2007 HSC Notes from the Marking Centre

Software Design and Development

Published by Board of Studies NSW
GPO Box 5300
Sydney 2001
Australia

Tel: (02) 9367 8111

Fax: (02) 9367 8484

Internet: http://www.boardofstudies.nsw.edu.au

ISBN 978 174147 8495

2007752

Contents

2007 HSC NOTES FROM THE MARKING CENTRE SOFTWARE DESIGN AND DEVELOPMENT

Introduction

This document has been produced for the teachers and candidates of the Stage 6 course in Software Design and Development. It contains comments on candidate responses to the 2007 Higher School Certificate examination, indicating the quality of the responses and highlighting their relative strengths and weaknesses.

This document should be read along with the relevant syllabus, the 2007 Higher School Certificate examination, the marking guidelines and other support documents which have been developed by the Board of Studies to assist in the teaching and learning of Software Design and Development.

General Comments

In 2007, approximately 1820 candidates attempted the Software Design and Development examination.

Teachers and candidates should be aware that examiners may ask questions in Sections I and II that combine knowledge, skills and understandings from across the core of the HSC syllabus.

Section I

Question	Correct Response
1	B
2	C
3	A
4	B
5	D
6	B
7	C
8	B
9	D
10	A

Question	Correct Response
11	B
12	D
13	D
14	C
15	D
16	C
17	A
18	C
19	A
20	C

Section II

General Comments

The 2007 Higher School Certificate Examination in Software Design and Development required candidates to analyse and interpret situations and to apply their knowledge to these situations. Many responses showed a sound understanding of concepts but were less able to apply this knowledge appropriately, often giving general answers or answers not directly related to the particular situation described in the question.

Question 21

(a) (i) Some responses indicated confusion as to whether quality assurance is a specific item or a process. Weaker responses tried to define quality assurance by using the terms 'quality' and 'assurance' in their answer thereby simply restating the question. Better responses referred to a 'standard'.

(ii) While many candidates demonstrated that they understood what was needed in good user documentation, only the better responses really described processes that a developer uses for quality assurance when producing user documentation. Weaker responses had difficulty distinguishing between user documentation and internal documentation..

(b) (i) Typical responses identified general characteristics of RAD, but had difficulty with including specifics that related to the required printer software. Justifications for its use in the given scenario were poor – many responses demonstrated only a very superficial understanding of the given context. Better responses described issues such as cost and the need for the software to be developed quickly to get the printer onto the market.

(ii) Typical responses identified features of prototyping, but did so without specifically relating to the scenario. However, many candidates had the ability to talk about a process in this part and often described what they personally would do, though terminology was often vague. Responses tended not use terms such as 'model' in their discussion – instead they tended to use the term 'prototype' as given in the question. Better responses described the particular features that would be included in a model such as the user interface and the process by which this interface could be refined.

(c) (i) Weaker responses to this part highlighted a lack of understanding of the use of the FOR NEXT loop construct. They did not recognise the syntax and so were distracted from the true logic error. Many wanted to initialise the counter before the loop and include an increment of the counter within the loop. Weaker responses also indicated a lack of understanding of the purpose of Line 40, claiming that the counter was incremented twice in this line. The better responses clearly identified the fact that the algorithm produces a times table from 1 to 13, and described how to fix the error.

(ii) Typical responses demonstrated an understanding that the contents of the variable 'table' needed to change, with the weakest responses just replacing the value 6 with a different constant. Responses that indicated changing lines in the algorithm with a simple explanation tended to score better than those that included a lengthy narrative description only. Better responses demonstrated the need to make the value for 'table' a user input rather than a constant, though many then did not change the literal value 6 in the print

statement to suit. The best responses indicated the need to include a second loop with user variables to cater for the range of values as required.

(d) (i) Weaker responses found it difficult to correctly describe a data structure. They did not appropriately name a variable and simply provided a data dictionary with little relation to the required record structure.

