NSW Education Standards Authority

2022 HSC Software Design and Development Marking Guidelines

Section I

Multiple-choice Answer Key

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

Section II

Question 21

Criteria	Marks
• Describes TWO responsibilities of software developers	3
• Describes ONE responsibility of software developers OR • Outlines TWO responsibilities	2
• Demonstrates some understanding of software developers' responsibilities	1

Sample answer:

Intellectual property – appropriate acknowledgement of intellectual property means that a developer cannot use copied code fragments without first contacting the author, asking their permission and acknowledging the authorship visibly in their product.

Response to user issues – a developer should respond appropriately and promptly, possibly by amending code.

Question 22 (a)

Criteria	Marks
• Explains the need for maintenance in the software development cycle	3
• Describes the need for maintenance in the software development cycle	2
• Provides some relevant information	1

Sample answer:

Maintenance is needed if user requirements change, requiring new functionality in the code. Poor interface design will require relevant modules to be updated. Changes to the data may require extra fields to be defined and used in the code.

Question 22 (b)

Criteria	Marks
• Explains TWO other ways of writing easily maintainable code	3
• Explains ONE other way of writing easily maintainable code OR • Outlines TWO other ways	2
• Demonstrates some understanding of maintenance	1

Sample answer:

By using one logical task in a single module, it is easier to identify the module that needs to be updated.

By inserting blank lines between modules, the source code becomes easier to read and understand.

Question 23

Criteria	Marks
• Provides a thorough comparison between compilers and interpreters from both perspectives	4
• Provides a comparison between compilers and interpreters from either perspective	3
• Demonstrates an understanding of a translation method(s)	2
• Provides some relevant information	1

Sample answer:

A compiler displays all syntax errors in one list to the programmer. Once the syntax errors have been removed, the compiler converts the whole source code to machine language. However, the interpreter deals with one statement at a time, displaying errors and then converting each source code statement to machine language, making it easier to debug.

From the client's perspective, they cannot see the source code of a compiled program, making it harder to edit or plagiarise the code, but the speed of execution is faster than for an interpreted program.

Question 24

Criteria	Marks
• Describes three syntax errors	3
• Describes at least one syntax error	2
• Demonstrates some understanding of EBNF OR • Identifies an error	1

Sample answer:

Any three of:

Line 1. Definition of Assignment specifies | (or), so "age" and 18 cannot both be written.

Line 2. Definition of Iteration specifies "DO", so line 2 should end with DO.

Lines 3,4,5. Definition of Iteration specifies [ ] (optional) around the second statement, so a maximum of 2 statements only is permitted.

Line 9. Definition of Selection specifies "and" prior to any subsequent statements, so this line should begin with "and".

Line 9. Should be Make This Work, F, as an Assignment must start with Make This.

Question 25 (a)

Criteria	Marks
• Recommends a suitable development approach for this study app with appropriate justification	3
• Outlines a development approach for the study app with some justification	2
• Outlines a development approach	1

Sample answer:

An agile approach will allow improved versions of the app to be available to students in a short time frame. Agile allows for quick delivery of updates of the app, based on feedback from students.

Question 25 (b)

Criteria	Marks
• Describes ONE benefit of the pilot method of installation for this app	2
• Shows some understanding of the pilot method of installation	1

Sample answer:

The software developer could select a small group of students to use the study app and test its functionality and identify any problems, minimising the chance of the released product having errors.

Question 25 (c)

Criteria	Marks
• Outlines TWO reasons for system testing of this app	3
• Identifies TWO reasons for system testing OR • Outlines ONE reason	2
• Shows some understanding of system testing	1

Sample answer:

System testing will enable the study app to be tested in a range of operating systems and user hardware likely to be used by students.

It can also be tested more fully by using load testing, with a high transaction rate.

Question 25 (d)

Criteria	Marks
• Provides an explanation of two factors outside the developers' control which can affect a user's experience with this app	3
• Explains ONE factor that can affect a user's experience OR identifies TWO factors	2
• Demonstrates some understanding of a user's experience	1

Sample answer:

The quality of a user's experience could be affected by potential problems with the display on different devices, such as screen size, which could lead to poor legibility or excessive scrolling on small screens.

Incompatibility with other software installed on the user's device could affect the quality of a user's experience, eg the HSC app may crash.

