VCE Applied Computing: Software Development: Administrative information for School-based Assessment in 2025

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 8−17. This assessment is subject to the VCAA’s statistical moderation process.

The 2025 VCE Applied Computing: Software Development assessment sheet on page 21 is to be used by teachers to record the SAT score. The completed assessment sheet must be made available on request by the VCAA*.*

The mandated assessment criteria are published annually on the Applied Computing: Software Development study page of the VCAA website and notification of their publication is given in the February *VCAA Bulletin*.

Details of authentication requirements and administrative arrangements for School Assessed Tasks are published annually in the [*VCE Administrative Handbook 202*](https://www.vcaa.vic.edu.au/administration/vce-vcal-handbook/Pages/index.aspx)*5*.

The Authentication record form on pages 19–20 is to be used to record information for each student and must be made available on request by the VCAA.

The SAT has eight components. They relate to:

* Unit 3 Outcome 2 (five components)
* Unit 4 Outcome 1 (three components)

Teachers should be aware of the dates for submission of scores into VASS in July and October. These dates are published in the [2025 Important Administrative Dates and 2025 Assessment Schedule](file:///E:\2024\VCAA%20Documents\VCE%20Administrative%20information\vcaa.vic.edu.au\pages\schooladmin\admindates\index.aspx), published annually on the VCAA website.

Unit 3

Software development: analysis and design

Outcome 2

Document a problem, need or opportunity, formulate a project plan, document an analysis, and generate design ideas and a preferred design for creating a software solution.

Nature of task

A brief outlining the proposed solution and a project plan (Gantt chart) indicating tasks, times, milestones, dependencies and the critical path

**And**

Analytical tools that depict the interactions between systems, users, data and networks

**And**

An analysis that defines the requirements, constraints and scope of a solution in the form of a software requirements specification

**And**

A folio of design ideas and evaluation criteria

**And**

Detailed design specifications of the preferred design.

Time allocated should be at least 8–10 weeks of class time.

Scope of task

Development of a brief and project plan

In preparation for the SAT students will need to be able to identify a real-world problem, need or opportunity that can be developed as a software solution for a client or potential users.

Criterion 1 assesses students’ skills in developing a brief and in project management. Students will need to prepare a brief that documents their problem, need or opportunity. Further details regarding the process of developing a brief is in the *Support material*.

Teachers should have discussions with their students regarding their problem, need or opportunity and to have a process for approving the brief before students commence their project plan. Students are encouraged to document their ideas to convince their teacher that they will be able to develop a software solution.

Students will prepare a Gantt chart using an appropriate software tool that documents all the stages and the activities of the problem-solving methodology for Unit 3 Outcome 2 and Unit 4 Outcome 1 (both parts of the SAT).

Students will need to document all the relevant tasks, sequencing, time allocations, milestones, dependencies and critical path.

The evidence from this task is observed through Observation 1 and assessed through Criterion 1.

Once the project plan has been developed, it needs to be monitored, modified and annotated during the life of the project (both Unit 3 Outcome 2 and Unit 4 Outcome 1). It will then be assessed as part of Criterion 10 in Unit 4 Outcome 1.

Documentation of the analysis

Criterion 2 assesses students’ skills in documenting the analysis. Students are required to collect and prepare data for analysis using appropriate data collection methods. The data collected will enable students to document the relevant features of the selected analytical tools and depict the relationships between the data, users and systems. Sample representations of the analytical tools are in the *Support material*. The process of data collection may involve students communicating back-and-forth with their clients or potential users.

The data collected and the analytical tools will assist in the development of a software requirements specification in Criterion 3.

The evidence from this task is observed through Observation 2 and assessed as part of Criterion 2.

Development of a software requirements specification

Criterion 3 assesses students’ skills in documenting a software requirements specification. Students will document the functional and non-functional requirements, constraints and scope as well as the user characteristics and the technical environment for the proposed software solution. The software requirements specification is to be presented as a formal document. Analytical tools developed in Criterion 2 are to be included in the appendix of the software requirements specification. An outline of the content required in the software requirements specification is in the *Support material*.

