2003 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: www.boardofstudies.nsw.edu.au

ISBN 1 7409 9997 5

2004045

Contents

2003 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 2003 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.

It is essential for this document to be read in conjunction with the relevant syllabus, the 2003 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 2003, approximately 3 300 candidates attempted the Software Design and Development examination, a decrease over last year's candidature. Of these, a greater number of candidates attempted option 2 rather than option 1.

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

Section I

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

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

Section II

General Comments

The 2003 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 candidates, as in 2002, 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.

Specific Comments

Question 21

Most of the candidates responded to all sections of this question and scored well on parts (a) and (c). Parts (b) and (d) were not handled as well. A significant number of candidates performed very well.

(a) Candidates generally performed well on this part of the question. Some had problems deciding on the names of the stages and many of these mixed up the order of the stages. Many candidates had not learned the names used in the HSC course. Some candidates did not have any understanding of how a Gantt chart is represented.

(b) Many candidates failed to apply the use of data dictionary or test data to the modification of an existing program. Some candidates focused on test data in identifying errors rather than being used to test all paths and boundary conditions. Better responses included the idea that the original test data could be used to confirm that the modified program still completed the original task and/or met the original specifications.

(c) (i) Some candidates performed well on this part, however desk checking was a problem for many candidates. Some failed to show any process of stepping through the algorithm. A significant number added the GST to the 'Amount' during their calculations. Some candidates applied their understanding of GST to the question but this did not assist in answering the question asked. Many candidates had problems with the 'ZZZ'. Some recognised it as being non numeric and identified it as an error rather than recognising it as the sentinel value. Candidates who did not recognise the sentinel value, incorrectly tried to add 'ZZZ' to the total.

(ii) Candidates generally performed well on this part. Those who performed well generally used pseudocode rather than a flowchart. Those who had problems generally missed the idea that they had to initialise a counter and increment it within the loop. This also meant that they had no way of correctly calculating the average. Some candidates who rewrote the algorithm made errors in the location of lines or in copying. Some candidates had problems understanding the record notation 'Transaction.amount'. Candidates who attempted to use a flowchart generally made structural errors and performed poorly.

(d) (i) This part was not answered well by the majority of candidates. Many rewrote the question but failed to add any new or relevant information. Many candidates failed to recognise that they had to use their knowledge to 'Justify' their answer. Few candidates found reasons for using this particular CASE tool.

(ii) Candidates who performed well on this part showed a good understanding of how to represent file and record structures using algorithms. The poorer responses involved the use of array syntax and array indexing. Many candidates failed to correctly identify the fields in the records to be tested and stored. Few candidates successfully handled opening, closing, reading and writing files.

Question 22

(a) (i) This question asked the candidates to define the term Outsourcing. Poor responses demonstrated a limited understanding of outsourcing. The best responses defined outsourcing and often included an example.

(ii) Many candidates gave excellent responses to the implications of outsourcing, but did not relate them to the given scenario. Better responses addressed a number of implications, all of which related to the system. Poorer responses mostly identified only one implication of outsourcing and did not relate it to the software solution.

(iii) Better responses for this part were in context to the given scenario. Poor responses were non-contextual. The better recommendations related to areas such as testing, communication, and security. Many responses gave a single recommendation or simply identified reliability issues without recommendations. A number of candidates focused on the system development cycle. The best responses gave a number of recommendations, covering a number of different steps relating to the development of reliable software that was also related to the internal network system.

(b) (i) Candidates were required to interpret two symbols that would appear on a structure chart. A number of candidates, however, related the question to the given structure chart, which was not asked. These responses were often not the required responses as they simply described what they did on the structure chart. Better responses used correct terminology.

(ii) Some candidates showed little understanding of algorithms. Some attempted to locate a logic error OR described an effect on the system OR made an attempt to remove an error. Candidates in this category showed little understanding of sub-programs.

Many candidates demonstrated a good understanding of algorithms but did not fully address the question. They could locate a logic error and correctly remove it without describing it OR they could locate a logic error and describe its effect but they often did not successfully correct the error in the algorithm.

