NSW Education Standards Authority

2021 HSC Software Design and Development Marking Guidelines

Section I

Multiple-choice Answer Key

Question	Answer
1	D
2	B
3	C
4	D
5	D
6	A
7	C
8	A
9	B
10	C
11	D
12	B
13	C
14	A
15	A
16	D
17	B
18	B
19	C
20	D

Section II

Question 21

Criteria	Marks
• Outlines TWO techniques that prevent software piracy	3
• Outlines ONE relevant technique OR • Identifies TWO relevant techniques	2
• Identifies a technique for preventing software piracy	1

Sample answer:

Serial numbers
When a licence is purchased the user is sent a unique serial number which they enter to allow them to use the product.

Installation counter
Each time a user logs on to use an application a counter on the server is incremented. When the counter exceeds the maximum number of allowed users, access to the application is denied.

Question 22

Criteria	Marks
• Explains why live test data is useful in testing this system	3
• Shows some understanding of live testing	2
• Identifies a feature of testing	1

Sample answer:

Live testing gives confidence that the new system will perform as expected when used in the company's environment.

Specifically, it tests using a large number of typical queries, possible keywords and customers, by making use of real queries.

Because live testing also involves volume testing, it will ensure the system can cope with the anticipated number and transaction mix of customer query topics in an appropriate response time.

Question 23 (a)

Criteria	Marks
• Outlines TWO responsibilities of the developers of this software	2
• Identifies ONE responsibility of software developers	1

Sample answer:

Responsibilities of developers include ensuring that copyright is not breached. For example, by not incorporating a photograph of an athlete without the photographer's permission. They must also ensure that each athlete's data is validated and can be stored securely.

Question 23 (b)

Criteria	Marks
• Describes TWO ways in which CASE tools can assist in the development of this software	3
• Outlines ONE way in which CASE tools can assist in the development of this software OR • Identifies TWO ways in which CASE tools can assist	2
• Demonstrates some understanding of CASE tools	1

Sample answer:

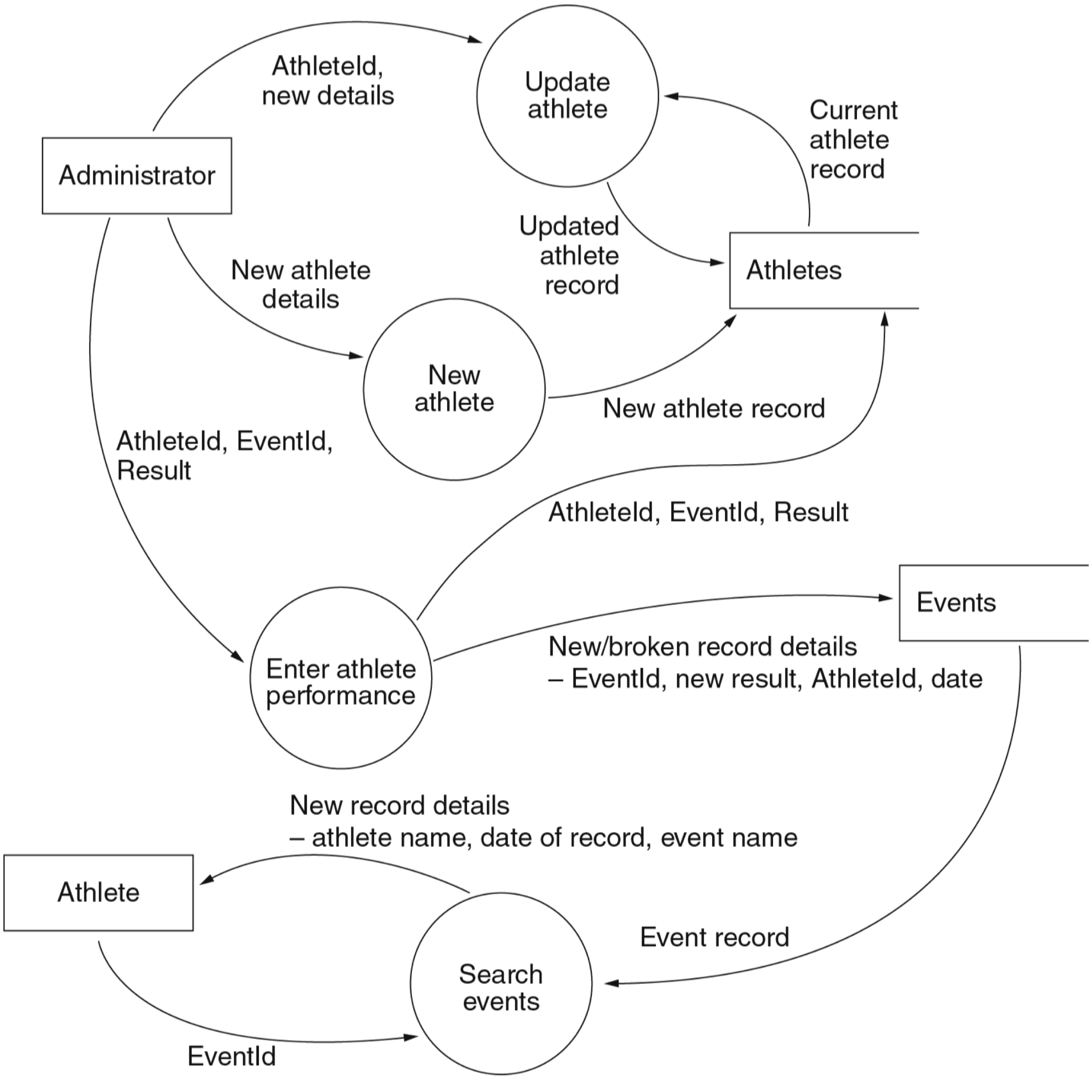
CASE tools automatically create a record of all changes made to code and documentation, thus giving developers easy access to current versions.

