the student's thinking and working practices throughout the unit to authenticate the work as the student's own. Students must acknowledge the source of materials and information used to support the development of their work.
5. Students should be encouraged to complete their work at school. Where students use external service providers, their documentation should demonstrate ongoing progress throughout the SAT.
6. During the generation of the software solution teachers must plan and use observations of student work in order to monitor and record each student's progress as part of the authentication process. Teachers must ensure that all source and reference material, all use of non-school (home, outsourced) resources and any external assistance (for example, tutors) are acknowledged on the authentication form. If a student acknowledges using external resources or receiving external assistance, the teacher should record complete details as an attachment to the Authentication Form.
7. Teachers are reminded that the authentication procedures must be followed for all student work in relation to this SAT. School-based assessment audits include the inspection of authentication records. If authentication records are not provided, the school is automatically audited the following year.

VICTORIAN CURRICULUM AND ASSESSMENT AUTHORITY

# Authentication Record Form: VCE Computing: Software Development SAT 2019

This form must be completed by the class teacher. It provides a record of the monitoring of the student's work in progress for authentication purposes. This form is to be retained by the school and filed. It may be collected by the VCAA as part of its School-based Assessment Audit.

Student name …….. Student No | | | | | | | | | |

School ……

Teacher: ……

| Component of School-assessed Task | Date observed/ submitted | Authentication comments | Teacher's initials | Student's initials |
|---|---|---|---|---|
| **Observation 1: Identification of need or opportunity**<br>Student has identified a real-world problem/opportunity that can be solved through a software solution | | | | |
| **Observation 2: Preparation of project plan**<br>Student has prepared a Gantt chart for the entire project | | | | |
| **Observation 3: Data collection**<br>Student has identified data sources and collection methods for analysis | | | | |
| **Observation 4: Development of software requirements specification**<br>Student has documented the analysis in the form of an SRS | | | | |
| **Observation 5: Folio**<br>Student has developed the folio of design ideas, criteria and detailed preferred design | | | | |
| **Observation 6: Progressive development of software solution**<br>Student has continued to develop the software solution | | | | |
| **Observation 7: Preparation and conduct of useability test**<br>Student has prepared, conducted and documented the useability test | | | | |
| **Observation 8: Evaluation and assessment**<br>Student has submitted the evaluation of the solution and the assessment of the project plan | | | | |

I declare that all resource materials and assistance used have been acknowledged and that all unacknowledged work is my own.

Student signature …… Date ……

**2019**

# Victorian Certificate of Education
## Computing: Software Development Assessment Sheet
## School–assessed Task

STUDENT NAME

This assessment sheet will assist teachers to determine their score for each student. Teachers need to make judgments on the student's performance for each criterion. Teachers will be required to choose one number from 0–10 to indicate how the student performed on each criterion with comments, as appropriate. Teachers then add the subtotals to determine the total score.

STUDENT NUMBER

ASSESSING SCHOOL NUMBER

**Criteria for the award of grades**

**The extent to which the student demonstrates:**

| | | Not Shown (0) | Very Low (1–2) | Low (3–4) | Med (5–6) | High (7–8) | Very High (9–10) |
|---|---|---|---|---|---|---|---|
| 1 | understanding of project management concepts and processes | | | | | | |
| 2 | skills in using analytical tools to depict relationships between data, users and digital systems | | | | | | |
| 3 | interpretation of data to identify and document the software requirements specification | | | | | | |
| 4 | skills in generating design ideas and designing preferred solutions | | | | | | |
| 5 | skills in using a programming language to develop a software solution that meets specific needs or opportunities | | | | | | |
| 6 | skills in organising and managing data and files | | | | | | |
| 7 | skills and strategies for testing a software solution to meet a specific need or opportunity | | | | | | |
| 8 | skills in evaluating the software solution and assessing the effectiveness of the project plan in monitoring progress. | | | | | | |
| | **SUBTOTALS** | | | | | | |

**TOTAL SCORE**

**Performance on Criteria: Teacher's Comments**
You may wish to comment on aspects of the student's work that led to your assessment.

If a student does not submit the School–assessed Task at all, N/A should be entered in the total score box.