(ii) Many responses outlined basic features of sequential versus random access but without appropriately demonstrating a real understanding of how these files could be used in this particular scenario. Only the best responses truly provided an evaluation of each method in this system, perhaps indicating a lack of understanding of the possible uses of the file 'WaterUsage'. Typical responses were simply descriptions, with most responses outlining sequential access as reading the file from beginning to end.

Question 22

(a) (i) Weaker responses did not provide a clear purpose for peer checking.

(b) Typical responses described the use of breakpoints and single line stepping. Better responses provided an explanation of how these techniques are used to locate logic errors. Many weaker responses did not clearly describe the use of both terms, neglecting the advantages of using both strategies to complement each other to find the location and cause of logic errors.

(c) Typical responses provided reasons why screen design is important but few explained why it should be considered during the *defining and understanding the problem* phase. Better responses demonstrated a good understanding of the need to clarify inputs and outputs (including specific data variables required) in this early stage of a project.

(d) (i) Weaker candidates demonstrated a poor understanding of a program counter and its role, with many incorrectly describing a program counter as a variable in a program used to count the number of loops. Better responses described the role of program counter in executing a program. Some responses were unclear about what the program counter actually stored, describing it as storing the instruction to be executed rather than the address of the next instruction to be executed.

(ii) Typical responses identified the impact on the browser, even where they struggled to describe the purpose of a program counter in part (i). Better responses gave the reason for what might occur and provided possible results that would eventuate due to the described changes.

(iii) Poorer responses gave only a general description of a malicious program, or provided a scenario of a virus that changed the location stored in the program counter rather than placing malicious code into that location and thus taking advantage of the known error. Better responses provided a scenario which demonstrated how the error causing the content of the program counter to change could be utililised by the malicious program, without the permission or knowledge of the user.

(e) (i) Typical responses recognised that the error produced is a logic error and provided an accurate description of its effect.

(ii) Most candidates were able to demonstrate their understanding of the algorithm provided in the question, and recognised where changes were needed in the algorithm. Poorer responses were unable to show the logic required to make these changes appropriately. Some responses showed a poor understanding of how random numbers are generated. Better responses identified the need to reduce the number of runners allocated to lanes (in lines 190 to 220) as well as changing the algorithm logic to keep lane 6 free.

Question 23

(a) (i) Better responses mentioned boundary values for repetition and selection control structures or that test data needed to test all possible paths through an algorithm or program. Poorer responses simply mentioned that test data was used to check a program.

(ii) Better responses indicated that live test data was real data used to simulate the final environment in which the software solution would be run. These responses did not just provide a fuller description/explanation of test data but also discussed the type of testing required, focusing on volume testing, response times and beta testing. Also included in these responses were benefits to the developer of using this form of testing. Weaker responses either simply expanded on their answer to part (i) or provided a superficial response without showing any understanding of the process.

(b) (i) Most candidates recognised that *isprime* is a Boolean variable or flag. Better responses described how the *WHILE* loop terminated as soon as *isprime* was set to false and that *isprime* is used to communicate the result of the *Checkprime* test to the main program. Weaker responses simply stated that *isprime* was set false when a factor was found but did not describe the purpose of *isprime* in the overall algorithm

(ii) Better responses described the fact that when j = 1 the *IF* statement is always true as 1 is a factor of all numbers. This makes *isprime* always false, and so all numbers will be reported as non-prime (which is clearly a logic error). Unfortunately many candidates recognised much of this logic but then stated that all numbers would be reported as prime. Weaker responses showed little understanding of the algorithm, particularly with respect to the *mod* function, even though its use was well described in the question.

(iii) The best responses used a *FOR* loop and an *IF* statement with the *mod* condition which resulted in a clear elegant brief solution. Responses using a *WHILE* loop often had difficulty with correct initialisation and incrementing of the loop. Better responses looped through all values from 1 to *n*, tested each value to see if it was a factor of *n* and then print it. Weaker responses simply copied parts of the algorithm from the question and made a superficial attempt at a solution. Many candidates were unsure of the use of the parameter *n* and included input statements within their subprocedure.

