2001 HSC
Notes from
the Examination Centre
Software Design and Development

Contents

2001 HSC NOTES FROM THE EXAMINATION CENTRE SOFTWARE DESIGN AND DEVELOPMENT

Introduction

This document has been produced for teachers and candidates of the Stage 6 course Software Design and Development. It provides comments with regard to responses to the 2001 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 documentation should be read in conjunction with the Software Design and Development Stage 6 Syllabus (1999) and the Software Design and Development Stage 6 Software and Course Specifications (2001) and other Software Design and Development support documents developed by the Board of Studies and available on the Board's website.

A set of marking guidelines was developed at the same time as the examination paper. These guidelines were used as the basis for marking all candidate scripts during the marking operation.

Marking of Software Design and Development candidate scripts was carried out by eight teams of 6 to 8 markers, each team lead by a Senior Marker. Senior Markers were briefed on the application of the marking guidelines and read a large number of scripts to become familiar with the application of these guidelines. Senior Markers selected a wide range of candidate responses for each question to use as examples during briefing of Markers. Copies of theses scripts were annotated with explanatory notes as to what mark each response would receive and why. The guidelines and annotated scripts were used to brief Markers prior to pilot marking.

Written Examination

General Comments

In 2001, 3330 candidates presented for the Software Design and Development written examination. Of these candidates approximately 43% attempted Question 24 (Evolution of Programming Languages option) and 57% attempted Question 25 (The Software Developer's View of the Hardware option).

The 2001 Higher School Certificate Examination in Software Design and Development focused on the application of concepts to given scenarios. This required candidates to interpret and analyse situations and to apply their knowledge to those situations, as opposed to rote recall of facts. In all questions a significant number of candidates showed that they were able to recall definitions and to describe concepts, but were unable to either analyse a situation for its key elements or apply the appropriate knowledge in that situation.

Candidates should be given extensive opportunities to encounter key concepts in a variety of contexts and to have extensive practice in analysing given scenarios.

There were many cases in which candidates did not distinguish between key words, for example, 'justify', 'describe', 'assess' and 'identify'. There was a disturbing number of candidates who relied on defining and describing where the question required a more elaborate response.

Section I – Multiple Choice

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

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

Section II

General comments

In this section, many candidates were able to give fairly general responses, drawing on their knowledge of concepts and definitions. However, a significant number of candidates had difficulty in filtering this knowledge and applying just what was required in a given scenario. For many candidates a question appeared to be treated as a stimulus to write whatever could be remembered about a particular topic.

Questions which carried 5 or 6 marks were most commonly answered in this 'general' way.

Specific comments

Question 21

(a)   This question required candidates to demonstrate a good understanding of control structures, parameter passing and desk checks. The majority of candidates were able to carry out a desk check, but many had trouble in tracing through values that were passed as parameters. Candidates used a wide variety of methods for presenting a desk check, but those who did well on this question used a table, clearly displayed with all relevant variables shown.

Many candidates were able to identify where the algorithm broke down, but did not make reference to the desk check in explaining their answer, as the question required.

(b)   Candidates were required to demonstrate an understanding of some of the ethical issues in relation to the responsibilities of software developers, and the role of managers in ensuring that developers carry out these responsibilities. Most candidates were able to outline two of the ethical issues, plagiarism and copyright, and were able to identify and describe strategies for management. The discussion of these strategies sofwas generally poor, with few candidates able to provide points for and/or against the use of these strategies. Better answers referred directly to the scenario described in the question.

Question 22

(a)   Most candidates were able to design and draw a reasonable interface for entering data, with the better interfaces including navigational elements, such as return to the main screen and a button to 'commit' the transaction. Good answers also included some indication of the size of the required fields. Many candidates confused the two files and included data entry fields from the inventory file.

There was a wide variety of IPO charts or diagrams presented. Candidates who scored well on this question used the layout described in the Software and Course Specifications document. Many candidates did not identify files as input or output and did not identify error or status messages as outputs.