Students will document evidence of their critical and creative thinking using questions and follow-up questions to clarify the development of the solution requirements as part of the Analysis stage in Criterion 2 and 3. Refer to the Skills underpinning the Analysis stage in the Units 1 to 4: Problem-solving methodology specifications on page 18 of the study design.

The evidence from this task is observed through Observation 3 and assessed through Criterion 3.

Generating design ideas and developing evaluation criteria

Criterion 4 assesses students’ skills in generating design ideas and developing evaluation criteria. Students will generate design ideas for the software solution using a range of ideation tools. Design ideas are to be annotated to explain the appearance and functionality of the software solution. They will develop evaluation criteria for their design ideas and the software solution. Evaluation criteria will reference the functional and non-functional requirements for the design ideas and be used to measure the efficiency and the effectiveness of the software solution in Criterion 10. Students will need to justify which of the elements of the design ideas should be further developed into detailed designs.

The evidence from this task is observed through Observation 4 and assessed through Criterion 4.

Producing detailed designs

Criterion 5 assesses students’ skills in producing detailed designs. Students will produce their detailed designs for the software solution using a range of design tools. They will also document the use of design principles that influence the appearance and functionality of the detailed designs as well as the characteristics of user experience that affect the detailed designs.

Students will document evidence of their critical and creative thinking through the development of design ideas, solution requirements and the detailed designs as part of the Design stage in Criterion 5. Refer to the Skills underpinning the Design stage in the Units 1 to 4: Problem-solving methodology specifications on page 19 of the study design.

The evidence from this task is observed through Observation 5 and assessed through Criterion 5.

Authentication

Teachers should monitor students’ progress on a regular basis and use the Authentication record form to record this information. The authentication process and feedback are clearly identified on this form so that teachers can provide feedback at various stages of the process. It is recommended that students back up files with copies of work in progress using, for example, an external drive, network drive or secure cloud storage.

Issues identified after marking Unit 3 Outcome 2

At the completion of Unit 3 Outcome 2 students may experience issues that will have a negative effect on the development of their software solution in Unit 4 Outcome 1. Teachers can provide feedback on the quality of the designs, however, the adjustments must be initiated by the student and not directed by the teacher. While students can make changes to their designs they will not be reassessed and their original score will stand.

Unit 4

Software development: development and evaluation

Outcome 1

Develop and evaluate a software solution that meets requirements and assess the effectiveness of the project plan.

Nature of task

A software solution that meets the software requirements specification

**And**

Preparation and conduction of beta testing

**And**

* + an evaluation of the efficiency and effectiveness of the software solution
  + an assessment of the effectiveness of the project plan (Gantt chart) in monitoring project progress

in one of the following:

* + a written report
  + an annotated visual plan.

Time allocated should be at least 8 weeks of class time.

Scope of task

Development of the software solution

Criterion 6 assesses students’ skills in using the features of the programming language to develop the software solution. In order to develop the software solution students are required to use an appropriate programming language that meets the [prescribed list of software tools and functions, and outcome specific requirements](https://www.vcaa.vic.edu.au/curriculum/vce/vce-study-designs/appliedcomputing-softwaredevelopment/Pages/index.aspx) of the study. Students will also use a range of appropriate data types, data structures and data sources.

The evidence from this task is observed through Observation 6 and assessed through Criterion 6.

Criterion 7 assesses students’ skills in developing the software solution. Students will apply suitable naming conventions, write internal documentation and apply appropriate validation techniques.

The evidence from this task is observed through Observation 7 and assessed through Criterion 7.

Debugging and alpha testing of the software solution

Criterion 8 assesses students’ skills in debugging and alpha testing the software solution. Students will document the use of debugging and testing techniques to ensure their software solution functions as expected. Further details regarding solution testing are in the *Support material*.