(c) (i) Some candidates wasted time by describing what the processes did, when the question only asked them to identify the processes. Weaker responses attempted to describe the

data movement or interaction of external identities with the system. Some weaker responses indicated the three processes but did not use the process names as used in the question.

(ii) Better responses stated that the additional data flow diagrams were to show the next level of detail within the process, indicating the hierarchical nature of data flow diagrams. They also explained the benefits of this to the developer such as ease of understanding the system, particularly when subsequently coding or modifying the system.

(iii) Better responses replaced the single *Transact Loan* process with processes for *OverdueBooks* and *NumberOfBooks* plus another new process for carrying out the approval. These responses also shared all data between the three new processes and the other elements as shown in the question. Weaker responses did not recognise the need to include processes for *OverdueBooks* and *NumberOfBooks* as stated in the question. Many weaker responses also included the *Transact Loan* process in their diagram, indicating a poor understanding of the hierarchy involved in constructing data flow diagrams.

Question 24 – Evolution of Programming Languages

(a) (i) Weaker responses demonstrated a poor understanding of the functional paradigm compared to the other paradigms. A large number of these responses considered functions in terms of mathematical functions rather than software functions or confused functions with function calls. Many responses failed to note that a function is a separate module or subprogram.

(ii) Some responses included the use of ‘text1 text2 text3’ as used in the question for the function names, indicating a poor understanding of the use of syntax specifications. A significant number of responses also failed to demonstrate working as specifically asked for in the question. However those responses that did include working were mostly done well.

(iii) Weaker responses failed to explicitly relate to the historical reasons. Many responses were not in the form of a discussion, instead providing simply a lower level description of the paradigm or listing reasons for development of the paradigm.

(b) (i) Many responses confused a fact with a constant used in a non-logic paradigm or confused a fact with a goal. Some better responses included an example of a fact to illustrate their definition.

(ii) Better responses demonstrated good understanding of forward and backward chaining. Weaker responses defined one concept of a pair and then declared the second as being ‘the opposite’.

(iii) A large number of weaker responses simply substituted ‘cat’ for ‘X’ and did not describe the evaluation with reference to the code. Many tried to use the mammal rule to prove that a cat is an animal and a cat suckles its young rather than using these facts to prove that a cat is a mammal.

(iv) Many responses indicate a fail use to add another *suckles-young* fact for kangaroos. A substantial number of responses used *animal(X)* instead of *mammal(X)* in their final rule, demonstrating a limited understanding of the need to create a new rule to solve this problem.

(c) (i) Many weaker responses confused the object oriented programming paradigm with a programming language that supports a particular interface. Given that candidates could choose any concept related to OOP in their response, it should have been relatively easy to select a concept with which they were familiar. Unfortunately many of those who chose to describe polymorphism demonstrated a very poor understanding of the concept.

(ii) A significant number of weaker responses wrote about abstraction as being the copying of code from one program to another.

Question 25 – The Software Developer's View of the Hardware

(a) (i) Better responses recognised ASCII as a numeric / binary code used to represent characters. Weaker responses simply expanded the acronym.

(ii) Better responses demonstrated the ability to perform the shift and add process in binary. Some responses indicated that candidates found it easier to convert the negative number to a positive, multiply it and then convert it back to a negative. Most responses did not recognise that the answer should also have been in 8 bit two's complement.

(iii) Better responses demonstrated understanding that real numbers have fractional parts that are represented using negative powers of two, and that also some method of locating the radix point is required. They included a description of the features of fixed and floating point representations and the limitations of each. Weaker responses focussed heavily on representation of negative or very large integers, ignoring non-integers.

(b) (i) Better responses recognised a logic gate as a piece of hardware that processes Boolean data. Some responses described the gate as a diagram or process, others failed to indicate its purpose.

(ii) Responses that provided circuit diagrams were better able to distinguish between the operation of the two devices. Weaker responses often indicated the incorrect number of input and/or output bits.