Few candidates were able to handle the retrieval of data from a random access file, many treating the file as sequential. Those candidates who used flowcharts in describing the algorithm were more likely to present unstructured algorithms and confused logic. Many candidates were apparently unfamiliar with the use of a sentinel value ·999 in this case. Many candidates were unable to manipulate two files and often confused the two.

(b)   This question required candidates to go beyond defining types of documentation and stages of software development. Many candidates were able to identify various types of documentation and to identify and describe various stages in the development process. However few were able to successfully relate documentation to these stages and to justify the use of documentation in that stage.

Candidates who scored lower marks on this question were unable to identify clearly the development stage or stages they were addressing and often used vague terms to describe these stages.

Question 23

(a)   This question was poorly answered, with few candidates able to demonstrate an understanding of BNF and its limitations.

(b)   Most candidates were able to state and describe some general effects of new technologies but did not assess those effects on the everyday usage of the system described in the question.

(c)   Candidates were generally able to outline a range of methods of involvement eg surveys, interviews and observations, but had difficulty in tying these methods to different types of users, with many candidates referring to 'users' in very general terms. Those candidates who indicated that, for example, doctors and receptionists would have different skills and different needs and would be involved in different ways, generally did well on this question.

Most candidates were able to identify and describe several methods of software development. Better answers evaluated the appropriateness of these methods in the situation described in the question, including the short development time and the need to have the major, but not necessarily all, components operating.

Section III

General comments

Candidates were to attempt just one question from this section. About 40 Candidates attempted both options, generally poorly. Candidates are advised to attempt only one option and concentrate preparation time and examination time on that option.

Questions in this section required candidates to apply knowledge to given situations.

Specific comments

Question 24

(a)   Candidates were required to identify a paradigm and to justify their choice by reference to features of the code fragment. Many candidates were able to identify correctly the fragment but were not able to relate the features of the paradigm to the code.

Many candidates confused 'paradigms' with 'language generations'.

(b)   Many candidates were able to identify several characteristics of Object Oriented languages such as inheritance and encapsulation, but did not link these to reasons for the emergence of the paradigm.

Good responses to this question identified a number of emerging needs and related each to a feature of the Object Oriented paradigm. Better answers also indicated ways in which Object Oriented languages were superior to other languages in meeting those needs.

Many candidates incorrectly identified the Object Oriented paradigm with the presence of screen elements such as windows and buttons.

(c)   This question required candidates to describe a logic error. While many candidates identified a syntax error, better answers clearly identified the error by line number and gave two clearly different methods of correcting the error.

Many candidates seemed to lack a basic understanding of the differences between pre-tested and post-tested loops. Some candidates incorrectly chose to fix the error by using a selection structure in place of the repetition.

Most candidates were able to answer part (ii) satisfactorily, although some candidates only declared the triangle area function, rather than the whole triangle class.

(d)   Some candidates approached this question well, identifying the features of the scenario and justifying the choice of a paradigm by matching the needs identified within the scenario to the strengths of that paradigm.

Many candidates were able to list the features of a paradigm, but did not analyse the scenario.

There was some confusion between language paradigms and language generations.

Question 25

(a) This question was not well done, with few candidates relating ASCII and hexadecimal to binary representations and using that as a means of comparison. There were a number of misconceptions including ASCII being to the base 10 and Hexadecimal only representing numbers.

Better responses made reference to the importance of readability by people, and made use of examples to illustrate the point.

(b) In this question candidates were expected to apply their knowledge of algorithm design and data structures, including arrays of records. The question also required the extraction of data from a data stream. Many candidates were able to analyse what was required although many candidates had trouble in expressing this in appropriate pseudocode.

(c) Candidates who gained full marks in this question showed all their working, including well-constructed and complete truth tables showing how they arrived at the output produced. It was pleasing to see that a large proportion of candidates were able to recognise that 'section X' was equivalent to an XOR gate. A significant proportion of candidates misinterpreted 'section X' as a half-adder, even though there was no notion of a 'carry'.