A CASE tool can generate relevant test data that tests all pathways and boundaries (such as an athlete performance that is equal to, less than and greater than the stored record result).

Question 23 (c)

Criteria	Marks
• Provides a substantially correct data flow diagram that includes suitable external entities, processes and data stores, with suitable data flow indicated	4
• Provides a relevant data flow diagram that addresses several of the requirements	3
• Provides a diagram with some correctly labelled data flow diagram symbols	2
• Provides a diagram that shows some understanding of the scenario	1

Sample answer:



Question 24

Criteria	Marks
• Outlines TWO ways in which a log book can assist development	3
• Outlines ONE way in which a log book can assist development	2
• Identifies a feature of a log book	1

Sample answer:

A log book details what a programmer is up, allowing others to take over if that programmer leaves the project.

The log book can be used to allow current or future developers to look back on the progress of the development, allowing them to see reasons for various actions that were taken.

Question 25 (a)

Criteria	Marks
• Describes ONE relevant social or ethical issue	2
• Demonstrates a basic understanding of a social or ethical issue	1

Sample answer:

Privacy
As the conversations are transmitted and possibly stored for processing, concerns may arise regarding the security of private conversations.

Question 25 (b)

Criteria	Marks
• Outlines an appropriate installation method relevant to this app	2
• Identifies an installation method	1

Sample answer:

Direct cutover: each individual gets the complete system as soon as they purchase the application.

Question 25 (c)

Criteria	Marks
• Justifies an appropriate software development approach for this app	3
• Describes a software development approach	2
• Demonstrates a basic understanding of software development	1

Sample answer:

Agile
This approach enables the development of this application relatively quickly in a possibly competitive environment. The agile approach means the system can be offered initially with two or three languages and limited functionality, but with updates offered after a short period of time to progressively include more languages and more features in response to feedback.

Question 25 (d)

Criteria	Marks
• Describes ONE performance issue and ONE compatibility issue relevant to this app	4
• Outlines ONE performance issue and ONE compatibility issue relevant to this app	3
• Outlines a performance or compatibility issue OR • Identifies performance and/or compatibility issues	2
• Provides some relevant information	1

Sample answer:

Compatibility: The application must execute consistently between different mobile phones, operating systems and networks. This means that the system must be tested using a wide range of environments.

Performance: Processing times must allow normal conversations to proceed without delays. Some language translations do not lend themselves to word for word conversions, meaning longer processing times are required. The application must load quickly to minimise frustration.