(iii) Better responses included truth tables.

(iv) Many responses correctly recognised the need for a NOT gate. Some responses indicated some confusion with the question and were looking to add a gate that changed just a single line of the truth table. Candidates are reminded to use standard representations for logic gates in their responses.

(c) (i) A number of responses simply listed rather than described the main components of a data stream. Weaker responses were often specifically focused on transmission of a byte rather

than a more general data stream. Better responses clearly described the components of the data stream.

(ii) Weaker responses had difficulty explaining how control characters in a data stream are used to control a printer and tended to focus on the communication process/protocol rather than control of the appearance of the output. Better responses clearly explained the use of control characters and provided examples of how control characters can be used to change typeface or colour or move to new lines or pages to support their discussion.

Software Design and Development

2007 HSC Examination Mapping Grid

Question	Marks	Content	Syllabus outcomes
Section I			
1	1	9.2.2, 9.2.3	H4.2, H4.3
2	1	9.2.3	H4.2, H4.3
3	1	9.2.3	H4.3, H4.2
4	1	9.1.2	H2.2
5	1	9.2.3	H1.1
6	1	9.2.3	H4.2, H4.3
7	1	9.2.2	H4.2, H4.3
8	1	9.2.2, 9.2.3	H4.2, H4.3
9	1	9.1.2	H5.1, H5.3
10	1	9.2.4	H4.2, H4.3
11	1	9.2.1	H4.2, H4.3, H5.2
12	1	9.1.2	H5.1, H5.2
13	1	9.2.3	H4.2, H4.3
14	1	9.2.3	H6.2
15	1	9.2.3	H4.2, H4.3, H6.2
16	1	9.2.2, 9.2.3	H4.2, H4.3
17	1	9.2.3	H4.2, H4.3
18	1	9.1.1	H3.1
19	1	9.2.1	H4.2, H4.3
20	1	9.2.1	M4.2, H4.3
Section II			
21 (a) (i)	1	9.2.4	H6.2
21 (a) (ii)	2	9.2.4	H6.2
21 (b) (i)	3	9.1.2	H4.2
21 (b) (ii)	3	9.1.2	H4.2
21 (c) (i)	2	9.2.4	H4.2
21 (c) (ii)	3	9.2.5	H4.2
21 (d) (i)	2	9.2.2	H4.2
21 (d) (ii)	4	9.2.3	H4.2
22 (a)	2	9.2.3	H1.2, H4.2, H5.1, H6.3
22 (b)	3	9.2.3, 9.2.4	H4.2, H5.1, H5.3
22 (c)	2	9.2.1	H4.3, H5.2, H6.4
22 (d) (i)	2	9.2.3	H1.1, H1.3

Question	Marks	Content	Syllabus outcomes
22 (d) (ii)	2	9.2.3	H1.1, H1.3
22 (d) (iii)	4	9.1.1	H3.1
22 (e) (i)	2	9.2.2	H4.2, H4.3
22 (e) (ii)	3	9.2.2	H4.2, H4.3
23 (a) (i)	2	9.2.4, 9.3	H4.2, H4.3
23 (a) (ii)	3	9.2.4, 9.3	H4.2, H4.3
23 (b) (i)	2	9.2.3, 9.3	H4.2, H4.3, H1.3
23 (b) (ii)	2	9.2.3, 9.3	H4.2, H4.3, H1.3
23 (b) (iii)	4	9.2.1, 9.2.2, 9.2.3, 9.3	H4.2, H4.3, H5.2
23 (c) (i)	2	9.2.2, 9.2.1	H4.2, H4.3, H5.2
23 (c) (ii)	2	9.2.2, 9.2.1	H4.2, H4.3, H5.2
23 (c) (iii)	3	9.2.2, 9.2.1	H4.2, H4.3, H5.2
Section III			
24 (a) (i)	1	9.4.1	H2.1, H4.1
24 (a) (ii)	2	9.4.1	H2.1, H4.1
24 (a) (iii)	3	9.4.1	H2.1, H4.1
24 (b) (i)	1	9.4.1	H2.1, H4.1
24 (b) (ii)	2	9.4.1	H2.1, H4.1
24 (b) (iii)	2	9.4.1	H2.1, H4.1
24 (b) (iv)	3	9.4.1	H2.1, H4.1
24 (c) (i)	3	9.4.1	H2.1, H4.1
24 (c) (ii)	3	9.4.1	H2.1, H4.1
25 (a) (i)	1	9.4.2	H1.3
25 (a) (ii)	2	9.4.2	H1.3
25 (a) (iii)	3	9.4.2	H1.3
25 (b) (i)	1	9.4.2	H1.1
25 (b) (ii)	2	9.4.2	H1.3
25 (b) (iii)	2	9.4.2	H1.3
25 (b) (iv)	3	9.4.2	H1.3
25 (c) (i)	3	9.4.2	H1.1
25 (c) (ii)	3	9.4.2	H1.1