The majority of candidates recognised the 'flip-flop' and its function, but few demonstrated an understanding of how a 'flip-flop' works by stepping through the workings.

(d) It was pleasing to see how many candidates were able to design an appropriate circuit for this question. Many supplied a correct truth-table, but neglected to verify their solution by identifying why the output was that expected for the given conditions.

Better responses clearly identified the three inputs required, together with a truth table containing the eight relevant entries.

Some candidates had some difficulty with the state of the fault detector.

Software Design and Development

2001 HSC Examination Mapping Grid

Question	Marks	Content		Syllabus outcomes
1	1	9.2.3	Implementation of software solution	H1.1
2	1	9.2.2	Planning and design of software solutions	H1.2
3	1	9.2.3	Implementation of software solution	H4.2
4	1	9.3	Developing a solution package	H4.2
5	1	9.1.2	Application of software development approaches	H5.1
6	1	9.1.2	Application of software development approaches	H4.2
7	1	9.1.2	Application of software development approaches	H2.2, H5.1
8	1	9.2.1	Defining and understanding the problem	H4.2
		9.2.2	Planning and design of software solutions	
9	1	9.2.3	Implementation of software solution	H4.2
10	1	9.2.1	Defining and understanding the problem	H4.2
		9.2.3	Implementation of software solution	
11	1	9.2.4	Testing and evaluation of software solutions	H5.3
12	1	9.3	Developing a solution package	H4.1, H4.2
13	1	9.2.3	Implementation of software solution	H4.2
14	1	9.2.2	Planning and design of software solutions	H4.2
15	1	9.3	Developing a solution package	H5.1
16	1	9.2.3	Implementation of software solution	H1.1
17	1	9.2.3	Implementation of software solution	H3.2
18	1	9.2.1	Defining and understanding the problem	H6.1
19	1	9.2.3	Implementation of software solution	H6.4
20	1	9.3	Developing a solution package	H6.1
21 (a) (i)	3	9.2.4	Testing and evaluation of software solutions	H4.2, H4.3
21 (a) (ii)	3	9.2.4	Testing and evaluation of software solutions	H4.2, H4.3
21 (a) (iii)	4	9.2.4 9.2.5	Testing and evaluation of software solutions Maintenance of software solutions	H4.2, H4.3
21 (b) (i)	4	9.1 9.2.4 9.2.5	Development and impact of software solutions Testing and evaluation of software solutions Maintenance of software solutions	H3.1, H3.2, H4.1
21 (b) (ii)	6	9.1 9.3	Development and impact of software solutions Developing a solution package	H5.2
22 (a) (i)	3	9.2.3	Implementation of software solution	H1.2, H4.3
22 (a) (ii)	4	9.2.1	Defining and understanding the problem	H1.2, H4.3
22 (a) (iii)	7	9.2.2	Planning and design software solutions	H1.2, H4.3
22 (b)	6	9.3	Developing a software package	H5.2
23 (a) (i)	3	9.2.3	Implementation of software solution	H1.3
23 (a) (ii)	3	9.2.3	Implementation of software solution	H1.3

Question	Marks	Content		Syllabus outcomes
23 (b)	5	9.1.2	Application of software development approaches	H2.2, H3.1, H3.2
		9.2.3	Implementation of software solution	
23 (c) (i)	3	9.2	Software development cycle	H6.2
		9.3	Developing a solution package	
23 (c) (ii)	6	9.2.1	Defining and understanding the problem	H1.2, H4.1
		9.1.2	Application of software development approaches	
		9.3	Developing a solution package	
24 (a)	3	9.4.1	Evolution of programming languages	H1.2
24 (b)	5	9.4.1	Evolution of programming languages	H2.1, H2.2
24 (c) (i)	3	9.4.1	Evolution of programming languages	H4.2
24 (c) (ii)	3	9.4.1	Evolution of programming languages	H4.2
24 (d)	6	9.4.1	Evolution of programming languages	H3.1
25 (a)	3	9.4.2	The software developer's view of the hardware	H1.1, H1.3
25 (b)	5	9.4.2	The software developer's view of the hardware	H1.3, H4.1, H4.3
25 (c)	6	9.4.2	The software developer's view of the hardware	H1.1, H4.1
25 (d)	6	9.4.2	The software developer's view of the hardware	H1.1, H4.2

