2005 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 1741473314

2006014

Contents

2005 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 provides comments with regard to responses to the 2005 Higher School Certificate examination, indicating the quality of candidate responses and highlighting the relative strengths and weaknesses of the candidature in each section and each question.

This document should be read along with the relevant syllabus, the 2005 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 2005, approximately 2100 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	C
2	C
3	C
4	D
5	B
6	C
7	A
8	A
9	D
10	C

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

Section II

General Comments

Many candidates, as in past years, showed an 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) Most candidates gave a definition for the phased method of implementation. Better responses also included a diagram to enhance their answer.

(ii) This question was generally well answered. Many candidates gave multiple reasons for using phased implementation and related the reason to a general scenario. In the better responses, candidates related their reasons to a realistic/specific situation.

(b) (i) Most candidates identified the conditions of the while loop and described what each did. The better responses also described why the conditions were necessary, for example, the condition $b \leq n/2$ loops until the middle letter of the word is reached. Better responses explained that we needed to loop for only half the number of letters because 2 letters are checked at a time on each loop.

(ii) This question asked that a logic error be identified using the word BLOB. In the better responses, candidates used the word BLOB and performed a desk check or walk through to describe where and what the logic error was. These candidates were then able to suggest an appropriate fix which would work for all words rather than just BLOB. Most candidates were able to say what the output would be. Some of the weaker responses attempted to change the condition of the WHILE loop in part 5 rather than the IF condition in part 6. Others suggested errors which were not logical errors.

(c) (i) Most candidates named a standard sort and described some of its features. In the better responses, candidates selected aspects of the student's sort and a standard sort that were similar and different while making sure that they clearly demonstrated their understanding of the standard sort.

(ii) In the better responses, candidates recognised that one loop was required which compared elements of the first array with the elements of the second array and then placed the lowest element (alphabetically) into the final array. In the best responses, candidates also included array bound-checks.

In the weaker responses, candidates did not attempt to use arrays and their indexes, and in some cases candidates demonstrated little or any knowledge of how to correctly structure pseudocode.

Question 22

(a) (i) Weaker responses included a discussion on how the game would be structured rather than the scheduling of the game's development. Good responses related the time needed to develop the game and linked this with a description as to whether it would be feasible; for example, describing whether the game would be able to be developed for Christmas.

(ii) Good responses correctly described the features of event-driven languages that made it suitable to the development of the adventure game. For example, some included a discussion on how an event such as clicking on a button triggers the relevant code. Weaker responses showed little understanding of the difference between event-driven and sequential languages. Many candidates described how to play the adventure game with only superficial links to the features of event-driven and sequential approaches. Many candidates incorrectly indicated that the user had no control whatsoever in games developed with the sequential approach.

(iii) The vast majority of candidates were able to at least identify appropriate issues with the hardware. Good responses described a hardware issue and discussed how the game would need to be modified to suit the difference in the hardware.

(b) (i) The majority of responses attracted full marks on this part. However, some candidates did not identify the variables in the algorithm. Weaker responses included a description of the data type (eg true or false for Boolean), rather than describing the function of the variable.

(ii) This question was poorly answered by many candidates. Weaker responses discussed the function of the CountFishFingers subroutine without identifying the parameter or its purpose. Some candidates only identified the parameter (fingers) and its purpose as initialising the variable fingers to 0, but did not justify the necessity of passing the parameter. In the better responses, candidates analysed the necessity of passing the parameter into the routine. Some candidates incorrectly thought that because fingers was a parameter it had to be a global variable.

(iii) Most candidates understood the need for a counter. Weaker responses did not initialise the counter, or placed the initialisation in the incorrect position.

(iv) The majority of candidates recognised that the data needs to be saved to a permanent storage (eg file or database) on a daily basis. Good responses outlined a solution for the redesigned system. They outlined how a maximum, minimum and total value could be found for a month, for example, by using a sort or linear search of the month's data to find the minimum number of packed fingers. Weaker responses simply listed the new functions of the system, for example, finding the minimum number of packed fingers. Many candidates did not recognise that arrays of records are relevant in primary storage, whereas files are relevant in secondary storage.

Question 23

(a) (i) Better responses to this question defined reverse engineering as the process of analysing a completed product using a variety of means which may or may not include decompilation. They also identified essential features of the process, such as the purpose of reverse engineering. Poorer responses regarded reverse engineering as simply decompilation.