Question 26

Criteria	Marks
• Explains TWO ways in which CASE tools can improve a team's productivity for large projects	4
• Explains ONE way in which CASE tools can improve productivity OR outlines TWO uses of CASE tools	3
• Outlines ONE use of CASE tools OR identifies TWO uses of CASE tools	2
• Provides some relevant information	1

Sample answer:

Version control – if logic errors are located during testing, changes are required in specific modules. As modules are modified, version control is required to prevent two or more developers concurrently making changes to the same module, which would waste time.

Test data generator – a CASE tool can randomly generate a large volume of relevant test data, saving time for the developer and ensuring thorough testing.

Question 27 (a)

Criteria	Marks
• Creates a substantially correct IPO diagram, with details for all processes (authentication, purchase, EFT)	4
• Creates a partially correct IPO diagram, with details for most processes	3
• Demonstrates reasonable understanding of an IPO diagram with at least one process addressed	2
• Demonstrates some understanding of an IPO diagram	1

Sample answer:

Input	*Process*	*Output*
CustomerId Password Customer File	Use CustomerId to retrieve Customer Record Check entered password matches stored password	"No such Customer" "Incorrect Password"

ItemNumber
Quantity
Item File
	For each item
	   Use ItemNumber to retrieve Item record from Item File
	   Cost = Item.Cost * Quantity
	   TotalCost = TotalCost + Cost
	IF TotalCost + Customer.PurchasesToDate > 100 THEN TotalCost = TotalCost $\times$ 0.9
	Add TotalCost to Customer.PurchasesToDate
	Rewrite Customer Record
		Display Item.Name, Item.Cost
		
		Updated customer file
		
		Display TotalCost

TotalCost	Transfer TotalCost from customer account to restaurant account	Updated accounts

Question 27 (b)

Criteria	Marks
• Provides an appropriate data structure, with fields, data types and sizes required to complete the specified processing	3
• Provides a substantially correct data structure	2
• Demonstrates some understanding of an appropriate data structure	1

Sample answer:

A record to be stored in the Customer File with at least the following fields:

Customer Id	Text	6 characters
Password	Text	20 characters
Total Purchases to date	Real	4 bytes
BSB	Text	6 characters
AccountNum	Text	9 characters

Question 27 (c)

Criteria	Marks
• Describes the project manager's use of a Gantt chart throughout the development of this system, with reference to time and skills	3
• Shows some understanding of how a project manager can use a Gantt chart	2
• Provides some relevant information	1

Sample answer:

Using a Gantt chart, the project manager needs to identify each required task (such as authentication, customer input of food choice), allocate a time frame for each and identify the order and linkages / dependencies between the tasks. The project manager allocates team members to tasks that meet their skills, while not allocating overlapping tasks to the same team member.

The project manager can use the Gantt charts to monitor progress, extending individual task time frames and reallocating team members as required.

Question 28 (a)

Criteria	Marks
• Completes a substantially correct desk check	3
• Completes a partially correct desk check	2
• Shows some understanding of desk checking this algorithm	1

Sample answer:

phrase	Len	NewPhrase	done	output
dddg				
dddgX	5	""	F	
		d		
	4			
ddgX	3			
dgX			T	
	2			
gX			F	
		dg		
			T	
	1			
X			F	
		dgX		
				Error message?

At this point the algorithm is trying to compare X with a character that doesn't exist.

Question 28 (b)

Criteria	Marks
• Correctly identifies two errors and changes the code	3
• Identifies two errors or corrects one error	2
• Show some understanding of the algorithm	1

Sample answer:

Line 16 becomes UNTIL Len < = 1.
Swap lines 17 and 18.

Question 29

Criteria	Marks
• Explains why programmers need to use different data types, and provides an example	3
• Provides a basic explanation of different data types	2
• Demonstrates some understanding of data types	1

Sample answer:

Different data types are stored and used in different ways. By defining a variable as a particular data type, the programmer can make use of the data in an appropriate way.

For example, text data types can have characters appended to them or extracted from them, and are stored in ASCII format with one byte per character, but cannot be used in arithmetic calculations.