2001 HSC Software Design and Development Marking Guidelines

Question 21 (a) (i) (3 marks)

Outcomes assessed: H4.2, H4.3

<div style="text-align:center">MARKING GUIDELINES</div>

Criteria	Marks
• Table structure including columns/rows for each variable to be traced and row/columns for description of each location in the algorithm • Includes all of the variables, Train etc AND • Performs a desk check with ALL 3 test data	3
• Includes a table structure AND • Includes any of the variables listed OR • Performs a partial desk check with ONE of the test data given	2
• Indicates a table structure OR • Indicates at least TWO of the variables listed	1

Question 21 (a) (ii) (3 marks)

Outcomes assessed: H4.2, H4.3

MARKING GUIDELINES

Criteria	Marks
• Identifies the two errors and refers to desk check to explain why the algorithm doesn't work	3
• Identifies one error and gives an explanation of how it affects the algorithm OR • Identifies both errors but no description of these errors	2
• Identifies an error but with no description	1

Question 21 (a) (iii) (4 marks)

Outcomes assessed: H4.2, H4.3

MARKING GUIDELINES

Criteria	Marks
• Provides a solution to the problem areas of the loop counter and the initiation of location variables	4
• Provides a solution to 1 of the 2 problem areas and successfully identifies the other area	3
• Identifies both problem areas OR • Provides a solution to one problem area	2
• Identifies one problem area	1

Question 21 (b) (i) (4 marks)

Outcomes assessed: H3.1, H3.2, H4.1

MARKING GUIDELINES

Criteria	Marks
• Identifies and outlines the main features of at least 2 ethical issues involved in this scenario	4
• Identifies 2 ethical issues and one is related and clearly linked to scenario	3
• Identifies 1 ethical issue and relates it to the scenario OR • Identifies 2 ethical issues within 2 broad areas but doesn't relate these to the scenario	2
• Identifies 1 ethical issue	1

Question 21 (b) (ii) (6 marks)

Outcomes assessed: H5.2

MARKING GUIDELINES

Criteria	Marks
• Identifies a number of responsibilities of the software developers relevant to the scenario. Provides several strategies that management could employ. Provides arguments for and/or against their implementation	5–6
• Identifies some responsibilities relevant to the scenario • May not necessarily distinguish between management and programmers roles in providing a strategy/strategies	3–4
• Gives a strategy/strategies without discussion • Identifies a responsibility without discussion or reference to the scenario OR • Responsibility to develop a code of conduct	1–2

Question 22 (a) (i) (3 marks)

Outcomes assessed: H1.2, H4.3

MARKING GUIDELINES

Criteria	Marks
• Layout design conforms to good design principles AND • All necessary fields are included and identified AND • The size of fields is indicated AND • Includes navigation elements	3
• All necessary fields are included and identified AND • Layout design conforms to good design principles OR • All necessary fields are included and identified AND • Size of fields is indicated	2
• Layout design conforms to good design principles OR • All necessary fields are included and identified	1

Question 22 (a) (ii) (4 marks)

Outcomes assessed: H1.2, H4.3

MARKING GUIDELINES

Criteria	Marks
• Draws an IPO chart or diagram that consists of a table format, correctly indicating inputs, processes and outputs AND lists the correct inputs, processes AND outputs	4
• Draws an IPO chart or diagram that consists of a table format AND correctly labels the columns AND fills in correctly at least 2 components each column OR • Identifies correct inputs, processes and outputs BUT fails to label diagram correctly	3
• Draws some other form of diagram AND correctly labels the parts AND lists inputs, processes and outputs OR • Draws a 3 column IPO chart or diagram with correct column labels BUT only one column filled in correctly OR • Correctly identifies a component in each column	2
• Draws some other form of diagram BUT fails to label correctly AND does not list the inputs, processes and outputs OR • Draws a three column IPO chart or diagram (labelled or unlabelled) BUT does not list any inputs, processes or outputs	1

