VCE Algorithmics (HESS) Administrative information for School-based Assessment in 2017

Unit 3 School-assessed Task

The School-assessed Task (SAT) for Unit 3 contributes 20 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 1, Outcome 2 and Outcome 3. The recorded scores must be based on the teacher's assessment of the student's performance according to the criteria on pages 6-12.This assessment is subject to the VCAA's statistical moderation process.

The 2017 VCE Algorithmics (HESS) assessment sheet on page 15 is to be used by teachers to record the Unit 3 SAT score. The completed assessment sheet for each student's SAT must be available on request by the VCAA. Student scores for the Unit 3 SAT must be submitted via VASS no later than 28 July.

The performance descriptors for the assessment criteria are published annually on the Algorithmics (HESS) study page of the VCAA website. The performance descriptors for the assessment criteria for the Unit 4 SAT will be published on the Algorithmics (HESS) study page in Term 1, 2017. Details of authentication requirements and administrative arrangements for School-assessed Tasks are published in the *VCE and VCAL Administrative Handbook 2017*.

The School-assessed Task for Unit 3 relates to:

- Outcome 1
- Outcome 2
- Outcome 3.

Unit 3

Data modelling with abstract data types

Outcome 1

Devise formal representations for modelling various kinds of information problems using appropriate abstract data types, and apply these to a real-world problem.

Nature of tasks

- Folio of two to four tasks using a range of abstract data types (ADTs) to model the salient aspects of problems
- A written explanation of the specification and application of standard ADTs (approximately 45−60 minutes)
- A data model of a real-world problem, including
  - specification of the data model
  - a concrete instance of the data model (worked example).

Scope of tasks

Folio

Teachers must provide students with a range of small tasks that require them to use a variety of ADTs to model aspects of problems. These tasks should not be onerous, but rather form part of the teaching and learning program, while still subject to authentication conditions. The folio will typically comprise incremental pieces of work, reflecting the student’s learning progression through the levels. It is recommended that students submit four pieces of work and the teacher select the best two or more pieces for assessment. The folio pieces should allow students to provide evidence for statements at all levels of performance through criterion 2.

Written explanation

Teachers must provide students with a task that allows them to explain in writing the specifications and applications of standard ADTs. This would be done under test conditions within a timeframe of 45 to 60 minutes. It is recommended that the task include a mixture of question types that allows the students to provide evidence for statements at all levels of performance. The evidence from this task is assessed through Criterion 1.

Data model

Students devise specific-task definitions based on a real-world problem in order to create a data model. Teachers can provide generic real-world problems, but each student must model, using mainly the graph ADT, a specific or concrete instance of that problem. Teachers can better authenticate student work when there are individual instances of a generic problem. Teachers should approve each student’s proposal before they model the problem as a worked example. The performance descriptors at the highest level for Criterion 2 should be used to help make this decision. Note: This data model will form the basis of a task in Outcome 2.

The evidence related to the quality of the data model is assessed through Criterion 2.

Algorithm design

Outcome 2

Design algorithms to solve information problems using basic algorithm design patterns, and implement the algorithms.

Nature of task

- Folio of two to four tasks using a range of algorithm design patterns to specify algorithms to solve problems
- A written explanation of the specification and application of algorithms for graphs (approximately 45−60 minutes)
- An algorithm to solve a real-world problem that builds on an existing data model including:
  - pseudocode to solve the problem
  - implementation of the algorithms in a high-level programming language making appropriate use of standard ADTs.

Scope of task

Folio

Teachers must provide students with a range of small tasks that require them to use a variety of standard algorithms for graphs to solve problems. These tasks should not be onerous, but rather form part of the teaching and learning program, while still subject to authentication conditions. The folio will typically comprise incremental pieces of work, reflecting the student's learning progression through the levels. It is recommended that students submit four pieces of work and the teacher select the best two or more pieces for assessment. The folio pieces should allow students to provide evidence for statements at all levels of performance through Criterion 4.

Written explanation

Teachers must provide students with a task that allows them to explain in writing the specification and application of algorithms for graphs. This would be done under test conditions within a timeframe of 45 to 60 minutes. It is recommended that the task include a mixture of question types that allows the students to provide evidence for statements at all levels of performance. The evidence from this task is assessed through Criterion 3.