(ii) Many candidates recognised licensing agreements which allow reverse engineering under the Copyright Act such as public domain and freeware licences. Poorer responses confused purchase of a licence to use the software with ownership of the copyright. Some candidates discussed specific court cases in which individuals managed to circumvent copyright rather than referring to legal instances. A small number of candidates described an illegal situation instead of a legal one, as required by the question.

(b) (i) Many candidates recognised language as one of the areas to be considered in this question. Where this was the case, most of the responses were well written and described a second issue relevant to the scenario. A large number of candidates described issues associated with inclusivity in general rather than issues related to the examination scenario which specified the need for international acceptance of a piece of software.

(ii) Many candidates recognised that the use of modules in this scenario would allow for the parts of the program related to culture to be defined in a separate routine which could be easily edited or replaced as required. Some candidates confused a modular approach to design with the structured programming approach.

(c) (i) Most candidates correctly recognised one input and one output of this system. A small number of responses named input or output devices, or confused inputs and outputs with processes.

(ii) Better responses identified a computer process from the scenario and also included a detailed description of the steps involved in that process. Many candidates simply rewrote parts of the scenario given rather than attempting to provide a more specific description of a single computer process. Some candidates incorrectly chose to describe a manual process rather than a computer process.

(iii) Despite being provided with the symbols used in a system flow chart, most candidates did not analyse the scenario in sufficient detail to achieve full marks in this question. The better responses included only those symbols which are valid for a system flowchart, while poorer responses confused a system flowchart with an algorithm flowchart, including such symbols as BEGIN and END. These responses did not account for input from files or output to files.

Section III

Question 24 – Evolution of Programming Languages

(a) (i) Many candidates did not answer the question in terms of the given logic paradigm. Instead they answered in general terms that related to genealogy rather than software design. Weaker responses used background knowledge to deduce facts that were not present in the provided code segment.

(ii) Good responses identified BOTH answers to the query, whilst weaker responses only identified ONE answer.

(iii) Many candidates did not answer the question in terms of the given logic paradigm, but rather answered in general terms relating to genealogy as opposed to the design of the software. Good responses included a discussion of chaining of rules or recursion. Weaker responses attempted to explain the two cases represented by the rules independently, without linking them or indicating that they were linked.

(iv) The majority of candidates could define the 'male' fact. In the better responses, candidates attempted to define a rule for 'son (X,Y)' whilst in the weaker responses, candidates expressed this merely as a series of facts. Many responses also confused the relationship between the X and Y in 'son (X,Y)' with that of X and Y in 'parent (X, Y)'. Better candidates followed the example given in the comment for the parent rule. Some candidates misidentified the comment on line 3 and confused it with an actual RULE.

(b) (i) Examples provided needed to be appropriate to the functional paradigm.
Good responses provided sound examples with clear coded samples of a related function that demonstrated a good understanding of the functional paradigm.
Weaker responses only gave a general description or outline of a function which demonstrated only a limited understanding.

Some candidates identified poor examples and did not relate them to a function. These responses demonstrated a poor understanding of the features of the functional paradigm.

(ii) This question required candidates to discuss 'functions' in general, NOT specifically the functional paradigm. Candidates needed to relate the features of functions to their effect on programmer productivity. Many candidates identified areas of productivity from those specified in the course syllabus. Some candidates demonstrated a poor understanding of the features of the functional paradigm or did not link these features appropriately to their effect on programmer productivity.

Candidates were not required to compare functions or the functional paradigm to other paradigms as this was not specified in the question.

(c) (i) Most candidates provided an appropriate declaration of the 'num-Borrowed' variable. In the weaker responses, candidates did not identify an appropriate data-type or declared the variable in the 'public' section of the class rather than 'private'.

For questions such as this one, candidates are reminded not to re-write the entire code fragment but rather to use references to code line numbers to express the placement of their answer within the code. Some answers were overly verbose and went into far too much depth.

(ii) Many candidates demonstrated a reasonable understanding of encapsulation. Better responses linked the features of encapsulation to maintenance and also used the BOOK class in the question as an example to support their answer. Weaker responses did not demonstrate as thorough an understanding of encapsulation and tended to focus too much on maintenance issues.