Question 30

Criteria	Marks

• Provides a substantially correct algorithm that addresses the following features:
  – Initialises highest score
  – Determines highest score by looping through the array
  – Displays the highest score
  – Loops through the whole array, printing names of all players with the highest score
  – Repeats the above for the second highest score
	5

• Provides a substantially correct algorithm that addresses most of the features	4
• Provides an algorithm that attempts to address some of the features	3
• Provides an algorithm that attempts to address a feature	2
• Shows some understanding of the problem	1

Sample answer:

```
BEGIN
    first = 0
    second = 0
    FOR i = 1 to 200
        IF Results.Score(i) >= first THEN
            first = Results.Score(i)
        ELSE
            IF Results.Score(i) > second THEN
                second = Results.Score(i)
            ENDIF
        ENDIF
    NEXT
    print "Highest score was " first
    FOR i = 1 to 200
        IF Results.Score(i) = first THEN
            print = Results.Name(i)
        ENDIF
    NEXT
    print "Second highest score was " second
    FOR i = 1 to 200
        IF Results.Score(i) = second THEN
            print = Results.Name(i)
        ENDIF
    NEXT
END
```

Question 31 (a)

Criteria	Marks
• Contrasts the purposes of the program counter and the accumulator	2
• Provides relevant information about the purpose of the program counter or the accumulator	1

Sample answer:

A program counter holds the current address of the next instruction to be decoded and executed whereas the accumulator is used to hold the results of intermediate calculations.

Question 31 (b)

Criteria	Marks
• Completes a substantially correct desk check	3
• Completes a partially correct desk check	2
• Demonstrates some understanding of the required desk check	1

Sample answer:

Program counter	a	b	c	d	Accumulator	Output
2000	6	4	1	1	6	
2004	6	4	1	1	2	
2008	6	4	1	1	2	
2012	2	4	1	1	2	
2016	2	4	1	1	1	
2020	2	4	1	1	2	
2024	2	4	1	2	2	
2028	2	4	1	2	2	
2000	2	4	1	2	2	
2004	2	4	1	2	–2	
2008	2	4	1	2	–2	
2032	2	4	1	2	2	
2036	2	4	1	2	1	
2040	2	4	1	2	1	1
2044						

Section III

Question 32 (a)

Criteria	Marks
• Describes abstraction in object oriented programming	2
• Shows some understanding of abstraction	1

Sample answer:

In OOP, abstraction enables a programmer to make use of an existing object without having to know the detail of all of the attributes or methods of that object.

Question 32 (b) (i)

Criteria	Marks
• Explains the importance of inheritance in OOP, with reference to the code	3
• Outlines the importance of inheritance in OOP	2
• Provides some relevant information	1

Sample answer:

Inheritance enables attributes of a superclass to be automatically assumed by its subclasses, thus saving the programmer from having to redefine them. In this case, the WATERBIRD class will automatically include all of the attributes of the BIRD class.

Question 32 (b) (ii)

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

Sample answer:

```
subclass SEABIRD is a WATERBIRD {
        public -
                Ocean: String
                MaxDistance: Integer
                MaxAltitude: Integer

                HowHigh(CommonName)
                        return MaxAltitude
                end HowHigh
}
```

Question 32 (c)

Criteria	Marks
• Describes how the software uses pattern matching to compare the photograph to stored known images	3
• Provides some description of pattern matching or AI OR • Shows how software can be used to solve this problem	2
• Shows some understanding of the problem	1

Sample answer:

This system compares the image of the photograph taken by the camera to a range of stored photographs of phones being used while driving. The system uses pattern matching to determine the probability that the features are similar enough to decide that a use of a phone has been detected.

Question 32 (d) (i)

Criteria	Marks
• Explains the result of the query	2
• Shows some understanding of the query	1

Sample answer:

No. Because there is no fact or rule that matches the query.

Question 32 (d) (ii)

Criteria	Marks
• Provides a correct query	3
• Provides a partially correct query	2
• Demonstrates some understanding of rules or queries in the logic paradigm	1

Sample answer:

? is_bigger(donkey, X), is_bigger(X, cat)

Question 32 (e)

Criteria	Marks
• Explains how appropriate programming paradigms can improve productivity, using examples	4
• Describes how a programming paradigm can improve productivity OR • Provides an example of a programming paradigm and its use	3
• Identifies some characteristics of programming paradigms	2
• Shows some understanding of programming paradigms	1

Sample answer:

Where a problem involves large amounts of data that are used to make decisions, eg in expert systems, the logic paradigm is applicable. The data is treated as facts to which rules can be applied. This saves the programmer from hard coding the detailed logic.

For a situation that involves categories of objects that share characteristics, eg for many games, an OOP paradigm is applicable where classes and subclasses can be established, saving the programmer from having to redefine the characteristics.