Students will document evidence of their critical and creative thinking through the modification of designs, evaluation criteria and the development of the software solution as part of the Development stage in Criterion 8. Refer to the Skills underpinning the Development stage in the Units 1 to 4: Problem-solving methodology specifications on page 22 of the study design.

The evidence from this task is observed through Observation 8 and assessed through Criterion 8.

Beta testing

Criterion 9 assesses students’ skills in conducting beta testing. Students will document the preparation of a beta testing plan and test scenarios and then conduct the beta testing. After performing the beta testing with their clients or potential users, students will document the results. Students will use the results of the beta testing to make recommendations for modifications to the software solution. Students could choose to make modifications to the software solution or to document the actual modifications they would make to the software solution in a written report.

The evidence from this task is observed through Observation 9 and assessed through Criterion 9.

Evaluation of the software solution and assessment of the project plan

Criterion 10 assesses students’ skills in evaluating the software solution and assessing the project plan. Students will document the evaluation of the efficiency and effectiveness of the software solution using the evaluation criteria developed in Criterion 4. This includes the extent to which it meets the functional and non-functional requirements. Students will then propose an evaluation strategy to be conducted in the future.

Students will also need to document evidence of their critical and creative thinking through the evaluation of the process they followed through the analysis, design and development stages and discuss improvements that could be made to the software solution as part of the Evaluation stage in Criterion 10. Refer to the Skills underpinning the Evaluation stage in the Units 1 to 4: Problem-solving methodology specifications on page 23 of the study design.

Students will document the modifications made to the initial project plan throughout the duration of the project and then assess the effectiveness of the project plan.

The evidence from this task is observed through Observation 10 and assessed through Criterion 10.

Authentication

Teachers should monitor students’ progress on a regular basis and use the Authentication record form to record this information. The authentication process and feedback are clearly identified on this form so that teachers can provide feedback at various stages of the process. It is recommended that students back up files with copies of work in progress using, for example, an external drive, network drive or secure cloud storage.

Using the rubric

The following rubric is used to assess student achievement for 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. The rubric describes typical evidence associated with five different levels of performance for a criterion (five levels; 10 marks).

Criteria 1 to 5 relate to Unit 3 Outcome 2.

Criteria 6 to 10 relate to Unit 4 Outcome 1.