Question 26

Criteria	Marks
• Provides a substantially correct algorithm that includes the following features: – Generates a valid random integer – Enters guess – Checks guess and displays message – Loops correctly – Exits loop appropriately – Deals with 10 incorrect guesses appropriately	4
• Provides an algorithm that addresses most of the features	3
• Provides an algorithm that addresses some features	2
• Shows some understanding of the problem	1

Sample answer:

```
BEGIN GuessMyNumber
        NumGuesses = 0
        Done = FALSE
        Num = RandomNumber
        WHILE Done = FALSE and NumGuesses < 10
            Input YourGuess
            NumGuesses = NumGuesses + 1
            IF YourGuess = Num THEN
                Display "Correct!"
                Done = TRUE
            ELSE
                IF YourGuess > Num THEN
                    Display "Too High!"
                ELSE
                    Display "Too Low!"
                ENDIF
            ENDIF
        ENDWHILE
        IF Done = FALSE  THEN
            Display Num , "Sorry, too many guesses"
        ENDIF
END GuessMyNumber
```

Question 27

Criteria	Marks
• Describes how the data in the files are used	3
• Shows some understanding of how the data in one file are used in this system	2
• Shows some understanding of the system	1

Sample answer:

The customer file contains up-to-date details of each customer. This file is used to authenticate users by retrieving their customer record. The record is updated when a car is hired and returned.

The car file stores details of all cars. It is used to show available cars for hire when a customer wishes to hire a car. This file is updated to set a car as unavailable once hired and then as available once it is returned.

Question 28 (a)

Criteria	Marks
• Provides a substantially correct desk check, indicating a never-ending loop	3
• Provides a partially correct desk check	2
• Shows some understanding of desk checking	1

Sample answer:

Low	High	Found	SearchItem	Index	List(index)	Output
1	7	FALSE	42	4	44	
1	4			2	38	
2	4			3	40	
3	4			3	40	

Does not end

Question 28 (b)

Criteria	Marks
• Correctly modifies both lines	2
• Demonstrates some understanding of the problem	1

Sample answer:

Line 12 high = index –1
Line 14 low = index +1

Question 29

Criteria	Marks
• Identifies the flag and explains how using it helps the efficiency of the algorithm	3
• Identifies the flag and shows some understanding of its effect on the algorithm	2
• Shows some understanding of flags OR the logic of the algorithm	1

Sample answer:

The variable Found is used as the flag. This is set to False until the String is located. The efficiency is improved by allowing the WHILE loop to be exited as soon as String is located, thus not requiring any further searching, reducing processing time.

Question 30

Criteria	Marks
• Provides a substantially correct algorithm that includes the following features: – Looks for a match in the description – Checks the required number of characters – Uses the library routines appropriately	4
• Provides an algorithm that addresses most of the features	3
• Provides an algorithm that attempts to address some features	2
• Shows some understanding of the problem	1

Sample answer:

```
BEGIN IsMatch (string, BigString )
    Located = FALSE
    NumToCheck = length (BigString) – length (string) + 1
    FOR j = 1 to NumToCheck
        Check = extract (BigString, j, length (string))
        IF Check = string THEN
            Located = TRUE
        ENDIF
    NEXT j
    RETURN Located
END IsMatch
```

Question 31

Criteria	Marks
• Provides a substantially correct definition	3
• Provides a partially correct definition	2
• Shows some understanding of the problem	1

Sample answer:

Multiway Selection = SELECT case <option> {<option>} [OTHERWISE : [<statement>] {;<statement>} END SELECT

option = <condition> : <statement>{;<statement>}

Question 32

Criteria	Marks
• Provides a substantially correct description of the process, including the fetch–execute cycle and use of the program counter	4
• Provides a partially correct description of the process	3
• Provides a description of an aspect of the process	2
• Identifies a feature of the fetch–execute cycle	1

Sample answer:

Fetch: The instruction at the address currently in the program counter (SUB R3, R1, R2) is copied from RAM.

Decode: The opcode (the first byte) is used to identify the subtraction action. The total length of the instruction (in bytes) is added to the program counter so it now contains the address of the next instruction to be executed (STORE R2, N3).

Execute: The contents of R3 are placed in the accumulator and the data from R1 are then subtracted from it in the ALU.

Store: The result is stored in R2.