Better responses demonstrated a full understanding of algorithms and how logic flowed through them. These candidates were capable of locating the logic error, describing its effect

well and then successfully removing it from the algorithm using pseudocode.

(iii) This question required candidates to write an algorithm that would 'Accept' and 'Validate' a Password.

Poorer responses demonstrated little understanding of the problem and poor attempts were made at developing an algorithm related to solving the problem.

A number of candidates demonstrated an understanding of the system and how to develop an algorithm. Many algorithms exhibited simple errors such as failing to increment a counter, not iterating the correct number of times, or not exiting the loop when the password was validated.

A significant number of candidates demonstrated a good understanding of the system and developed error-free algorithms.

Question 23

(a) This question was aimed at testing the candidates' understanding of the similarities and differences between custom-designed software and customised off-the-shelf software.

Good responses demonstrated how the two methods of delivering a solution were similar and different, and related to the development of a database for a small company. Most candidates were able to show differences but few candidates showed similarities. Weaker responses included listing characteristics of the two methods without necessarily comparing or contrasting. Some candidates listed advantages and/or disadvantages rather than similarities and differences, indicating a lack of understanding of the terms 'compare' and 'contrast'.

(b) (i) Most candidates were able to identify the internet as a hardware development. Better responses gave details about the actual hardware and software that is required for high speed, global communications.

The fact that many candidates confused open-source software development with the provision of freeware and shareware software, highlights the need to carefully read the stimulus material before framing a response.

(ii) Better responses to this part recognised that the focus was management of a global project and discussed multiple project management issues related to the scenario.

Many candidates focused their responses on social and ethical issues rather than project management issues.

(c) (i) Most candidates recognised the fact that a register is an accumulator. Better candidates named the register that was being used as the accumulator in the stimulus material.

(ii) Some candidates were able to add the two hexadecimal numbers without converting

through binary and/or decimal. Some candidates failed to read the question carefully ('After execution:') and stated the initial value in the memory location.

(iii) Good responses related Reg 3 in this part to Mem 6 in the previous part. Many candidates were able to show their working by indicating the place value of the hexadecimal digits within the number. Poorer responses recognised, for example, that A=10 and 1=1 but then these were added to arrive at 11 or joined to form 101.

(iv) Better responses recognised that the code would require two ADD instructions. The best responses also included the STOP instruction in their code. Some candidates decided to define their own instruction, MULTIPLY, which did not provide a correct answer to the question.

Section III

General Comments

Candidates were to attempt just one question from this section. A small number of candidates attempted both options. Candidates are advised to attempt only one option and to concentrate preparation and examination time to that option.

Question 24 – Evolution of Programming Languages

(a) Candidates were expected to provide an evolutionary context to specific knowledge in order to achieve full marks in questions throughout the option. Many candidates did not give due consideration to relating the scenario to theoretical knowledge. Candidates need to ensure that they have a thorough understanding of the key terms and apply them in providing responses.

(i) The better responses used specific knowledge regarding procedures and functions in both the named paradigms. Good responses clearly defined that functions and procedures are subprograms or modules and that functions return a value. Candidates were required to relate these to the two paradigms.

(ii) Better responses were able to link the characteristics of the functional paradigm development to the needs created by the evolutionary context. Average responses listed the syllabus headings, such as solving new problems, but failed to explain these appropriately for the paradigm. Poor responses often made simple statements such as 'it is simpler to code'.

(b) (i) This question was well answered by the majority of candidates. Candidates were able to list features of a logic paradigm. Better candidates could link all three named features to code samples.

(ii) Generally well answered. Candidates were able to interpret and extrapolate from the sample code.

(c) Many candidates were able to outline the basic building blocks and concepts of OOP and the other paradigms. Better responses were able to relate these basic building blocks and concepts to the scenario. Many candidates failed to both compare and contrast the appropriateness of OOP to the other paradigms. Candidates often just contrasted and failed to also compare how the different paradigms can achieve similar effects. The relevance of the other paradigms in providing solutions to this particular scenario was not as thorough as discussions of OOP's applicability.