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

(a) (i) Although a number of candidates described in general terms where the EOD character would fit (at the end of the message), significantly fewer candidates were familiar with the required syllabus term 'trailer'.

(ii) Many candidates included the wording from the question as stated in the description for the control character OUTP, attracting few if any marks. Better candidates demonstrated their understanding of the protocols required for communication between devices, where it is necessary to first send an initialising control character to the device indicating that data is to follow. In the better responses, candidates discussed the importance of this process, describing how it could otherwise cause valuable data items to be lost if the printer was otherwise occupied or in 'sleep mode' when the data was sent.

(iii) Most candidates described how the MEOP control character indicated the end of one page, meaning that it was not necessary to also send an MSOP control character to then define the start of the next. Fewer candidates recognised that at the end of a document, the EOD control character also implicitly specifies the end of the page.

(iv) Most candidates demonstrated their understanding of how to extend the protocol by creating new control characters, together with a description for each one provided. Most candidates recognised the need for at least two new control characters specifying that printing be stopped and only resumed when paper was again present. Fewer candidates described that the protocol required a control character to be sent from the printer to the computer when it sensed that it was out of paper, asking that no more packets of data be sent. Once the paper tray was refilled, the printer would sense that printing could continue, and so it would send another control character to the computer requesting that packets be sent again.

(b) (i) Most candidates were familiar with the binary representation of numbers, and correctly stated that the first bit represents the sign of the number (positive or negative). A significant number of candidates recognised that the first two numbers were positive, but the derived answer was apparently negative. Most of these candidates also described the problem, in that the question specified the multiplication of 3 by 5, but the answer obtained is apparently -7. The better candidates stated the reason for this apparent error, which was because there were not enough bits allocated to hold the full answer. Only 4 bits were allocated, but the answer obtained in this example is +15, which requires at least 5 bits (01111) to represent it correctly in binary.

(ii) Candidates are reminded that an understanding of floating point representation is an important part of this option. A significant number of candidates did provide a description of how floating point representation is implemented. Candidates are reminded to refer to the glossary of key words which provides definitions of terms such as *discuss*, *explain* or *outline*.

(c) (i) This part was answered well by the majority of candidates. Some provided a partial truth table to indicate the result of passing the signal 1000 through the circuit, whilst others provided a fully labelled circuit diagram. Either method demonstrated candidates' understanding of the concepts required in the question and their knowledge of AND, OR and NAND gates.

(ii) This question resulted in some wonderfully creative circuit designs, many of which solved the stated problem very well. Candidates who seemed to have solved the problem with minimal fuss were able to verbalise the circuit requirements as NOT A and B and C and D, which resulted in an excellent solution. A significant number of candidates did not demonstrate a sufficiently methodical approach to solving problems of this type, requiring pages of trial and error attempts. Some candidates produced some complex solutions without any apparent working. It is important to remember that marks are awarded for the level of knowledge and understanding demonstrated in questions such as this, and evidence of 'working' either through a narrative discussion or truth tables should be shown.

Software Design and Development

2005 HSC Examination Mapping Grid

Question	Marks	Content	Syllabus outcomes
Section I			
1	1	9.2.1	H6.3
2	1	9.1.2	H4.2
3	1	9.2.3	H4.3
4	1	9.2.5	H6.4
5	1	9.2.2	H5.2
6	1	9.2.4	H6.2
7	1	9.2.1	H4.1
8	1	9.2.3	H4.3
9	1	9.3	H6.2
10	1	9.2.3	H6.2
11	1	9.2.2	H5.3
12	1	9.2.3	H1.3
13	1	9.3	H5.2
14	1	9.2.3	H1.1
15	1	9.2.2	H1.3
16	1	9.2.4	H5.1
17	1	9.2.2	H4.3
18	1	9.3	H5.1
19	1	9.3	H5.2
20	1	9.2.3	H1.3