Question 22 (a) (iii) (7 marks)

Outcomes assessed: H1.2, H4.3

MARKING GUIDELINES

Criteria	Marks
Components necessary are: A • indicates a logical structure of an algorithm related to the problem B • priming read • loop through transaction file until 999 found • read at bottom of loop C • access inventory file as a random access file • uses a flag to check for record found • displays an error if corresponding record not found D • prints message when quantity on hand is below 3 within the inventory group	7
Combinations of the above will gain students marks ie A = 1 mark B = 2 marks C = 2 marks D = 2 marks	

Question 22 (b) (6 marks)

Outcomes assessed: H5.2

MARKING GUIDELINES

Criteria	Marks
• Gives any 2 documentation methods with their stages. MUST be from different stages of the cycle AND justifies the use of each type in that stage	6
• Identifies TWO types of documentation with their stages without fully justifying the use in each stage	4–5
• Identifies 2 types of documentation with their stages AND provides justification for the use of ONE OR • Identifies ONE type of documentation with its stage and justifies its use in that stage	2–3
• Identifies TWO types of documentation	1

NOTE:
A student who names TWO pieces of documentation used in the SAME stage OR ONE piece of documentation used in TWO different stages should receive no more than 4 marks.

Question 23 (a) (i) (3 marks)

Outcomes assessed: H1.3

MARKING GUIDELINES

Criteria	Marks
• Identifies a problem and shows its effects	3
• Identifies and gives some indication of its effect	2
• Identifies a problem	1

Question 23 (a) (ii) (3 marks)

Outcomes assessed: H1.3

MARKING GUIDELINES

Criteria	Marks
• Defines an extra item for hexadecimal digits that can be included in the definition of <hexadecimal>**::=**.......|<hexadecimal>	3
• Defines <hexadecimal> with only numerical digit and A to F characters AND • Includes ....|<hexadecimal>	2
• Defines <hexadecimal> with only numerical digit and A to F characters OR • Incorrect definition of <hexadecimal> BUT • Has included ....|<hexadecimal>	1

Question 23 (b) (5 marks)

Outcomes assessed: H2.2, H3.1, H3.2

MARKING GUIDELINES

Criteria	Marks
• Identifies two or more effects and makes a valid assessment of their impacts within the context of the scenario	4–5
• Identifies two or more effects with a description of each with little assessment of the effects on the system	3
• Identifies one or more effects with a description of each with little reference to the scenario	2
• Identifies one or more possible effects	1

Question 23 (c) (i) (3 marks)

Outcomes assessed: H6.2

MARKING GUIDELINES

Criteria	Marks
• Identifies characteristics of any two types of users or representatives of the user group • Links appropriate methods of involvement	3
• Identifies characteristics of at least two different types of users or representatives of the user group • Lists at least two different methods of involvement OR • Identifies an appropriate method of involvement for one identified type of user	2
• Identifies characteristics of at least two different users OR • Lists at least two methods of involvement OR • Identifies one user and a method of involvement which is not necessarily related to the user given	1

Question 23 (c) (ii) (6 marks)

Outcomes assessed: H1.2, H4.1

MARKING GUIDELINES

Criteria	Marks
• Discusses each of the approaches and makes a judgement on the most appropriate for this scenario • Justification for using either RAD or prototyping	5–6
• Identifies characteristics of the problem that will need to be addressed • Describes two or more software development approaches	3–4
• Identifies two or more software development approaches	1–2

Question 24 (a) (3 marks)

Outcomes assessed: H1.2

MARKING GUIDELINES

Criteria	Marks
• Identifies both paradigms correctly and justifies the identification by linking examples from each fragment to the features of each paradigm	3
• Identifies one paradigm correctly and justifies the identification by linking examples from one fragment to the features of that paradigm	2
• Identifies the paradigm in (i) as being logic and/or linking the paradigm in (ii) as functional	1