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

(a) Better responses were able to show how single precision floating point is used to represent fractional (or non-integer) numbers, however, not all candidates understood the need to make specific reference to the binary format used. Those candidates who used specific examples in their answer found it easiest to explain the nature of the sign, exponent and mantissa.

A significant number of candidates struggled to explain the concept of the 'floating' point necessary to express the number in the format 1.nnn x 2 $^{\text{exponent}}$, and the notion of the missing or hidden 1 when expressing the mantissa (nnn). Weaker responses included simply stating the existence of the sign, exponent and mantissa without attempting to describe their purpose or format.

(ii) This question required candidates to convert a decimal number into hexadecimal format. Many candidates scored well on this question, although some struggled with correct division by 16, or correct selection of 'D' as the hex digit to represent 13_{10} or 1101_2. It is important for all candidates to show all working to allow them to gain partial marks even if their answer is incorrect.

(iii) This question required candidates to demonstrate their understanding of the use of 2's complement in performing subtraction with 2 binary numbers. Candidates who understood the concept of adding the two's complement of a number instead of trying to subtract it had no difficulty with this question. Some candidates did not know how to generate a 2's complement of a number, or tried to generate the 2's complement of both numbers rather than just the one with the negative value to be added.

Candidates need to set out their work, clearly showing each of the intermediate steps, and labelling each step to show what they are doing at each stage. The better responses to this question clearly showed the leftmost 1 dropped off to give the final answer of 0111_2.

(b) (i) This question required candidates to state the purpose of a flip-flop and then explain how its design enables it to store a bit. Many candidates struggled to clearly explain the function of storing a bit in memory, or the elements in its design that allow it to effectively store a bit even when the input signal drops.

Although many candidates included a diagram, it was often only a rough approximation to a correct diagram, or it was not referred to in the explanation given. Better responses used the diagram to explain the function of the feedback loop or latch in allowing the flip-flop to remain in a stable state even after the signal has been received on the set or reset line.

A number of candidates simply produced a rote-learned answer or truth table rather than trying to answer the specific question.

(ii) This question required candidates to produce a truth table for an AND gate from a flowchart, and then to produce a similar flowchart showing the logic for an OR gate. Many candidates scored well on this question, particularly those who constructed flowcharts from

standard control structures. A significant number of candidates produced poorly structured algorithms, particularly with respect to the binary selections required. Candidates who produced a well-structured flowchart scored well.

Poorer responses did not refer to the example flowchart for the AND gate provided in the question, and produced poor attempts at an algorithm. Almost all candidates chose to produce a truth table for the OR gate, even though this was not a requirement.

In terms of the truth table for the AND gate, most candidates were able to produce a perfect response, although some poorer responses did not know the standard format of a truth table.

(c) This question required candidates to demonstrate their understanding of the format of data stream generated from a fingerprint scanner, and to compare and contrast this with the data stream sent to the door to either close or open it.

The better responses to this question included detailed descriptions of the format of the two data streams with particular reference to probable contents of each of the header, data and trailer sections of the first data stream. Good responses referred to start bits, address of fingerprint scanner, length or resolution of the bitmap to follow, and packet number for a multi-packet stream in the header. The data was well described by some candidates in terms of the use of a 1 to represent a ridge and a 0 to represent a valley, whilst the trailer discussion related to the use of error-checking features such as a checksum or CRC and a stop bit to mark the end of the packet or data stream.

Good responses then compared this long and complex data stream to that sent to the door, which probably only contains a small header with a start bit and possibly to indicate that it is a door message, with a Boolean value to specify whether the door is to be opened or kept shut, and a much less stringent error checking mechanism and stop bit.

Weaker responses showed a much more superficial understanding of the actual nature and format of the data stream required in this question, with a number of poor responses simply rewording the question.