Question 33

Criteria	Marks
• Provides a substantially correct algorithm that includes the following features: – Reads every record in Overtime file – Validates the EmpID for each record – Retrieves the correct Employee record, using EmpID as the key – Updates the OvertimeHours in the Employee record with the amount from the Overtime record – Rewrites the Employee record using EmpID as the key	5
• Provides an algorithm that addresses most of the features	4
• Provides an algorithm that attempts to address some features	3
• Provides an algorithm that shows some understanding of the problem	2
• Shows some understanding of the problem	1

Sample answer:

```
BEGIN OvertimeUpdate
    OPEN Overtime for Input
    OPEN Employees for Relative access
        Read empID, Hours from Overtime
        WHILE not EOF
            IF EmpID is a number AND empID >= 10000 AND empID <= 99999 THEN
                Read recEmployee into Name, OvertimeHours using EmpID
                recEmployee.OvertimeHours = recEmployee.OvertimeHours + Hours
                Write recEmployee from Name, OvertimeHours using EmpID
               ELSE
                Display “Employee ID”  EmpID “is not valid”
               END IF
            Read empID, Hours from Overtime
        ENDWHILE
    CLOSE Employees
    CLOSE Overtime
END OvertimeUpdate
```

Section III

Question 34 (a) (i)

Criteria	Marks
• Provides THREE correct, related examples	3
• Provides TWO correct, related examples OR THREE correct but unrelated examples	2
• Provides ONE correct example	1

Sample answer:

Class	Book
Subclass	Fiction
Attribute	Author

Question 34 (a) (ii)

Criteria	Marks
• Describes how OOP methods can be used to add the necessary changes	3
• Outlines how an OOP method can be used in this system	2
• Shows a basic understanding of OOP methods	1

Sample answer:

OOP methods can be used to update attributes. Here a method can be used to compare the due date attribute from the Book class to the current date. When the current date is greater than the due date a fine can be calculated and a flag set.

A different method can be used to set a flag in the connected instance of the Person class indicating that a notification is to be sent out. It also adds the fine to the balance attribute for that person.

Question 34 (a) (iii)

Criteria	Marks
• Explains how AI can assist in this scenario	3
• Shows some understanding of the relevance of AI to this scenario	2
• Shows a basic understanding of AI	1

Sample answer:

The system stores details of every book borrowed by every borrower. An AI system can track borrowing habits by looking for patterns in this stored data.

For example when a person borrows a certain book, the system stores details of the books that are borrowed at the same time. The AI system can find patterns of borrowing using this collected data and therefore recommend similar books.

Question 34 (b) (i)

Criteria	Marks
• Provides a correct query	2
• Shows some understanding of a relevant query	1

Sample answer:

```
?eats(human, A)
```

Question 34 (b) (ii)

Criteria	Marks
• Provides a correct rule	2
• Provides a partially correct rule	1

Sample answer:

```
self_eater(A) :- eats(A, A)
```

Question 34 (b) (iii)

Criteria	Marks
• Demonstrates an understanding of the difference between forward and backward chaining for this query	3
• Demonstrates some understanding of forward and/or backward chaining	2
• Demonstrates a basic understanding of how a query is evaluated	1

Sample answer:

Forward chaining starts from the first fact and locates all the facts that include human, identifying the last three facts. From there it will evaluate if the facts make the rules herbivore and carnivore true. Which are both true for a human.

Backward chaining assumes the query is true. Then it uses the omnivore rule to check if herbivore and carnivore are both true. They are both true for humans so the assumption was correct.

Question 34 (c)

Criteria	Marks
• Explains a range of factors that influence the choice of paradigm	4
• Outlines relevant factors that influence the choice of paradigm	3
• Outlines a relevant factor OR identifies relevant factors that influence the choice of paradigm	2
• Provides some relevant information	1

Sample answer:

Understanding the nature of the problem will help determine which paradigm is appropriate, eg OOP for game development.

The skills of the available personnel may be best suited to the use of certain paradigms, reducing the learning curve and increasing the speed of code generation.

Response times of the solution might be different when coded in different paradigms, which may influence the choice of paradigm.

Question 35 (a)

Criteria	Marks
• Provides a correct hexadecimal representation	2
• Shows some understanding of ASCII or HEX	1

Sample answer:

Since M is the next letter after L and is upper case, its representation in binary is 0100 1101. This then converts to the hexadecimal number 4D.

Question 35 (b)

Criteria	Marks
• Explains why the result is displayed as a negative number, based on a substantially correct binary multiplication	3
• Shows some understanding of 2's complement AND/OR binary multiplication	2
• Shows some understanding of the problem	1

Sample answer:

```
00100000 ×
00000101
-------------
00100000
10000000
-------------
10100000
```

Since the leading digit is a 1, the number represented in 2's complement is negative.

Because 5 × 32 = 160, and 160 is outside the allowable range for positive numbers in 8 bit 2's complement (127 is the highest positive available) it appears as a negative number.