Solution to a real-world problem

Students build on the data model they developed in Outcome 1 and create a solution by developing an algorithm and implementing it using a high-level programming language. In some circumstances it may be appropriate for the teacher to provide students with a data model, rather than the student using their own. If the student-generated data model is incomplete or contains significant errors, and this would prevent the demonstration of the highest level of achievement on the relevant criteria, then the student could use a provided model. Teachers could either modify the student-generated model or provide a new one.

When assessing performance with the programming language, it is important that students are judged on their conceptual use of the language rather than their mastery of the language or its libraries. The programming language requirements are published in Notice to Schools 122/2016 and are posted on the Algorithmics study page.

The evidence related to the quality of the algorithm is assessed through Criterion 4.

Applied algorithms

Outcome 3

Evaluate and document algorithms and data representations, and solve a real-world problem, the solution for which requires the integration of algorithms and data types.

Nature of tasks

- an evaluation of an existing data model and algorithm in the form of a written report
  (approximately 300-500 words)
- documentation that communicates the:
  - data model development approach (approximately 300-500 words)
  - algorithm development approach (approximately 300-500 words).

Scope of task

Evaluation

Students must evaluate a variety of algorithms and ADTs when determining a suitable combination to solve their real-world problem. Their evaluation involves measuring the extent to which different combinations of algorithms and data models are best fit for purpose. The basis for evaluation includes selection of salient features of the problem, the modular representation of data using ADTs and the quality of the result generated by the algorithms. The word range for this evaluation is 300 to 500 words. The evidence from this task is assessed through Criterion 5.

Documentation

Students must document the development of their data model and algorithm. The documentation should include:

- a statement of the problem
- a systematic description of the methods used to develop the data model and algorithm
- identification of alternative solutions that were considered
- presentation of results, including a solution to the problem.

The evidence from this task is assessed through Criterion 6. Students should be encouraged to use appropriate metalanguage in their documentation.

The following rubric is used to assess student achievement on Unit 3 Outcome 1, Outcome 2 and Outcome 3. Teachers assess evidence produced from the tasks against the criteria and performance descriptors to grade achievements.

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).

Note: This rubric is based on the premise that each column includes all evidence of the columns to the left for each criterion, that is, the conditions are cumulative.

The following table shows the relationships between the outcomes, tasks and criteria.

Outcome	Task	Criteria
Outcome 1	A folio of two to four tasks using a range of ADTs to model the salient aspects of problems	2
Outcome 1	A written explanation of the specification and application of ADTs	1
Outcome 1	A data model of a real-world problem including: • the specification of the data model • concrete instance of the data model (worked example)	2
Outcome 2	A folio of two to four tasks using a range of algorithm design patterns to specify algorithms to solve problems	4
Outcome 2	A written explanation of the specification and application of algorithms for graphs	3
Outcome 2	An algorithm to solve a real-world problem that builds on an existing data model including: • pseudocode to solve the problem • implementation of the algorithms in a high-level programming language making appropriate use of the standard ADTs	4
Outcome 3	An evaluation of an existing data model and algorithm in the form of a written report	5
Outcome 3	Documentation of the data model development approach to solve a real-world problem	6
Outcome 3	Documentation of the algorithm development approach to solve a real-world problem that builds on an existing data model	6