Software Design and Development

2003 HSC Examination Mapping Grid

Question	Marks	Content	Syllabus outcomes
Section I			
1	1	9.3 Developing a solution package	H5.2
2	1	9.2.1 Defining and understanding the problem	H4.1, H6.4
3	1	9.1.2 Application of software development approaches	H5.1
4	1	9.2.1 Defining and understanding the problem	H1.3
5	1	9.2.4 Testing and evaluation of software solutions	H4.2
6	1	9.1.2 Application of software development approaches	H4.2
7	1	9.1.2 Application of software development approaches	H4.1, H6.1
8	1	9.1.1 Social and ethical issues	H3.1
9	1	9.2.2 Planning and design of software solutions	H4.3
10	1	9.2.3 Implementation of software solutions	H1.2
11	1	9.2.1 Defining and understanding the problem	H4.3
12	1	9.2.3 Implementation of software solutions	H4.3
13	1	9.2.1 Defining and understanding the problem	H1.2
14	1	9.1.1 Social and ethical issues 9.2 Software development cycle	H3.1, H4.2
15	1	9.2.3 Implementation of software solutions	H4.3
16	1	9.2.1 Defining and understanding the problem	H5.2
17	1	9.2.1 Defining and understanding the problem	H5.2
18	1	9.1.1 Social and ethical issues	H3.1
19	1	9.2.2 Planning and design of software solutions	H4.2
20	1	9.3 Developing a solution package	H5.2

Question	Marks	Content	Syllabus outcomes
Section II			
21(a)	4	9.3 Developing a solution package 9.2.5 Maintenance of software solutions	H5.1, H5.2, H6.3
21(b)	4	9.2.4 Testing and evaluation of software solutions	H5.3
21(c)(i)	3	9.2.5 Maintenance of software solutions	H4.2, H4.3
21(c)(ii)	3	9.2.5 Maintenance of software solutions	H4.2, H4.3
21(d)(i)	2	9.2.2 Planning and design of software solutions 9.2.5 Maintenance of software solutions	H5.3
21(d)(ii)	4	9.2.2 Planning and design of software solutions 9.2.3 Implementation of software solutions	H4.2, H4.3
22(a)(i)	2	9.1.2 Application of software development approaches 9.2.1 Defining and understanding the problem	H2.2, H3.1
22(a)(ii)	3	9.2.1 Defining and understanding the problem	H4.2, H5.2
22(a)(iii)	4	9.1.2 Application of software development approaches 9.2.4 Testing and evaluation of software solutions	H5.2, H5.3
22(b)(i)	2	9.2.1 Defining and understanding the problem	H1.3, H4.2
22(b)(ii)	4	9.2.1 Defining and understanding the problem 9.2.2 Planning and design of software solutions 9.2.3 Implementation of software solutions 9.2.4 Testing and evaluation of software solutions 9.2.5 Maintenance of software solutions	H4.2, H4.3
22(b)(iii)	5	9.2.1 Defining and understanding the problem 9.2.2 Planning and design of software solutions 9.2.5 Maintenance of software solutions	H4.2
23(a)	4	9.1.2 Application of software development approaches	H1.2, H2.2
23(b)(i)	3	9.1.2 Application of software development approaches	H2.2
23(b)(ii)	3	9.1.2 Application of software development approaches 9.3 Developing a solution package	H2.2
23(c)(i)	2	9.2.3 Implementation of software solutions	H1.1, H3.1
23(c)(ii)	2	9.2.3 Implementation of software solutions	H1.1, H1.3
23(c)(iii)	2	9.2.3 Implementation of software solutions	H1.1, H1.3
23(c)(iv)	4	9.2.3 Implementation of software solutions	H1.1, H1.3

Question	Marks	Content	Syllabus outcomes
Section III			
24(a)(i)	3	9.4.1 Evolution of programming languages	H1.2
24(a)(ii)	4	9.4.1 Evolution of programming languages	H2.1
24(b)(i)	3	9.4.1 Evolution of programming languages	H1.2
24(b)(ii)	4	9.4.1 Evolution of programming languages	H4.2
24(c)	6	9.4.1 Evolution of programming languages	H2.2, H4.1
25(a)(i)	3	9.4.2 Software developers' view of hardware	H1.3
25(a)(ii)	2	9.4.2 Software developers' view of hardware	H1.3
25(a)(iii)	2	9.4.2 Software developers' view of hardware	H1.3
25(b)(i)	3	9.4.2 Software developers' view of hardware	H1.1
25(b)(ii)	4	9.4.2 Software developers' view of hardware	H1.1
25(c)	6	9.4.2 Software developers' view of hardware	H4.1