Question 35 (c)

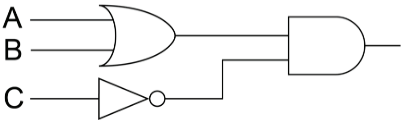
Criteria	Marks
• Provides a correct circuit with no more than four gates, showing working	4
• Provides a substantially correct circuit which shows evidence of simplification	3
• Provides a partially correct circuit OR shows understanding of Boolean logic OR shows understanding of relevant truth tables	2
• Shows some understanding of the problem	1

Sample answer:

$$\overline{A}B\overline{C} + AB\overline{C} + A\overline{B}\overline{C}$$

$$= \overline{C}\left(\overline{A}B + AB + A\overline{B}\right) \quad \text{Factorising}$$

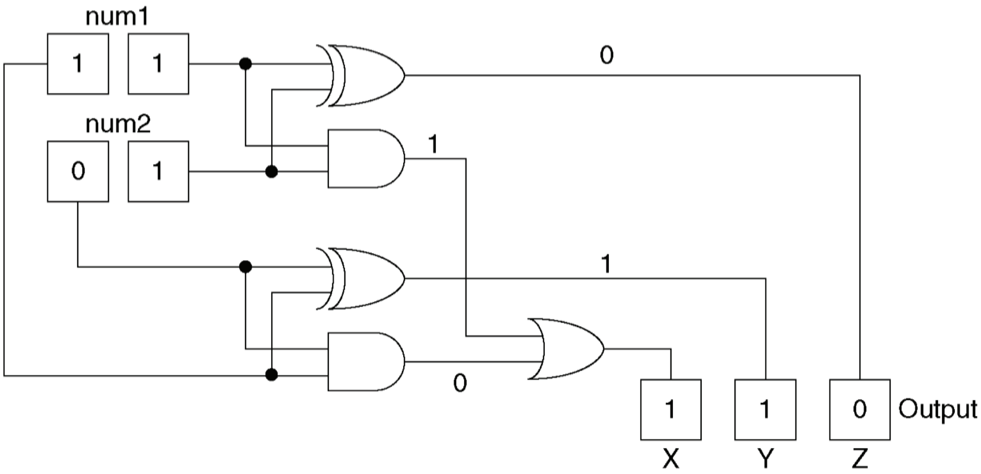
$$= \overline{C}\left(A + B\right) \quad \text{because } \overline{A}B + AB + A\overline{B} = A + B$$



Question 35 (d)

Criteria	Marks
• Provides correct outputs AND identifies a valid reason	3
• Provides correct outputs OR identifies a valid reason	2
• Shows some understanding of the circuit	1

Sample answer:



This produces a sum of 110 instead of 100. The carry from the addition of pairs of bits is not processed correctly.

Answers could include:

There needs to be a third half-adder.

Question 35 (e) (i)

Criteria	Marks
• Outlines the structure of a typical data stream	2
• Identifies a component of a data stream	1

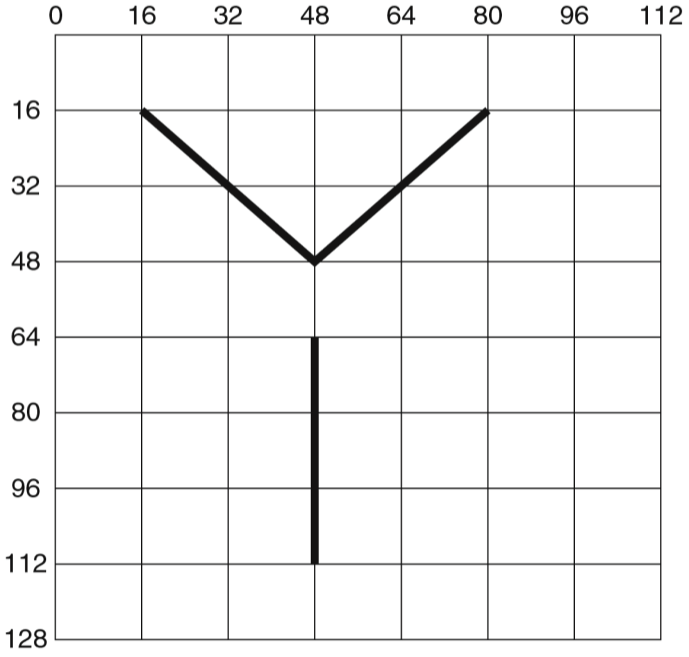
Sample answer:

A data stream will typically contain a header that shows the start of the stream, a data block that contains instructions and a trailer that may contain error checking and the end of the stream.