Before scoring the task for Unit 3 and Unit 4, the school should determine if the student has met the requirements of the task to receive an S or an N. The task is then scored by the school. If a student does not submit any of the SAT, a score of N/A should be entered for the School-assessed Task. A score of ‘zero’ is deemed a score for each criterion. A zero is only awarded if the work has not met the minimum requirements of the criterion.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 3 Outcome 2**  **1. Skills in developing a brief and in project management.** | * Documents a problem, need or opportunity as a brief. | Insufficient evidence | Identifies a problem, need or opportunity. | Describes how the proposed solution aims to address the identified problem, need or opportunity.  Identifies the users or clients of the proposed solution. | Explains how the proposed solution aims to address the identified problem, need or opportunity.  Describes the users or clients of the proposed solution. | Describes the relevant features of the programming language to be used within the proposed solution. | Justifies how the development of the proposed solution will be feasible, original and address the identified problem, need or opportunity. |
| * Prepares a Gantt chart using software that documents all stages and activities from the problem-solving methodology for Unit 3 Outcome 2 and Unit 4 Outcome 1. | Prepares a plan using software that documents the stages from the problem-solving methodology. | Prepares a Gantt chart using software that documents the stages from the problem-solving methodology for Unit 3 Outcome 2. | Prepares a Gantt chart using software that documents the stages from the problem-solving methodology for Unit 4 Outcome 1. | Prepares a Gantt chart that documents the stages and activities from the problem-solving methodology for Unit 3 Outcome 2 and Unit 4 Outcome1. | Prepares a Gantt chart that clearly and accurately documents the all the stages and activities from the problem-solving methodology for Unit 3 Outcome 2 and Unit 4 Outcome1. |
| * Documents all the relevant tasks, sequencing, time allocations, milestones, dependencies and critical path. |  | Lists relevant tasks. | Outlines a plan that includes tasks and time allocations. | Documents the appropriate sequencing of tasks, time allocations and teacher-provided milestones. | Documents the appropriate sequencing of student-provided milestones.  Explains why the project needs to be monitored. | Documents the dependencies and the critical path.  Discusses how the progress of the project will be monitored and documented. |
|  | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 3 Outcome 2**  **2. Skills in documenting the analysis.** | * Documents and prepares the data for analysis using appropriate data collection methods. | Insufficient evidence | Identifies data that is required to inform the analysis. | Collects data using one of the following data collection methods:   * interviews * observations * surveys * reports.   Outlines how the data will be collected to inform the analysis. | Collects data using two of the following data collection methods:   * interviews * observations * surveys * reports.   Describes how the data collected will be used to determine requirements, constraints and scope. | Collects data using three or more of the following data collection methods:   * interviews * observations * surveys * reports.   Describes how the data collected will be used to determine user characteristics and the technical environment. | Prepares the data for analysis by labelling and categorising the data.  Explains the use of the selected data collection methods. |
| * Uses relevant features of the selected analytical tools and illustrates the relationships between users, data and systems. |  | Identifies features of the selected analytical tools for illustrating. | Illustrates the features of the context diagram/s.  Illustrates the relationships between the existing system, entities and data flows.  Some errors or omissions exist. | Illustrates the features of the data flow diagram/s.  Illustrates the relationships between the processes, entities, data stores and data flows.  Some errors, inconsistencies or omissions exist. | Illustrates the features of the use case diagram/s.  Illustrates the relationships between the systems boundary, actors, associations, relationships (includes and extends) and use cases.  Minor errors, inconsistencies or omissions exist. | Illustrates correctly all the relevant features of the three selected analytical tools.  Illustrates correctly all the relationships between users, data and systems.  No errors, inconsistencies or omissions exist. |
|  | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 3 Outcome 2**  **3. Skills in documenting  a software requirements specification.** | * Documents the proposed software solution as part of  an SRS. | Insufficient evidence | Lists solution requirements. | Outlines the purpose and requirements of the proposed software solution in a formal document. | Documents the functional and non-functional requirements of the proposed software solution.  Documents the constraints that may impact the development of the proposed solution.  Documents the analytical tools. | Describes the characteristics of the users for the proposed software solution.  Describes the scope of the proposed software solution. | Describes the technical environment in which the proposed software solution will operate.  Organises the formal document using clear headings, sections and appendices. |
| * Documents the process of critical and creative thinking through critical analysis, the use of questions and follow-up questions to clarify the development of the software requirements specification. | Identifies the data that needs to be collected to inform the development of the software requirements specification. | Outlines the use of questions to critically analyse the data collected to inform the development of the software requirements specification. | Write questions to critically analyse the development of the software requirements specification. | Evaluates questions to critically analyse the development of the software requirements specification. | Writes follow-up questions to clarify the data collected to inform the development of the software requirements specification. |
|  | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 3 Outcome 2**  **4. Skills in generating design ideas and developing evaluation criteria.** | * Generates design ideas for the software solution. | Insufficient evidence | Lists design ideas for the software solution. | Illustrates design ideas for the software solution using one of the following ideation tools and techniques:   * mood boards * brainstorming * mind maps * sketches. | Generates design ideas for the software solution using two of the following ideation tools and techniques:   * mood boards * brainstorming * mind maps * sketches.   Uses annotations to identify the appearance of the software solution. | Generates design ideas for the software solution using three or more of the following ideation tools and techniques:   * mood boards * brainstorming * mind maps * sketches.   Uses annotations to explain the appearance and functionality of the software solution. | Generates feasible, original and distinctive design ideas using ideation tools and techniques that fully represent the software solution.  Uses annotations to describe and justify the appearance and functionality of the software solution. |