BOARD OF STUDIES
NEW SOUTH WALES

2003 HSC Software Design and Development Marking Guidelines

Question 21 (a)

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

MARKING GUIDELINES

Criteria	Marks
• Produces a graph including all stages, correct structure and correct relationship to time	4
• Produces a graph including some stages + some relationship to time	3
• Produces a graph including some stages	2
• Produces a graph that indicates a basic level of understanding	1

Question 21 (b)

Outcomes assessed: H5.3

MARKING GUIDELINES

Criteria	Marks
• Shows understanding of the terms data dictionary and test data and correctly shows the effect of both of these on the modification of an existing program	4
• Shows understanding of data dictionary and test data and correctly shows the effect of both of these on the development process	3
• Shows an understanding of the term data dictionary or test data and AND • Correctly shows the effect of one of these to the development process OR • Shows an understanding of the terms data dictionary AND test data	2
• Shows an understanding of the term data dictionary OR test data	1

Question 21 (c) (i)

Outcomes assessed: H4.2, H4.3

MARKING GUIDELINES

Criteria	Marks
• Shows an understanding of stepping through variable values and shows all of the correct variable values and correct outputs	3
• Shows an understanding of stepping through variable values and shows some of the correct variable values	2
• Shows an understanding of stepping through variable values	1

Question 21 (c) (ii)

Outcomes assessed: H4.2, H4.3

MARKING GUIDELINES

Criteria	Marks
• Shows a good understanding of the algorithm with use of a correct calculation for average and a counter	3
• Shows an understanding of the algorithm with use of average calculations	2
• Shows a basic understanding of the algorithm	1

Question 21 (d) (i)

Outcomes assessed: H5.3

MARKING GUIDELINES

Criteria	Marks
• Supports the use of this particular CASE tool, demonstrating why it is used	2
• Demonstrates an understanding of a CASE tool	1

Question 21 (d) (ii)

Outcomes assessed: H4.2, H4.3

MARKING GUIDELINES

Criteria	Marks
• Correctly storing the data in results file • Using a selection structure to identify file data to be stored • Reading and writing of correct file • Producing a correctly-terminated loop	4
• Any three points above	3
• Any two points above	2
• Any point above	1

Question 22 (a) (i)

Outcomes assessed: H2.2, H3.1

MARKING GUIDELINES

Criteria	Marks
• States the meaning of the term outsourcing, including an essential feature	2
• Limited definition of outsourcing	1

Question 22 (a) (ii)

Outcomes assessed: H4.2, H5.2

MARKING GUIDELINES

Criteria	Marks
• Discusses implications of outsourcing this particular software solution	3
• Discusses an implication of outsourcing this particular software solution	2
• Identifies an implication of outsourcing	1

Question 22 (a) (iii)

Outcomes assessed: H5.2, H5.3

MARKING GUIDELINES

Criteria	Marks
• Recommends steps that relate to the internal network	3–4
• Identifies steps that can be taken to ensure reliability	1–2

Question 22 (b) (i)

Outcomes assessed: H1.3, H4.2

MARKING GUIDELINES

Criteria	Marks
• Interprets two symbols correctly	2
• Interprets one symbol correctly	1

Question 22 (b) (ii)

Outcomes assessed: H4.2, H4.3

MARKING GUIDELINES

Criteria	Marks
• Location of a logic error, description of effect that indicates an understanding of the algorithm and its operation • Correctly removes the error	4
• Location of a logic error and correctly removes the error OR • Location of a logic error, description of effect AND attempts to remove the error which does not satisfy requirements	3
• Location of a logic error and describes effect that indicates an understanding of the algorithm OR • Makes an attempt to remove an error AND attempts to describe the effect on the system	2
• Location of a logic error OR • Describes effect on the system OR • Makes an attempt to remove an error	1