BOARD OF STUDIES
NEW SOUTH WALES

2007 HSC Software Design and Development Marking Guidelines

Section II

Question 21 (a) (i)

Outcomes assessed: H6.2

MARKING GUIDELINES

Criteria	Marks
• Provides a definition of quality assurance which demonstrates a basic understanding of the process	1

Question 21 (a) (ii)

Outcomes assessed: H6.2

MARKING GUIDELINES

Criteria	Marks
• Provides a description of how a software developer goes about producing good user documentation	2
• Identifies a feature or process involved with good user documentation	1

Question 21 (b) (i)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Provides justification for the use of a RAD approach relating to this scenario	3
• Provides a description demonstrating some understanding of the nature of the RAD approach	2
• Identifies a characteristic of the RAD approach	1

Question 21 (b) (ii)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Provides an explanation demonstrating a good understanding of the steps involved in prototyping	3
• Provides a description demonstrating a reasonable understanding of the steps involved in prototyping	2
• Identifies a feature of the prototyping approach	1

Question 21 (c) (i)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Provides a description of the error, its location and a solution	2
• Identifies an error	1

Question 21 (c) (ii)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Provides a description of a method indicating a good understanding of the problem	3
• Provides a description of a method indicating a reasonable understanding of the problem	2
• Identifies an item which demonstrates some understanding of the problem	1

Question 21 (d) (i)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Provides a design for a data structure using appropriate variable names	2
• Identifies an item indicating some understanding of the problem	1

Question 21 (d) (ii)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Provides an evaluation which demonstrates a good understanding of the difference between sequential and direct file access for this problem	4
• Provides a discussion which demonstrates a reasonable understanding of the difference between sequential and direct file access for this problem	3
• Provides a description indicating some understanding of file access	2
• Identifies a feature of file access	1

Question 22 (a)

Outcomes assessed: H1.2, H4.2, H5.1, H6.3

MARKING GUIDELINES

Criteria	Marks
• Provides a definition, demonstrating an understanding of peer checking	2
• Identifies a feature of peer checking	1

Question 22 (b)

Outcomes assessed: H4.2, H5.1, H5.3

MARKING GUIDELINES

Criteria	Marks
• Provides an explanation of the use of breakpoints and single line stepping	3
• Provides a description of the use of breakpoints and single line stepping	2
• Identifies a feature indicating a limited understanding of software debugging tools	1

Question 22 (c)

Outcomes assessed: H4.3, H5.2, H6.4

MARKING GUIDELINES

Criteria	Marks
• Provides an explanation demonstrating an understanding of the need for screen design in the first phase	2
• Identifies a factor relating to screen design	1

Question 22 (d) (i)

Outcomes assessed: H1.1, H1.3

MARKING GUIDELINES

Criteria	Marks
• Provides a definition and outline, demonstrating an understanding of the role of a program counter	2
• Identifies a factor indicating a limited understanding of a program counter	1

Question 22 (d) (ii)

