

Assessment Schedule – 2011

Technology: Demonstrate understanding of basic concepts from computer science (91074)

Final grades will be decided using professional judgement based on a holistic examination of the evidence provided against the criteria in the Achievement Standard.

Issues from the Specifications

Authentic candidate submissions will be recognisable because of specific contexts associated with the work. This does not imply that submissions will arise only from the candidate's practice. However, where the candidate's practice does not provide the immediate source of a specific context, one would expect to see that several sources of information relating to subsystems had been applied within a specific context. In both cases, the marker will be able to detect the candidate's voice. In situations where information does not have some aspect of student voice, it is difficult to establish whether the candidate has actually demonstrated understanding or simply identified information.

Candidates who have simply identified information by reproducing information from sources without making use of that information have not demonstrated understanding.

Where a candidate has provided a brief answer, the answer should not be penalised because of length.

Candidate work in excess of 14 pages should not be marked.

Where work is illegible, it cannot be marked.

Digital submissions that cannot be read cannot be marked.

| Not Achieved | Achievement | Achievement with Merit | Achievement with Excellence |
|---|---|--|---|
| Does not <i>demonstrate understanding of basic concepts from computer science</i> | Demonstrate understanding of basic concepts <i>from computer science</i> | Demonstrate in-depth understanding of basic concepts <i>from computer science</i> | Demonstrate comprehensive understanding <i>from computer science</i> |
| Evidence that supports the Not Achieved judgement | Evidence that supports the Achievement judgement | Evidence that supports the Achievement with Merit judgement | Evidence that supports the Achievement with Excellence judgement |
| <p>Candidates who have simply identified information by reproducing information from sources without making use of that information have not demonstrated understanding.</p> <p>Refer to Exemplars 1 and 2</p> | <p>Candidate:</p> <ul style="list-style-type: none"> describes the key characteristics and roles of algorithms, programs, and informal instructions describes an algorithm for a task, showing understanding of the kinds of steps that can be in an algorithm, and determining the cost of an algorithm for a problem of a | <p>Candidate:</p> <ul style="list-style-type: none"> explains how algorithms are distinct from related concepts such as programs and informal instructions shows understanding of the way steps in an algorithm for a task can be combined in sequential, conditional, and iterative structures and determining the cost of an iterative algorithm for a | <p>Candidate:</p> <ul style="list-style-type: none"> compares and contrasts the concepts of algorithms, programs, and informal instructions determines and compares the costs of two different iterative algorithms for the same problem of size n compares and contrasts high-level and low-level (or |

| | | | |
|--|---|--|--|
| | <p>particular size</p> <ul style="list-style-type: none"> describes the role and characteristics of programming languages, including the different roles and characteristics of high-level languages and low-level (or machine) languages, and the function of a compiler describes the role of a user interface and factors that contribute to its usability. <p>Refer to Exemplars 1, 2, and 3</p> | <p>problem of size n</p> <ul style="list-style-type: none"> explains how the characteristics of programming languages, including the different characteristics of high-level and low-level (or machine) languages, are important for their roles explains the need for programs to translate between high- and low-level languages explains how different factors of a user interface contribute to its usability. <p>Refer to Exemplars 4 and 5</p> | <p>machine) languages, and explains different ways in which programs in a high-level programming language are translated into a machine language</p> <ul style="list-style-type: none"> discusses how different factors of a user interface contribute to its usability by comparing and contrasting related interfaces. <p>Refer to Exemplars 6, 7, and 8</p> |
|--|---|--|--|

The *basic concepts from computer science* are the concept of an algorithm, the concept of a programming language, and the concept of a user interface and its usability.

An algorithm is a precise unambiguous specification of how to accomplish some computational task in a finite number of well-defined steps. An algorithm is distinct from a computer program. An algorithm has a cost (the number of steps it will perform) for a task. Different algorithms for the same task may have different costs.

A programming language is a precise, formal language for writing programs that can be run on a computer; it is distinct from pseudo-code and natural language. There are different levels of programming languages; programs can be translated from high level to low-level (or machine) languages.

A user interface is the part of a computer or electronic system that a human user interacts with to control the system. The usability of an interface is the key characteristic for evaluating an interface.