| * Develops evaluation criteria with reference to design ideas and the efficiency and effectiveness of the software solution. | Identifies measures to evaluate design ideas or the software solution. | Outlines criteria to evaluate the design ideas and the software solution, with reference to the functional and non-functional requirements. | Develops and applies evaluation criteria to determine which elements of the design ideas should be further developed into detailed designs. | Uses the evaluation criteria to explain which elements of the design ideas should be further developed into detailed designs. | Uses the evaluation criteria to justify which elements of the design ideas should be further developed into detailed designs. |
|  | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 3 Outcome 2**  **5. Skills in producing detailed designs.** | * Produces detailed designs for the software solution. | Insufficient evidence | Uses a mock-up to represent the user interface for the software solution. | Uses annotated mock-ups to represent the user interfaces within the software solution.  Uses a data dictionary with reference to data types. | Uses a detailed data dictionary, with reference to data types and data structures.  Uses an IPO chart and pseudocode to represent the functionality of the software solution.  Applies the relevant features of the design tools. | Uses a detailed data dictionary, with reference to data types, data structures and data sources.  Uses IPO charts and multiple modules of pseudocode to represent the functionality of the software solution.  Applies the relevant features of the design tools correctly. | Uses object descriptions and pseudocode to represent the functions and methods used within objects and classes.  Designs are feasible and complete. |
| * Documents the use of design principles and the characteristics of user experience in the detailed designs. | List design principles and characteristics of user experience that have been considered as part of the detailed designs. | Uses brief annotations to identify how the design principles have been applied within the detailed designs. | Uses annotations or written descriptions to describe how design principles have been applied within the detailed designs, with reference to appearance and functionality. | Uses annotations and written descriptions to document how relevant characteristics of user experience have been applied within the detailed designs. | Uses annotations and written explanations to demonstrate how the selected design principles and characteristics of user experience have been applied within the detailed designs. |
| * Documents the process of critical and creative thinking through the development of design ideas and the detailed designs. | Identifies existing and possible solutions to inform design ideas. | Outlines the connections between the design ideas using annotations. | Documents the connections between the design ideas and solution requirements. | Documents the connections between the design ideas, solution requirements and the detailed designs. | Documents possible contingencies when developing solution designs.  Documents possible solutions to mitigate issues. |
|  | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 4 Outcome 1**  **6. Skills in using the features of the programming language.** | * Uses a range of appropriate features of the programming language. | Insufficient evidence | Develops the solution using:   * instructions * arithmetic operators. | Develops the solution using:   * local variables and constants * logical and conditional operators * control structures of sequence and selection * graphical user interfaces (GUIs). | Develops the solution using:   * global variables * the control structure of iteration/repetition * relevant GUI controls. | Develops the solution using:   * functions and methods * access modifiers. | Develops the solution using:   * classes and objects.   Applies all relevant OOP principles. |
| * Uses a range of appropriate data types, data structures and data sources. | Identifies data types, including text, numeric and Boolean. | Uses data types for local variables.  Outlines, in the internal documentation, why the selected data types were used. | Uses appropriate data types for global variables and data structures, including arrays and/or records.  Identifies, in the internal documentation, why the selected data types and data structures were used. | Uses appropriate data types, data structures and data sources.  Describes, in the internal documentation, why the selected data sources were used. | Uses a range of data types, data structures and data sources.  Explains in the internal documentation, why the selected data types, data structures and data sources were used. |
|  | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 4 Outcome 1**  **7. Skills in developing the software solution.** | * Applies suitable naming conventions. | Insufficient evidence | Identifies naming conventions for use within the software solution. | Applies naming conventions to variables within the software solution. | Applies naming conventions to variables and interface controls within the software solution. | Applies naming conventions to variables, interface controls and code structures within the software solution. | Applies suitable naming conventions to all solution elements within the software solution. |
| * Writes comprehensive internal documentation. | Identifies the functioning of the software solution. | Outlines the functioning of the software solution using internal documentation. | Uses internal documentation to describe the:   * functionality of the software solution * use of data.   Displays evidence of code maintenance.  Some issues with clarity exist. | Uses internal documentation to explain the:   * functionality of the software solution * use of data * use of code structures.   Minor issues with clarity exist. | Uses internal documentation to explain the:   * functionality of the software solution * use of all data * use of all code structures.   Clear and concise. |
| * Applies appropriate validation techniques. | Identifies input data for validation. | Validates data using one of the following checks:   * existence * type * range. | Validates data using two of the following checks:   * existence * type * range.   Some inconsistencies are present. | Validates data using all the following checks:   * existence * type * range.   Minor inconsistencies are present. | Validates all relevant input data and checks the reasonableness and completeness of all input data.  No inconsistencies are present. |