VCE Algorithmics (HESS): Unit 3 School-assessed Task 2017						
Assessment criteria	Levels of Performance					
	Not shown	1 – 2 (low)	3 – 4	5 – 6 (medium)	7 – 8	9 – 10 (high)
1 Understanding of abstract data types		Describes motivations for the abstraction of data. Describes an appropriate example use-case for a given ADT. Uses some appropriate metalanguage when discussing the operations of ADTs and the components of an ADT signature specification. Uses appropriate metalanguage when describing graphs. Describes an example problem attribute that could be modelled by graph node or edge.	Explains the general concept of an ADT using more than one example. Describes the function and operations of a given ADT using appropriate metalanguage. Provides appropriate example use-cases for the ADT. Executes a sequence of ADT operations to a given ADT instance. Constructs an example of a specific class of graph. Confirms or rejects the properties of a graph given as a diagram. Describes how a graph property could be used to model a given aspect of a problem.	Describes the function and operations of several ADTs using appropriate metalanguage. Also, provides appropriate example use-cases for the ADTs. Translates a natural language description into a sequence of ADT operations. Writes signature specifications for some of the specified ADTs. Some minor errors or omissions exist. Constructs a graph satisfying multiple given conditions. States the graph properties satisfied by a graph given as a diagram.	Writes complete signature specifications for several ADTs, fully utilising appropriate metalanguage. Analyses the properties satisfied by a given graph and rigorously explains how these properties are satisfied. Explains the interconnections between the properties of cyclicity, connectedness and distance and the graph tree specialisation.	Describes a new operation for one of the standard ADTs to accommodate requirements that cannot be satisfied by the standard definition. Uses the specified properties of graphs to derive other graph properties.

VCE Algorithmics (HESS): Unit 3 School-assessed Task 2017						
Assessment criteria	Levels of Performance					
	Not shown	1 – 2 (low)	3 – 4	5 – 6 (medium)	7 – 8	9 – 10 (high)
2 Skills in the application of abstract data types		Selects ADT representations for problems that represent minimal features of the problem or model only a limited range of possible problem instances. The data model may include extraneous features. Little discrimination is demonstrated when identifying features of the problem. Models some aspects of a specific problem instance either in writing, graphically or within a programming environment. Manually follows a sequence of decisions through a decision tree or state transitions through a state diagram.	Selects ADT representations for problems that represent some critical features of the problem. Some discrimination is demonstrated in the identification of relevant problem features. The data model may not appropriately consider its scalability to larger problem instances. Models some aspects of several specific problem instances either in writing, graphically or within a programming environment. Applies ADT operations to an existing data model. Given a data model instances, including a state diagram or decision tree, describes some aspects of a problem.	Selects ADT representations, including use of the graph ADT, for problems that are fit for purpose. The full range of problem instances can be represented. Selects suitable graph attributes to represent a network problem based on information presented in a different form such as a table or diagram. Fully represents a specific problem instance as a data model either in writing, graphically or within a programming environment. Constructs accurately a decision tree, with multiple levels, for a simple hierarchical choice scenario. Clearly describes how aspects of a problem map to aspects of a data model.	Designs ADT representations, including use of the graph ADT, for problems that have a structure that cannot be modelled fully using a single abstract data type. Identifies aspects of problems that cannot be fully or adequately represented by ADTs as part of a data model. Fully represents several specific problem instances as data models either in writing, graphically or within a programming environment.	Designs ADT representations, including the use of the graph ADT, for problems that have a structure that cannot be modelled fully using a single abstract data type. The design is fit for purpose and appropriately prioritises aspects of the problem that are most important to the specific context or the requirements of algorithms utilising the data structure. Fully represents a complex problem instance as a data model either in writing, graphically or within a programming environment.

VCE Algorithmics (HESS): Unit 3 School-assessed Task 2017						
Assessment criteria	Levels of Performance					
	Not shown	1 – 2 (low)	3 – 4	5 – 6 (medium)	7 – 8	9 – 10 (high)
3 Understanding of the principles of algorithm design		Explains the concepts of decisions and iteration in algorithms. Identifies some algorithm design approaches. Identifies the elements of sequence, selection, and repetition in a given algorithm. Names and states correctly the computational applications of most of the specified graph algorithms. States informally the input types of the specified graph algorithms.	Explains the concept of modularisation. Explains the principles of the brute-force search or greedy algorithm design patterns, utilising appropriate examples. Explains the principles of each graph traversal technique, utilising appropriate examples. Interprets pseudocode with minimal errors, as demonstrated by the ability to execute it manually. The pseudocode includes sequence, selection, and iteration, but not nested iteration. Explains informally how some of the specified graph algorithms perform their computation and writes the approximate pseudocode for these. Describes an argument for the correctness of one of the specified	Explains the concept of recursion. Explains the principles of the decrease-and-conquer algorithm design pattern, utilising appropriate examples. Compares the relative advantages of the different graph traversal techniques. Interprets fluently pseudocode containing nested iteration and the use of ADTs, demonstrated by the ability to execute it accurately. Writes pseudocode for a procedure described in natural language, including nesting and modularisation. The pseudocode may contain minor errors. States precisely the input types of the specified graph algorithms. Executes any of the	Explains the concept of equivalence between recursive and iterative algorithms. Improves a piece of pseudocode by restructuring and modularisation. Explains the attributes required of problems for each of the algorithm design patterns to be applied. Identifies cases where a piece of pseudocode does not perform the desired operation correctly. Executes, without error, any of the specified graph algorithms using manual techniques for complex graphs. Describes an argument for the correctness of one of the specified graph algorithms that considers the general case of the problem but not all steps in the chain	Demonstrates the equivalence between recursive and iterative algorithms. Improves a piece of pseudocode by restructuring and modularisation using non-trivial user-defined functions. Evaluates the suitability of an algorithm design pattern for a specific problem. Explains in precise terms why any of the specified graph algorithms are not valid for some classes of graph or graphs with certain properties. Describes a valid argument for the correctness of at least one of the specified graph algorithms using either the induction or contradiction method.