Question 22 (b) (iii)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Development of an algorithm that will perform the task desired in the question	5
• Development of an algorithm that confirms a strong understanding of the operations of the system and is substantially correct	4
• Development of an algorithm that confirms a level of understanding of the operation of the system but is not substantially correct	3
• Development of an algorithm that indicates some understanding of the system	2
• Development of an algorithm or part of an algorithm that indicates some understanding of algorithm writing	1

Question 23 (a)

Outcomes assessed: H1.2, H2.2

MARKING GUIDELINES

Criteria	Marks
• Shows a good understanding of customised off-the-self software and custom-designed software, showing how they are similar and different	4
• Shows a good understanding of customised off-the-self software and custom-designed software, showing how they are similar or different	3
• Shows an understanding of customised off-the-shelf software and custom-designed software OR • Lists similarities or differences	2
• Shows an understanding of a customised off-the-shelf software or custom-designed software	1

Question 23 (b) (i)

Outcomes assessed: H2.2

MARKING GUIDELINES

Criteria	Marks
• Describes hardware AND software development(s) relating these to this software development approach	3
• Describes hardware OR software development(s) relating these to this software development approach OR • Describes hardware AND software development(s)	2
• Identifies hardware OR software	1

Question 23 (b) (ii)

Outcomes assessed: H2.2

MARKING GUIDELINES

Criteria	Marks
• Discusses multiple management issues and relates them to the scenario in the question	3
• Discusses a management issue and relates it to the scenario in the question	2
• Identifies a management issue	1

Question 23 (c) (i)

Outcomes assessed: H1.1, H3.1

MARKING GUIDELINES

Criteria	Marks
• Correctly identifies Register Reg 3 as the accumulator	2
• Identifies that the accumulator is a register	1

Question 23 (c) (ii)

Outcomes assessed: H1.1, H1.3

MARKING GUIDELINES

Criteria	Marks
• Correct value • Acceptable explanation of how it was obtained	2
• Correct value OR • Can demonstrate a reasonable understanding of ADD and STORE	1

Question 23 (c) (iii)

Outcomes assessed: H1.1, H1.3

MARKING GUIDELINES

Criteria	Marks
• Shows correct conversion from hexadecimal to decimal with adequate working	2
• Correct value, without demonstrating how this was obtained OR • Shows correct understanding of hexadecimal to decimal conversion	1

Question 23 (c) (iv)

Outcomes assessed: H1.1, H1.3

MARKING GUIDELINES

Criteria	Marks
• Logic correct • Correct use of LOAD, ADD, STORE and STOP	4
• Logic correct • Substantially correct use of LOAD, ADD, STORE and STOP	3
• Substantially correct use of LOAD, ADD, STORE and STOP	2
• Shows understanding of logic and/or syntax	1

Question 24 (a) (i)

Outcomes assessed: H1.2

MARKING GUIDELINES

Criteria	Marks
• Produces a detailed explanation of *procedure* and *function* in terms of their use in the named paradigms	3
• Produces a basic explanation of *procedure* or *function*	2
• Provides a rudimentary explanation of *procedure* or *function*	1

Question 24 (a) (ii)

Outcomes assessed: H2.1

MARKING GUIDELINES

Criteria	Marks
• Explains the development of the functional paradigm with some reference to productivity, repetitive programming tasks, solving different problem types and recognition of different basic building blocks and emerging technologies	4
• Describes the development of the functional paradigm with little reference to productivity, repetitive programming tasks, solving different problem types, and recognition of different basic building blocks and emerging technologies	3
• Identifies characteristics of the functional paradigm OR • Identifies reasons for the development of a paradigm	2
• Identifies characteristic of the functional paradigm OR • Identifies reason for the development of a paradigm	1

Question 24 (b) (i)

Outcomes assessed: H1.2

MARKING GUIDELINES

Criteria	Marks
• Identifies the paradigm as *logic*, lists three features and gives examples from the code for each feature identified which justify choice of paradigm	3
• Identifies the paradigm as *logic*, lists one or two features giving examples of either or both from code which justify choice of paradigm OR • Identify the paradigm as *logic*, lists three features of the logic paradigm without examples	2
• Identifies the paradigm as *logic*	1