|  | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 4 Outcome 1**  **8. Skills in debugging and alpha testing the software solution.** | * Documents the use of debugging and testing techniques to ensure the software solution functions as expected. | Insufficient evidence | Lists test data for testing the software solution.  Syntax errors are present. | Outlines some test data for the software solution in a testing table.  The testing table is incomplete.  Uses debugging statements to check the functionality of the software solution.  Logic errors are present. | Identifies test data for the software solution in a testing table with expected output stated.  Identifies input data to test validation checks.  Uses breakpoints to support debugging and testing.  Minor logic errors are present. | Documents test cases, with a range of test data, comparing both expected and actual output.  Documents suggested actions for rectifying errors in the testing tables.  Run-time errors are present. | Reports and displays evidence of test data for the software solution.  Documents a complete list of actions taken for failed tests in a testing table that compares both expected and actual output.  All modules function correctly. |
| * Documents the process of critical and creative thinking through the modification of the designs, evaluation criteria and the development of the software solution. | Identifies designs requiring modification. | Outlines the modifications and further development of the designs, using annotations. | Documents the modifications and further development of the evaluation criteria. | Explains why the modifications have been made to the designs and the evaluation criteria. | Documents possible contingencies when developing the software solution.  Documents possible solutions to mitigate issues. |
|  | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 4 Outcome 1**  **9. Skills in conducting beta testing.** | * Preparation of the beta testing plan and test scenarios. | Insufficient evidence | Lists components of the software solution to be included in beta testing. | Prepares a beta testing plan that targets the appearance of the software solution.  Outlines potential users to participate in beta testing. | Prepares a beta testing plan that targets the functionality of the software solution.  Explains why potential users have been selected to participate in beta testing. | Prepares a beta testing plan that targets the functional and non-functional requirements of the software solution.  Documents how the beta test results will be collected. | Prepares a beta testing plan that includes test scenarios that target relevant characteristics of user experience.  Clear and concise. |
| * Conduction of beta testing. | Conducts beta testing with one potential user. | Conducts beta testing and lists results using one data collection method. | Conducts beta testing and collects results by using two data collection methods. | Conducts beta testing with multiple potential users and collects results using three or more data collection methods. | Prepares the data collected for the documenting of the results of the beta tests. |
| * Documents the results of the beta tests. | Lists results of the beta tests. | Outlines the results of the beta tests. | Describes the results of the beta tests in a brief. | Documents the results of the beta tests in a report. | Documents and explains a complete set of results of the beta tests in a detailed report.  Clear and concise. |
| * Documents the recommended modifications to the software solution based on the results of the beta testing. | Identifies minor modifications to the software solution. | Outlines the modifications to the software solution. | Recommends and documents the modifications to the software solution. | Explains the purpose of the modifications to the software solution. | Evaluates the modifications to the software solution based on the results of the beta testing. |
|  | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 4 Outcome 1**  **10. Skills in evaluating the solution and assessing the project plan.** | * Documents the evaluation of the efficiency and effectiveness of the software solution. | Insufficient evidence | Identifies measures to evaluate whether the software solution has met solution requirements. | Outlines the evaluation criteria from Criterion 4 required to evaluate the efficiency and effectiveness of the software solution. | Uses the evaluation criteria from Criterion 4 to evaluate the efficiency and effectiveness of the software solution, and the extent to which it meets the functional and non-functional requirements. | Explains how the evaluation criteria has been measured to evaluate the efficiency and effectiveness of the software solution. | Proposes an evaluation strategy to be conducted sometime in the future to evaluate the efficiency and effectiveness of the software solution that includes:   * the time frame for the evaluation to be conducted * the evaluation criteria to be used * the individuals to conduct the evaluation and their responsibilities. |
| * Documents evidence of critical and creative thinking through the evaluation of the analysis, design and development stages and improvements to the software solution. | Identifies features of the software solution that meet functional and non-functional requirements. | Outlines the process of developing the software solution through the analysis, design and development stages. | Evaluates the use of the analysis, design and development stages in developing the software solution. | Critically evaluates the process of developing the software solution, from start to finish, through the analysis, design and development stages, and how this process assisted in meeting requirements. | Discusses and justifies improvements that could be made to the software solution by approaching the analysis, design and development stages differently. |
| * Documents the modifications made to the initial project plan throughout the duration of the project. | Lists modifications to the tasks during the project. | Uses annotations to outline the modifications made to the initial project plan. | Uses annotations to describe the modifications made to the initial project plan. | Uses adjustments or logs/journals to document and explain the modifications made to the initial project plan. | Evaluates the modifications made to initial project plan. |
| * Assesses the effectiveness of the project plan. | Lists factors that contributed to the effectiveness of the project plan. | Outlines the changes made to the project plan and how they impacted the effectiveness of the project plan. | Describes the reasons why changes were made to the project plan and how they impacted the effectiveness of the project plan. | Discusses how the changes made to the project plan impacted the completion of the project and the effectiveness of the project plan. | Evaluates the changes made to the project plan, with evidence, and how these changes impacted the completion of the project and the overall effectiveness of the project plan. |
|  | 0 ❑ | 1 ❑ 2 ❑ | 3 ❑ 4 ❑ | 5 ❑ 6 ❑ | 7 ❑ 8 ❑ | 9 ❑ 10 ❑ |