VCE Algorithmics (HESS): Unit 3 School-assessed Task 2017						
Assessment criteria	Levels of Performance					
	Not shown	1 – 2 (low)	3 – 4	5 – 6 (medium)	7 – 8	9 – 10 (high)
			graph algorithms that considers only the correctness of a specific example.	specified graph algorithms using manual techniques for given graphs; the execution is performed with few errors.	of argument are explained.	

<table>
<tr><th colspan="7">VCE Algorithmics (HESS): Unit 3 School-assessed Task 2017</th></tr>
<tr><th rowspan="2">Assessment criteria</th><th colspan="6">Levels of Performance</th></tr>
<tr><th>Not shown</th><th>1 – 2 (low)</th><th>3 – 4</th><th>5 – 6 (medium)</th><th>7 – 8</th><th>9 – 10 (high)</th></tr>
<tr><td>4 Skills in the application of the principles of algorithm design</td><td></td><td>Designs simple algorithms and writes these in pseudocode. There may be significant task scaffolding required, and the final algorithm may be an effective method for a trivial subset of problem instances.

Identifies an appropriate search algorithm to apply to a given problem.

Identifies the input and output data type required by an algorithm.</td><td>Designs simple algorithms and write these in pseudocode, with some task scaffolding. The algorithm is an effective method for a non-trivial subset of problem instances.

Compares the execution of depth-first search (DFS), breadth-first search (BFS) or best-first search on a specific problem instance.

Uses a programming environment to complete a partially translated algorithm, including iteration, from pseudocode to program code.</td><td>Designs algorithms, including the use of iteration, and writes these in pseudocode, with minimal errors.

Applies the general concept of a given algorithm design pattern to design an algorithm to solve a problem, with errors in the final solution.

Uses a programming environment to translate a complete algorithm, including nested iteration, from pseudocode to program code.</td><td>Designs algorithms for problems that have a structure that does not allow for the direct application of one of the studied algorithms. Uses iteration and recursion where appropriate, and writes these in pseudocode.

Applies a given algorithm design pattern to design an algorithm to solve a problem.

Uses a programming environment to translate a recursive algorithm from pseudocode into program code.</td><td>Designs algorithms for problems that have a structure that does not allow for the direct application of one of the studied algorithms. Uses iteration, recursion and functions where appropriate, and writes these in pseudocode.

Selects a suitable algorithm design pattern for solving an information problem and applies the design pattern to design an algorithm to solve the problem.

Uses a programming environment to translate an algorithm, containing non-trivial functions, from pseudocode into program code.</td></tr>
</table>