Question 35 (e) (ii)

Criteria	Marks
• Provides the correct diagram of the printed layer	3
• Provides a substantially correct diagram of the printed layer	2
• Shows some understanding of the problem	1

Sample answer:



Question 35 (e) (iii)

Criteria	Marks
• Provides a substantially correct data block incorporating the shape AND the loop	3
• Provides a partially correct data block incorporating the shape OR the loop	2
• Shows some understanding of the problem	1

Sample answer:

```
0001   1011   0000   1010   0001   0000
11   0000000   0000000
10   0010000   0000000
10   0010000   0010000
10   0000000   0010000
0001   1011   1111   1111
0001   1011   0001   1010
```

2021 HSC Software Design and Development Mapping Grid

Section I

Question	Marks	Content	Syllabus outcomes
1	1	9.2.1 Development cycle stages	H4.1
2	1	9.2.2 Screen elements	H6.4
3	1	9.2.1 Documentation	H5.2
4	1	9.1.1 Licences	H3.1
5	1	9.2.2 Event-driven programming	H4.1
6	1	9.2.2 Sequential files	H1.3
7	1	9.2.2 Control structures	H4.3
8	1	9.2.2 Parameter passing	H4.3
9	1	9.2.3 Types of errors	H4.2
10	1	9.2.1 Documentation	H5.2
11	1	9.2.3 Debugging	H4.2
12	1	9.2.1 Data types	H4.2
13	1	9.2.2 Output from flowchart	H4.2
14	1	9.2.2 Flowchart to pseudocode conversion	H4.3
15	1	9.2.2 Random number generator	H4.3
16	1	9.2.3 EBNF	H4.2
17	1	9.2.3 Output from pseudocode	H4.2
18	1	9.2.3 Lexical analysis	H1.1
19	1	9.2.2 Desk checking	H4.2
20	1	9.2.2 Sorting	H4.2

Section II

Question	Marks	Content	Syllabus outcomes
21	3	9.1.1 Protection against software piracy	H2.2, H3.2
22	3	9.2.4 Methods of testing	H4.2, H5.1
23 (a)	2	9.1.1 Responsibilities of developers	H3.1
23 (b)	3	9.1.2 CASE tools	H5.3
23 (c)	4	9.2.1 Data flow diagram	H5.2
24	3	9.3 Log books	H5.2
25 (a)	2	9.1.1 Social and ethical issues	H3.1
25 (b)	2	9.1.2 Installation method	H1.2
25 (c)	3	9.1.2 Software development approach	H1.2
25 (d)	4	9.2.3 Technical issues	H4.1
26	4	9.3 Design an algorithm	H4.2
27	3	9.2.1 Interpreting system flowchart	H1.3, H5.2

Question	Marks	Content	Syllabus outcomes
28 (a)	3	9.2.2 Desk check	H4.2
28 (b)	2	9.2.2 Locate and fix errors	H5.1
29	3	9.2.3 Algorithm efficiency – flags	H4.2
30	4	9.3 Design an algorithm	H4.2
31	3	9.2.3 EBNF for casewhere	H5.2
32	4	9.2.3 Fetch–execute cycle	H1.1, H1.3
33	5	9.3 Design an algorithm	H4.2

Section III

Question	Marks	Content	Syllabus outcomes
34 (a) (i)	3	9.4.1 OOP classes and attributes	H4.2
34 (a) (ii)	3	9.4.1 OOP methods	H4.2
34 (a) (iii)	3	9.4.1 Artificial intelligence	H2.2
34 (b) (i)	2	9.4.1 LOGIC query	H4.2
34 (b) (ii)	2	9.4.1 LOGIC rule	H4.2
34 (b) (iii)	3	9.4.1 LOGIC chaining	H4.2
34 (c)	4	9.4.1 Choice of paradigm	H2.1, H4.1, H5.3
35 (a)	2	9.4.2 ASCII and HEX	H1.1
35 (b)	3	9.4.2 2’s complement binary calculation	H1.1
35 (c)	4	9.4.2 Boolean logic – circuit design	H1.3
35 (d)	3	9.4.2 Incorrect full adder circuit	H1.3
35 (e) (i)	2	9.4.2 Data stream – structure	H1.3
35 (e) (ii)	3	9.4.2 Interpret a data stream	H2.2
35 (e) (iii)	3	9.4.2 Produce a data stream	H2.2