Question	Marks	Content	Syllabus outcomes
Section II			
21 (a) (i)	2	9.1.2	H5.1
21 (a) (ii)	4	9.1.2	H5.1
21 (b) (i)	3	9.2.1/9.2.2	H5.2
21 (b) (ii)	3	9.2.3	H5.2
21 (c) (i)	3	9.2.3/9.2.2	H4.1
21 (c) (ii)	5	9.2.2	H4.1, H5.2
22 (a) (i)	2	9.2.1	H5.2
22 (a) (ii)	3	9.2.2	H4.2
22 (a) (iii)	3	9.1.2	H5.2
22 (b) (i)	3	9.2.2	H4.3
22 (b) (ii)	3	9.2.2, 9.3	H4.3
22 (b) (iii)	2	9.2.2	H4.2, H4.3
22 (b) (iv)	4	9.2.2	H4.2
23 (a) (i)	2	9.1.1	H1.2
23 (a) (ii)	2	9.1.1	H3.1
23 (b) (i)	3	9.1.1	H3.1
23 (b) (ii)	3	9.2.5	H4.3
23 (c) (i)	2	9.2.1	H4.2
23 (c) (ii)	3	9.2.1	H4.2
23 (c) (iii)	5	9.2.1	H4.2
Section III			
24 (a) (i)	1	9.4.1	H1.2
24 (a) (ii)	2	9.4.1	H1.2
24 (a) (iii)	2	9.4.1	H1.2
24 (a) (iv)	3	9.4.1	H4.2

Question	Marks	Content	Syllabus outcomes
Section III			
24 (b) (i)	3	9.4.1	H4.1
24 (b) (ii)	3	9.4.1	H2.2
24 (c) (i)	2	9.4.1	H4.2
24 (c) (ii)	4	9.4.1	H2.1
25 (a) (i)	1	9.4.2	H1.1
25 (a) (ii)	2	9.4.2	H1.1
25 (a) (iii)	2	9.4.2	H1.1
25 (a) (iv)	3	9.4.2	H3.2
25 (b) (i)	3	9.4.2	H1.3
25 (b) (ii)	3	9.4.2	H1.3
25 (c) (i)	2	9.4.2	H1.3
25 (c) (ii)	4	9.4.2	H4.1

BOARD OF STUDIES
NEW SOUTH WALES

2005 HSC Software Design and Development Marking Guidelines

Section II

Question 21 (a) (i)

Outcomes assessed: H5.1

MARKING GUIDELINES

Criteria	Marks
• Provides a definition, demonstrating an understanding of phased implementation	2
• Identifies factors indicating a limited understanding of phased implementation	1

Question 21 (a) (ii)

Outcomes assessed: H5.1

MARKING GUIDELINES

Criteria	Marks
• Provides discussion of reasons for the use of phased implementation using realistic example(s) that support the reason(s)	4
• Provides discussion for a reason with an example that has some relationship with the reason	2–3
• Identifies a reason	1

Question 21 (b) (i)

Outcomes assessed: H5.2

MARKING GUIDELINES

Criteria	Marks
• Describes each of the conditions in the WHILE statement (part 5), demonstrating a clear understanding of the purpose for each condition	3
• Describes each of the conditions in the WHILE statement (part 5), demonstrating an incomplete understanding of the purpose for each condition OR • Describes ONE condition in the WHILE statement (part 5) demonstrating a clear understanding of the purpose for the condition	2
• Identifies condition(s) in the WHILE statement	1

Question 21 (b) (ii)

Outcomes assessed: H5.2

MARKING GUIDELINES

Criteria	Marks
• Uses BLOB and a logical method to correctly identify an error AND • Suggests a modification to the algorithm that would fix the error, demonstrating a strong understanding of the problem	3
• Identifies an error in the algorithm AND • Suggests a modification of the algorithm, demonstrating some understanding of the problem	2
• Identifies an error in the algorithm OR • Suggests a modification to the algorithm, demonstrating some understanding of writing code	1

Question 21 (c) (i)

Outcomes assessed: H4.1

MARKING GUIDELINES

Criteria	Marks
• Compares and contrasts the algorithm in the question with another standard sorting algorithm indicating a clear understanding of sorting	3
• Describes the algorithm in the question AND • Describes another standard sorting algorithm	2
• Identifies a sorting algorithm	1

Question 21 (c) (ii)

Outcomes assessed: H4.1, H5.2

MARKING GUIDELINES

Criteria	Marks
• Proposes pseudocode algorithm which will produce the required merges which is correct	5
• Proposes algorithm which demonstrates an understanding of the context of the problem and is substantially correct	4
• Proposes algorithm which demonstrates an understanding of the problem	3
• Proposes algorithm which demonstrates an understanding of developing algorithms correctly	2
• Identifies items that indicate some understanding of algorithms	1

Question 22 (a) (i)