VCE Algorithmics (HESS): Unit 3 School-assessed Task 2017						
Assessment criteria	Levels of Performance					
	Not shown	1 – 2 (low)	3 – 4	5 – 6 (medium)	7 – 8	9 – 10 (high)
5 Skills in the evaluation of algorithms and data representation		Describes reasons for the a choice of data model and algorithm based on the consideration of limited aspects of the problem.	Compares a data model and algorithm with another possible choice, as provided, based on a consideration of limited aspects of the problem.	Justifies a choice of data model and algorithm based on a comprehensive consideration of aspects of the problem. The selected data model and algorithm combination suitably addresses all essential aspects of the problem. Identifies competing aspects of a problem that are unable to be mutually satisfied.	Evaluates a choice of data model and algorithm. The evaluation considers the essential aspects of the problem and the properties of alternative choices of data model and algorithm. The chosen data model and algorithm suitably addresses the problem but may have shortcomings in terms of efficiency, clarity or succinctness. Justifies how competing aspects of a problem were prioritised when designing the data model and algorithm.	Compares and evaluates the relative advantages of data models and algorithm design patterns and consequently selects suitable data models and algorithms for solving a complex information problem to create a complete, correct and elegant solution.

VCE Algorithmics (HESS): Unit 3 School-assessed Task 2017						
Assessment criteria	Levels of Performance					
	Not shown	1 – 2 (low)	3 – 4	5 – 6 (medium)	7 – 8	9 – 10 (high)
6 Skills in the documentation and communication of solutions to information problems		Unfocused and superficial discussion of the methods used in the data model and algorithm design. Communication of simple ideas and arguments lacks some precision and contains some ambiguities. Documentation contains little organisation of ideas or use of the structural conventions of a written report.	Limited discussion of the methods used in the data model and algorithm design, possibly with some inconsistencies between the methodology described and the submitted design. Communication of simple ideas and arguments is mostly clear and coherent, with some use of metalanguage. Documentation contains some organisation of ideas, including some use of the structural conventions of a written report. Summarises a computed solution.	Relevant and clear discussion of the methods used in the data model and algorithm design. Communication of simple ideas and arguments is clear and makes use of appropriate metalanguage. Some imprecision or ambiguity exists when communicating more complex concepts. Documentation contains an orderly development of ideas and elements, including the use of the structural conventions of a written report. Describes a computed solution, with some connections made to the original problem context.	Thorough and sound discussion of the methods used in the data model and algorithm design. Communication of complex ideas and arguments is clear and uses appropriate metalanguage. Documentation contains a well-structured development of ideas and elements. Synthesises a computed solution to develop an answer in terms of the original problem context.	Highly detailed and complete discussion of the methods used in the data model and algorithm design. Communication of complex ideas and arguments is clear and concise, with proficient use of metalanguage. Documentation contains a well-structured and coherent development of ideas and elements.

Authentication of VCE Algorithmics (HESS) 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 2017*. 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 Algorithmics (HESS) 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 1, Outcome 2 and Outcome 3.

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 provides, their documentation should demonstrate ongoing progress throughout the SAT.

6. During development of the data model and 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.

Authentication Record Form VCE Algorithmics (HESS) 2017

School-assessed Task Unit 3

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: Development of folio (ADTs) Student has developed tasks for the folio for Outcome 1.				
Observation: Development of folio (algorithm design patterns) Student has developed tasks for the folio for Outcome 2.				
Observation: Progressive development of data model Student has continued to develop a data model and documentation.				
Observation: Submission of tasks Student has submitted the folio, data model and documentation.				
Observation: Progressive development of solution and documentation Student has continued to write pseudocode, implement it and prepare documentation.				
Observation: Submission of tasks Student has submitted the solution and documentation.				
Observation: Evaluation Student has submitted the evaluation of the data model and algorithm.				

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

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

2017

Victorian Certificate of Education
Algorithmics (HESS) Assessment Sheet
School–assessed Task: Unit 3

STUDENT NAME

STUDENT NUMBER

ASSESSING SCHOOL NUMBER

Teachers need to make judgments on the student's performance for each assessment 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.

Assessment criteria

The extent to which the student demonstrates:

	Not Shown (0)	Low (1–2)	(3–4)	Med (5–6)	(7–8)	High (9–10)
1 understanding of abstract data types						
2 skills in the application of abstract data types						
3 understanding of the principles of algorithm design						
4 skills in the application of the principles of algorithm design						
5 skills in the evaluation of algorithms and data representation						
6 skills in the documentation and communication of solutions to information problems						

Levels of Performance: Teacher's Comments
You may wish to comment on aspects of the student's work that led to your assessment of High, Medium, Low or Not Shown for specific criteria.

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