Authentication of VCE Applied 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* *Administrative Handbook 202*](http://www.vcaa.vic.edu.au/Pages/schooladmin/handbook/2018/index.aspx)*5.* 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 Applied Computing: Software Development School-assessed Task.

1. The body of work created for the 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 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 record form. If a student acknowledges using external resources or receiving external assistance, the teacher should record complete details as an attachment to the Authentication record form.

7. Teachers are reminded that the authentication procedures must be followed for all student work in relation to this SAT. The School-based Assessment Audit includes the inspection of Authentication record forms.

Authentication record form: VCE Applied Computing: Unit 3 Software Development SAT 2025

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |

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 the School-based Assessment Audit.

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

School …………………………………………………………………… Teacher: ……………………………………..………………………………………………….

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component of School-assessed Task** | **Date observed and submitted** | **Teacher comments** | **Teacher’s initials** | **Student’s initials** |
| **Observation 1: Development of a brief and project plan**  **(Criterion 1)**  The student is developing/has developed a brief documenting a real-world problem, need or opportunity that can be solved as a software solution. The teacher has approved or not approved the problem, need or opportunity. The student is preparing/has prepared a Gantt chart for both parts of the SAT (Unit 3 Outcome 2 and Unit 4 Outcome 1). | Observed | Observation of the development of the brief and the preparation of the project plan |  |  |
| Submitted | Submission of the brief and the project plan |  |  |
| **Observation 2: Documentation of the analysis (Criterion 2)**  The student is documenting and preparing /has documented and prepared the data and data collection methods for analysis. The student is using/has used analytical tools. | Observed | Observation of the documenting and preparing of the data for analysis and the use of analytical tools |  |  |
| Submitted | Submission of the documenting and preparing of the data for analysis and the use of analytical tools |  |  |
| **Observation 3: Development of a software requirements specification (Criterion 3)**  The student is documenting/has documented the analysis in the form of a software requirements specification. | Observed | Observation of the development of the software requirements specification |  |  |
| Submitted | Submission of the software requirements specification |  |  |
| **Observation 4: Generating design ideas and developing evaluation criteria (Criterion 4)**  The student is generating/has generated their design ideas and is developing/has developed their evaluation criteria. | Observed | Observation of the generation of the design ideas and the development of the evaluation criteria |  |  |
| Submitted | Submission of design ideas and evaluation criteria |  |  |
| **Observation 5: Producing detailed designs (Criterion 5)**  The student is developing/has developed the detailed designs. | Observed | Observation of the production of the detailed designs |  |  |
| Submitted | Submission of the detailed designs |  |  |