Outcomes assessed: H5.2

MARKING GUIDELINES

Criteria	Marks
• Provides description of one scheduling issue indicating understanding of determining feasibility	2
• Identifies an issue indicating elementary understanding of determining feasibility	1

Question 22 (a) (ii)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Provides suitable justification for the use of event driven language rather than a sequential approach	3
• Describes the features of event driven language and/or sequential approach	2
• Identifies a feature of an event driven language and/or sequential approach	1

Question 22 (a) (iii)

Outcomes assessed: H5.2

MARKING GUIDELINES

Criteria	Marks
• Provides an analysis of one hardware issue indicating understanding of changing the hardware	3
• Provides description of hardware issue relating to the context of the question	2
• Identifies hardware issue relating to the context of the question	1

Question 22 (b) (i)

Outcomes assessed: H4.3

MARKING GUIDELINES

Criteria	Marks
• Prepares a correct data dictionary, in tabular format, with headings variable, type, description	3
• Prepares a substantially correct data dictionary	2
• Prepares a data dictionary demonstrating some understanding of a data dictionary	1

Question 22 (b) (ii)

Outcomes assessed: H4.3

MARKING GUIDELINES

Criteria	Marks
• Analyses the necessity for passing parameters in relation to this problem, indicating a thorough understanding of parameter passing	3
• Describes parameter passing in relation to this problem	2
• Identifies a parameter that is being passed in this problem OR • Identifies issue relating to parameter	1

Question 22 (b) (iii)

Outcomes assessed: H4.2, H4.3

MARKING GUIDELINES

Criteria	Marks
• Writes the algorithm modification to fulfil the requirements in this question	2
• Writes an algorithm modification demonstrating elementary understanding of the algorithm	1

Question 22 (b) (iv)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Proposes a substantially correct solution which satisfies the requirements for the modification of this system	4
• Proposes a solution indicating an understanding of the context of this problem	2–3
• Identifies issues relating to this problem	1

Question 23 (a) (i)

Outcomes assessed: H1.2

MARKING GUIDELINES

Criteria	Marks
• Provides a definition which demonstrates an understanding of reverse engineering	2
• Identifies a feature of the legality of reverse engineering	1

Question 23 (a) (ii)

Outcomes assessed: H3.1

MARKING GUIDELINES

Criteria	Marks
• Provides a description which demonstrates an understanding of the legal use of reverse engineering	2
• Identifies a feature of the legality of reverse engineering	1

Question 23 (b) (i)

Outcomes assessed: H3.1

MARKING GUIDELINES

Criteria	Marks
• Describes two areas a software developer would consider when constructing a software package for the international market, demonstrating a thorough understanding of these issues	3
• Describes an area a software developer would consider when constructing a software package for the international market, demonstrating a thorough understanding of these issues OR • TWO areas with some understanding	2
• Identifies area(s) a software developer would consider when constructing a package for the international market	1

Question 23 (b) (ii)

Outcomes assessed: H4.3

MARKING GUIDELINES

Criteria	Marks
• Explains how a modular approach to software design could help when creating software for other cultures demonstrating a thorough understanding of modular design	3
• Describes how a modular approach could help when creating software for other cultures demonstrating some understanding of modular design	2
• Identifies aspects of modular design in context	1

Question 23 (c) (i)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Identifies one input and one output for the system	2
• Identifies one input or one output for the system	1

Question 23 (c) (ii)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Provides description of a computer process involved in the payroll system indicating strong understanding of the context of the problem and sufficient detail to enable the process to be coded	3
• Provides description of a computer process involved in the payroll system which indicates some understanding of the context	2
• Identifies a process related to the system	1

Question 23 (c) (iii)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Provides system flowchart that correctly represents the manual and computer processes of the system	5
• Provides system flowchart that substantially represents the manual and computer processes of the system	4
• Provides system flowchart that indicates elementary understanding of the context of the problem	3
• Describes graphically a relationship between 2 components, processes or equivalent that indicates some understanding of system flowcharts	2
• Identifies graphically a process, component or equivalent	1

Section III

Question 24 (a) (i)

Outcomes assessed: H1.2

MARKING GUIDELINES

Criteria	Marks
• Identifies a fact	1

Question 24 (a) (ii)

Outcomes assessed: H1.2

MARKING GUIDELINES

Criteria	Marks
• Correctly determines both answers	2
• Shows some understanding of the ancestor rule	1