Outcomes assessed: H1.1, H1.3

MARKING GUIDELINES

Criteria	Marks
• Provides an explanation demonstrating an understanding of the problem	2
• Identifies a factor indicating a limited understanding of the problem	1

Question 22 (d) (iii)

Outcomes assessed: H3.1

MARKING GUIDELINES

Criteria	Marks
• Constructs a scenario demonstrating a clear understanding of the problem	4
• Provides a discussion demonstrating an understanding of the problem	3
• Provides a description demonstrating a limited understanding of the problem	2
• Identifies a factor indicating an elementary understanding of the problem	1

Question 22 (e) (i)

Outcomes assessed: H4.2, H4.3

MARKING GUIDELINES

Criteria	Marks
• Provides a description demonstrating an understanding of the problem	2
• Identifies an error	1

Question 22 (e) (ii)

Outcomes assessed: H4.2, H4.3

MARKING GUIDELINES

Criteria	Marks
• Constructs an algorithm which is substantially correct	3
• Identifies items indicating some understanding of the problem and an understanding of control structures	2
• Identifies items indicating a limited understanding of the problem	1

Question 23 (a) (i)

Outcomes assessed: H4.2, H4.3

MARKING GUIDELINES

Criteria	Marks
• Provides definition of test data demonstrating understanding of the term	2
• Identifies a feature of test data	1

Question 23 (a) (ii)

Outcomes assessed: H4.2, H4.3

MARKING GUIDELINES

Criteria	Marks
• Provides explanation of the use of live test data indicating understanding of the process	3
• Provides a description of the use of live test data	2
• Identifies a feature of the use of live test data	1

Question 23 (b) (i)

Outcomes assessed: H4.2, H4.3, H1.3

MARKING GUIDELINES

Criteria	Marks
• Provides description of a purpose of isprime	2
• Identifies a feature of isprime	1

Question 23 (b) (ii)

Outcomes assessed: H4.2, H4.3, H1.3

MARKING GUIDELINES

Criteria	Marks
• Provides a description of a problem in Checkprime indicating an understanding of the algorithm	2
• Identifies a feature of Checkprime	1

Question 23 (b) (iii)

Outcomes assessed: H4.2, H4.3, H5.2

MARKING GUIDELINES

Criteria	Marks
• Constructs a sound algorithm that demonstrates understanding of the problem	4
• Constructs an algorithm that demonstrates understanding of the problem	3
• Attempts an algorithm indicating some understanding of control structures, with reference to the problem	2
• Identifies items indicating a limited understanding of the problem	1

Question 23 (c) (i)

Outcomes assessed: H4.2, H4.3, H5.2

MARKING GUIDELINES

Criteria	Marks
• Identifies processes in the data flow diagram	2
• Identifies a process	1

Question 23 (c) (ii)

Outcomes assessed: H4.2, H4.3, H5.2

MARKING GUIDELINES

Criteria	Marks
• Provides an explanation indicating understanding of the hierarchical nature of a set of data flow diagrams	2
• Identifies a feature of data flow diagrams	1

Question 23 (c) (iii)

Outcomes assessed: H4.2, H4.3, H5.2

MARKING GUIDELINES

Criteria	Marks
• Constructs a data flow diagram that demonstrates understanding of the problem	3
• Identifies features indicating some understanding of the problem	2
• Identifies features of the problem	1

Question 24 (a) (i)

Outcomes assessed: H2.1, H4.1

MARKING GUIDELINES

Criteria	Marks
• Provides a definition of a function	1

Question 24 (a) (ii)

Outcomes assessed: H2.1, H4.1

MARKING GUIDELINES

Criteria	Marks
• Provides a substantially correct evaluation showing working	2
• Provides an evaluation	1

Question 24 (a) (iii)

Outcomes assessed: H2.1, H4.1

MARKING GUIDELINES

Criteria	Marks
• Provides a discussion indicating an understanding of historical reason for the development of the functional paradigm	3
• Provides a description indicating some understanding of an historical reason for the development of the functional paradigm	2
• Identifies a reason for the development of the functional paradigm	1