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

Student signature ………………………………………………………………… Date …………………………………

Authentication record form: VCE Applied Computing: Unit 4 Software Development SAT 2025

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |

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 the School-based Assessment Audit.

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

School …………………………………………………………….…….... Teacher: ……………………………………..…………………………………………………….

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component of School-assessed Task** | **Date observed and submitted** | **Teacher comments** | **Teacher’s initials** | **Student’s initials** |
| **Observation 6: Development of the software solution (Criterion 6)**  The student is using/has used a range of features of a programming language to develop the software solution. The student is using/has used a range of data types, data structures and data sources. | Observed | Observation of the development of the software solution |  |  |
| Submitted | Submission of the software solution |  |  |
| **Observation 7: Development of the software solution  (Criterion 7)**  The student is developing/has developed naming conventions, internal documentation and validation techniques. | Observed | Observation of the development of the software solution |  |  |
| Submitted | Submission of the software solution |  |  |
| **Observation 8: Debugging and alpha testing the software solution (Criterion 8)**  The student is debugging and alpha testing/has debugged and alpha tested the software solution. | Observed | Observation of the debugging and alpha testing of the software solution |  |  |
| Submitted | Submission of the debugging and alpha testing of the software solution |  |  |
| **Observation 9: Conducting Beta testing (Criterion 9)**  The student is preparing and conducting/has prepared and conducted the beta testing. The student is documenting/has documented the results of the beta testing. The student is recommending/has recommended modifications to the software solution. | Observed | Observation of the preparation, conducting and documenting of beta testing |  |  |
| Submitted | Submission of beta testing |  |  |
| **Observation 10: Evaluation of the software solution and assessment of the project plan (Criterion 10)**  The student is documenting/has documented the evaluation of the software solution and the assessment of the project plan. | Observed | Observation of the evaluation of the software solution and assessment of the project plan |  |  |
| Submitted | Submission of the evaluation of the software solution and 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 …………………………………

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **2025** | Victorian Certificate of Education  Applied 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** | | | Not Shown (0) | Very Low (1–2) | Low  (3–4) | | | Med  (5–6) | High  (7–8) | | Very High (9–10) | | **Performance on Criteria: Teacher’s Comments**  You may wish to comment on aspects of the student’s work that led to your assessment. | | | | | | | | | | |
| **The extent to which the student demonstrates skills in:** | | |  |  |  | | |  |  | |  | |
| 1 developing a brief and in project management | | |  |  |  | | |  |  | |  | |
| 2 documenting the analysis | | |  |  |  | | |  |  | |  | |
| 3 documenting a software requirements specification | | |  |  |  | | |  |  | |  | |
| 4 generating design ideas and developing evaluation criteria | | |  |  |  | | |  |  | |  | |
| 5 producing detailed designs | | |  |  |  | | |  |  | |  | |
| 6 using the features of the programming language | | |  |  |  | | |  |  | |  | |
| 7 developing the software solution | | |  |  |  | | |  |  | |  | |
| 8 debugging and alpha testing the software solution | | |  |  |  | | |  |  | | |  |
| 9 conducting beta testing | | |  |  |  | | |  |  | | |  |
| 10 evaluating the solution and assessing the project plan. | | |  |  |  | | |  |  | | |  |
| If a student does not submit the School-assessed Task  at all, N/A should be entered in the total score box. | | **SUBTOTALS** |  |  | |  |  | | |  | |  | |  | | | | | | | | | |

**TOTAL SCORE**