Question 24 (a) (iii)

Outcomes assessed: H1.2

MARKING GUIDELINES

Criteria	Marks
• Explanation of definitions indicating understanding of recursion including the use of a base case to stop the evaluation	2
• Explanation of definitions indicating some understanding	1

Question 24 (a) (iv)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Provides extension to program that is substantially correct	3
• Provides extension to program that attempts to complete both components and shows some understanding	2
• Provides extension to program that attempts to complete ONE of the components and demonstrates some understanding	1

Question 24 (b) (i)

Outcomes assessed: H4.1

MARKING GUIDELINES

Criteria	Marks
• Provides outline of example of functional paradigm and specific function that indicates good understanding of functional paradigms	3
• Provides outline of example and/or specific function that indicates some understanding of functional paradigm	2
• Identifies feature of functional paradigm	1

Question 24 (b) (ii)

Outcomes assessed: H2.2

MARKING GUIDELINES

Criteria	Marks
• Discussion of the impact on programmer productivity indicating good understanding of functions and programming	3
• Description of the impact on programmer productivity	2
• Identification of issues relating to the impact on programmer productivity	1

Question 24 (c) (i)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Provides substantially correct amendment to add num-borrowed to the variable declarations (location – lines 3-7, syntax)	2
• Provides an amendment which demonstrates some understanding of required task	1

Question 24 (c) (ii)

Outcomes assessed: H2.1

MARKING GUIDELINES

Criteria	Marks
• Provides a substantially correct discussion of encapsulation indicating a thorough understanding of encapsulation in relation to this question	4
• Provides a discussion of encapsulation, indicating an understanding of encapsulation and the issues involved	3
• Provides a discussion of encapsulation and its impact on program maintenance demonstrating some understanding of the issues involved	2
• Identifies features relating to encapsulation	1

Question 25 (a) (i)

Outcomes assessed: H1.1

MARKING GUIDELINES

Criteria	Marks
• Identifies component of message	1

Question 25 (a) (ii)

Outcomes assessed: H1.1

MARKING GUIDELINES

Criteria	Marks
• Demonstrates understanding of the importance of control character in communication with the printer	2
• Identifies the purpose of control character	1

Question 25 (a) (iii)

Outcomes assessed: H1.1

MARKING GUIDELINES

Criteria	Marks
• Explanation of control character redundancy indicating impact of MSOP and EOD	2
• Explanation of control character redundancy indicating impact of MSOP or EOD	1

Question 25 (a) (iv)

Outcomes assessed: H3.2

MARKING GUIDELINES

Criteria	Marks
• Provides extension to communication protocol that includes: – a pause command to stop the packet being printed and send message to computer requiring a resume message return – a resume command to continue printing from the point of break. Uses control characters in context with those included in the question	3
• Provides extension to communication protocol that includes understanding of a pause and resume process	2
• Provides extension to communication protocol that indicates some understanding of protocols	1

Question 25 (b) (i)

Outcomes assessed: H1.3

MARKING GUIDELINES

Criteria	Marks
• Provides description indicating some understanding of data representation including how 4 bit sign and modulus cannot be used to perform this calculation • Provides evidence for above	3
• Provides description indicating some understanding of the problems of using 4 bit sign and modulus to perform this calculation	2
• Demonstrates some understanding of sign and modulus representation	1

Question 25 (b) (ii)

Outcomes assessed: H1.3

MARKING GUIDELINES

Criteria	Marks
Provides discussion of the use of floating point representation of fractions including: • Understanding of floating point • Discussion of implementation issues	3
• Provides description of the use of floating point in the representation of fractions	2
• Demonstrates some understanding of floating point representation	1

Question 25 (c) (i)

Outcomes assessed: H1.3

MARKING GUIDELINES

Criteria	Marks
• Explains how gate converts input to output	2
• Provides explanation which demonstrates some understanding of circuit	1

Question 25 (c) (ii)

Outcomes assessed: H4.1

MARKING GUIDELINES

Criteria	Marks
• Designs a circuit that will convert 0111 to 0 and everything else to 1	4
• Designs a circuit that is substantially correct that converts 0111 to 0 and some other values to 1	3
• Designs a circuit that provides a 0 as an output	2
• Designs a circuit that converts 4 bits to 1 bit output	1