Question 24 (b) (ii)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Adds lines of code to the rule indicated which *prompts* for AND *accepts* the capital data AND *adds* predicates for the new country, city and capital, using code that reflects a detailed understanding of the example code	4
• Shows a substantial ability to extrapolate from the example code to *prompt* for AND *accept* data AND *add* predicates for two of the new country, OR city OR capital	3
• Shows a basic ability to extrapolate from the example code to *prompt* for OR *accept* data OR *add* one or more predicates for the new data, including at least two processes	2
• Shows a rudimentary ability to extrapolate from the example code to *prompt* for a response OR *accept* data OR *add* data to the knowledge base	1

Question 24 (c)

Outcomes assessed: H2.2, H4.1

MARKING GUIDELINES

Criteria	Marks
• By comparing and contrasting, shows an understanding of the concepts and basic building blocks of OOP and one or more other paradigms showing for each paradigm how it is, or is not appropriate for use in the scenario described	5–6
• Discusses the concepts and basic building blocks of OOP, AND 1 one other paradigm showing links to the scenario OR 2 two or more other paradigms with no reference or tenuous reference to the scenario	3–4
• Identifies a paradigm and provides one or more characteristics of the named paradigm	1–2

Question 25 (a) (i)

Outcomes assessed: H1.3

MARKING GUIDELINES

Criteria	Marks
• Produces a detailed explanation of the representation of a fraction in single precision floating point binary format	3
• Produces a basic explanation of the representation of a fraction in single precision floating point binary format	2
• Provides a rudimentary explanation of the representation of a fraction in single precision floating point binary format	1

Question 25 (a) (ii)

Outcomes assessed: H1.3

MARKING GUIDELINES

Criteria	Marks
• Provides substantially correct solution with working	2
• Shows a rudimentary knowledge of the hexadecimal number system	1

Question 25 (a) (iii)

Outcomes assessed: H1.3

MARKING GUIDELINES

Criteria	Marks
• Provides correct solution, using four-bit binary representation and two's complement	2
• Shows a rudimentary knowledge of two's complement and/or four-bit binary representation of integers	1

Question 25 (b) (i)

Outcomes assessed: H1.1

MARKING GUIDELINES

Criteria	Marks
• Describes the function of flip-flop and provide a sound explanation of how a flip-flop achieves its purpose	3
• Describes the function of a flip-flop, and provides a rudimentary explanation of how it achieves its purpose OR • Provides a sound explanation of how a flip-flop achieves its purpose	2
• Identifies the purpose of a flip-flop	1

Question 25 (b) (ii)

Outcomes assessed: H1.1

MARKING GUIDELINES

Criteria	Marks
• Draws a correct 'AND' gate truth table and produce a flowchart for an 'OR' gate using correct flowchart symbols and logic	4
• Draws a correct 'AND' gate truth table and produce a flowchart which demonstrates a basic knowledge of the logic of an 'OR' gate OR • Produces a flowchart for an 'OR' gate using correct flowchart symbols and logic	3
• Draws a correct 'AND' gate truth table and produce a flowchart which demonstrates a rudimentary knowledge of the logic of an 'OR' gate OR • Produces a flowchart which demonstrates a basic knowledge of the logic of an 'OR' gate	2
• Draws a correct 'AND' gate truth table OR • Produces a flowchart which demonstrates a rudimentary knowledge of the logic of an 'OR' gate	1

Question 25 (c)

Outcomes assessed: H4.1

MARKING GUIDELINES

Criteria	Marks
• Compares and contrasts the header information, data characters and trailer information derived from this scenario	5–6
• Discusses header information, data characters and trailer information in data streams OR • Compares and contrasts two of header information, data characters or trailer information, with reference to the scenario	3–4
• Identifies some of the features of header information, data characters and trailer information in data stream with either no reference or tenuous reference to the scenario OR • Compares and contrasts one of header information, data characters or trailer information, with reference to the scenario	1–2