Question 24 (b) (5 marks)

Outcomes assessed: H2.1, H2.2

MARKING GUIDELINES

Criteria	Marks
• Discusses a number of reasons for characteristics of OOP, that give it advantages over its predecessors in addressing emerging needs	4–5
• Identifies reason(s) with discussion linking some characteristics of OOP with emerging needs	3
• Identifies one reason for emergence of OOP with brief discussion OR • Identifies a number of reasons with no supporting discussion	2
• Describes the object oriented paradigm OR • Identifies a reason with no supporting discussion	1

Question 24 (c) (i) (3 marks)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Correctly identifies the error as a failure to assign an initial value to Inst.Rectangle.height before entering the loop AND describes two different methods of correcting the error	3
• Correctly identifies the error as above AND describes one method of correcting it OR • Describes two different methods of correcting the error without explicitly stating the nature of the error	2
• Identifies the error correctly but does not describe any methods of correcting it OR • Describes one method of correcting the error without explicitly stating the nature of the error	1

Question 24 (c) (ii) (3 marks)

Outcomes assessed: H4.2

MARKING GUIDELINES

Criteria	Marks
• Defines a triangle object correctly with private base and height variables AND a prototype of the area function AND a definition of the area function AND adds an instance definition to the variables section	3
• Defines the triangle as above but fails to declare an instance of a triangle OR • Defines the triangle variables and defines an instance but does not define the function	2
• Defines only triangle private variables OR • Defines only the area function OR • Only declares an instance of a triangle OR • Is able to extrapolate from the example in some way	1

Question 24 (d) (6 marks)

Outcomes assessed: H3.1

MARKING GUIDELINES

Criteria	Marks
• Identifies key features of the system required and supports the selection of the paradigm by relating its features to the system requirements	5–6
• Chooses an appropriate paradigm linking some of its features to the system requirements	3–4
• Identifies some system requirements and/or attempts to link requirements to characteristics of a paradigm	1–2

Question 25 (a) (i) (3 marks)

Outcomes assessed: H1.1, H1.3

MARKING GUIDELINES

Criteria	Marks
• Describes the relationship between ASCII and hexadecimal by identifying their similarities and differences	3
• Describes both ASCII and hexadecimal characteristics	2
• Describes either ASCII or hexadecimal	1

Question 25 (b) (5 marks)

Outcomes assessed: H1.3, H4.1, H4.3

MARKING GUIDELINES

Criteria	Marks
Pseudocode that: • Correctly declares all variables • Reads data from hardware to the header variable • Controls hardware devices using the seek function • Correctly processes data by extracting data from the header variable • Uses a loop structure to read data	5
• Achieves 1 to 4 of the items above	1–4

Question 25 (c) (6 marks)

Outcomes assessed: H1.1, H4.1

MARKING GUIDELINES

Criteria	Marks
• Constructs truth table for section X showing correct structure • Includes 4 permutations of inputs at A & B • Correctly completes truth table to determine outputs of Section X • Identifies section X as XOR gate --- • Constructs a truth table for determining the values of Z and Z^1 (using results from previous truth table) • Identifies states of Z and Z^1 consistent with this truth table	6
• Any 1 to 5 of above points	1–5

Question 25 (d) (6 marks)

Outcomes assessed: H1.1, H4.2

MARKING GUIDELINES

Criteria	Marks

• Labelled circuit with correct inputs for
  A fault detector (0)
  B door closed (1)
  C light switch on (1)
  output (z=1)
 	
• Logic gates within circuit
 	
• Logic gates that correctly produce output =1
 	
• Supported by truth table that matches A, B, and C to inputs
 	
• Completing the truth table consistent with logic gates in drawn circuit.
 	
• Produces a unique solution for Z=1
	6

• Any 1 to 5 of above points	1–5