Question 24 (b) (i)

Outcomes assessed: H2.1, H4.1

MARKING GUIDELINES

Criteria	Marks
• Provides a definition of a fact	1

Question 24 (b) (ii)

Outcomes assessed: H2.1, H4.1

MARKING GUIDELINES

Criteria	Marks
• Distinguishes between forward and backward chaining	2
• Identifies a feature of forward or backward chaining	1

Question 24 (b) (iii)

Outcomes assessed: H2.1, H4.1

MARKING GUIDELINES

Criteria	Marks
• Provides a description indicating an understanding of rules in the logic paradigm	2
• Identifies a feature of rules in the logic paradigm	1

Question 24 (b) (iv)

Outcomes assessed: H2.1, H4.1

MARKING GUIDELINES

Criteria	Marks
• Provides modification that demonstrates a good understanding of the problem	3
• Identifies features indicating an understanding of the problem	2
• Identifies a feature indicating a limited understanding of the problem	1

Question 24 (c) (i)

Outcomes assessed: H2.1, H4.1

MARKING GUIDELINES

Criteria	Marks
• Provides a description demonstrating a good understanding of a concept of the object oriented paradigm	3
• Provide a description demonstrating a limited understanding of a concept of the object oriented programming paradigm	2
• Identifies a feature of the object oriented paradigm	1

Question 24 (c) (ii)

Outcomes assessed: H2.1, H4.1

MARKING GUIDELINES

Criteria	Marks
• Provides an explanation demonstrating a good understanding of abstraction	3
• Provides a discussion demonstrating an understanding of abstraction	2
• Identifies a feature of abstraction	1

Question 25 (a) (i)

Outcomes assessed: H1.3

MARKING GUIDELINES

Criteria	Marks
• Provides a definition of ASCII	1

Question 25 (a) (ii)

Outcomes assessed: H1.3

MARKING GUIDELINES

Criteria	Marks
• Provides a substantially correct evaluation showing working	2
• Provides an evaluation	1

Question 25 (a) (iii)

Outcomes assessed: H1.3

MARKING GUIDELINES

Criteria	Marks
• Provides a discussion indicating a good understanding of binary representation of real numbers	3
• Provides a description indicating some understanding of binary representation of real numbers	2
• Identifies a feature of binary representation of real numbers	1

Question 25 (b) (i)

Outcomes assessed: H1.1

MARKING GUIDELINES

Criteria	Marks
• Provides a definition for a Logic Gate	1

Question 25 (b) (ii)

Outcomes assessed: H1.3

MARKING GUIDELINES

Criteria	Marks
• Distinguishes between a half adder and a full adder in terms of what each does	2
• Identifies a feature of either a half adder or full adder	1

Question 25 (b) (iii)

Outcomes assessed: H1.3

MARKING GUIDELINES

Criteria	Marks
• Provides a description of the operation of a XOR logic gate	2
• Identifies a feature of an XOR logic gate	1

Question 25 (b) (iv)

Outcomes assessed: H1.3

MARKING GUIDELINES

Criteria	Marks
• Provides a modification that demonstrates understanding of the problem	3
• Identifies features indicating an understanding of the problem	2
• Identifies a feature indicating a limited understanding of the problem	1

Question 25 (c) (i)

Outcomes assessed: H1.1

MARKING GUIDELINES

Criteria	Marks
• Provides a description demonstrating a good understanding of the components of a data stream	3
• Provides a description demonstrating a limited understanding of a data stream	2
• Identifies a feature of a data stream	1

Question 25 (c) (ii)

Outcomes assessed: H1.1

MARKING GUIDELINES

Criteria	Marks
• Provides an explanation demonstrating how a data stream can be used to control a printer	3
• Provides a discussion demonstrating a limited understanding of how a data stream can be used to control a printer	2
• Identifies elements to be controlled by a data stream	1