Answers could include:

- learning curves
- use of reusable modules
- speed of code generation
- approach to testing

Question 33 (a)

Criteria	Marks
• Shows how TWO different integers can be represented	2
• Shows some understanding of integer representation	1

Sample answer:

Using Sign & Modulus: leading 1 => negative
0000011 = 3   Hence = −3

Using 2's complement:
Leading 1 => negative
2's Complement of 10000011 is   01111101 = 125   Hence = −125

Question 33 (b)

Criteria	Marks
• Provides a correct process for conversion into floating point representation	3
• Provides a partially correct conversion into floating point representation	2
• Shows some understanding of floating point representation	1

Sample answer:

$-\dfrac{1}{32} = -0.00001$ (given)

$= -1.0 \times 2^{-5}$ normalising

$\Rightarrow -1.0 \times 2^{122}$ adding 127 to exponent

$= -1.0 \times 2^{01111010}$ converting exponent of 122 to binary

Sign	Exponent	Mantissa
1	0111 1010	00000000000000000000000

Question 33 (c)

Criteria	Marks
• Provides a correct Boolean representation	2
• Attempts to write a Boolean representation	1

Sample answer:

$A.B + A.\bar{B} + A.\bar{C}$

Question 33 (d)

Criteria	Marks
• Provides a correct logic circuit	3
• Provides a partially correct logic circuit OR • Provides a correct truth table OR • Provides a correct Boolean representation	2
• Demonstrates some understanding of the problem	1
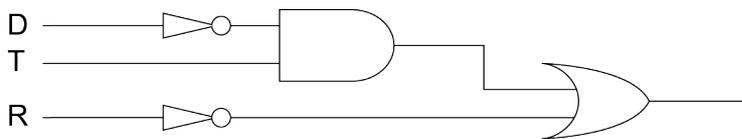
Sample answer:

Question 33 (e) (i)

Criteria	Marks
• Describes what the data stream achieves	3
• Provides a partial description of what the data stream achieves	2
• Shows some understanding of the data stream	1

Sample answer:

Move the cleaner to coordinates (100, 40).

Instruct the cleaner to change direction 90 degrees anticlockwise and move 1 square.

Move 1 square.

Question 33 (e) (ii)

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

Sample answer:

42 in binary = 0101010
18 in binary = 0010010
0 1 0 1 0 1 0 0 0 1 0 0 1 0 0 1

Question 33 (e) (iii)

Criteria	Marks

• Provides a substantially correct algorithm that has the following features:
    – Get initial stream from cleaner, extract bit to check if low battery
    – Send message to cleaner to come home
    – Keep checking stream from cleaner to check if object sensed
    – Once object sensed, move past object
    – Check current position
    – Loop until the controller is reached
	4

• Provides a partially correct algorithm that addresses some of the features	3
• Provides an algorithm that attempts to address some of the features	2
• Shows some understanding of the problem	1

Sample answer:

BEGIN CheckBattery
    Get Stream from cleaner
    BatteryStatus = Extract (Stream, 16, 1)
    IF BatteryStatus = 1 THEN
        REPEAT
            Send "1000010100000000" to cleaner
            Get Stream from cleaner
            SensorStatus = Extract (Stream, 15, 1)
            WHILE SensorStatus = 0
                Get Stream from cleaner
                SensorStatus = Extract (Stream, 15, 1)
            ENDWHILE
            IF SensorStatus = 1 THEN
                Send "0100000000000000" to cleaner
            ENDIF
            Get Stream from cleaner
            X = Extract (Stream, 1, 7)
            Y = Extract (Stream, 8, 7)
        UNTIL X = 5 AND Y = 0
    ENDIF
END CheckBattery

2022 HSC Software Design and Development Mapping Grid

Section I

Question	Marks	Content		Syllabus outcomes
1	1	9.2.2	Control structures	H4.2
2	1	9.2.1	Interface design	H6.4
3	1	9.1.2	Software development approaches	H1.2
4	1	9.2.1	System modelling tools	H5.2
5	1	9.1.1	Social issues (inclusivity)	H3.1
6	1	9.1.1	Collaboration	H6.3
7	1	9.2.2	Data structure / file types	H1.3
8	1	9.2.2	Output from an algorithm	H4.2
9	1	9.2.3	Documentation / syntax diagram	H5.2
10	1	9.2.2	Global variables	H1.3
11	1	9.2.2	Benchmark report	H1.1
12	1	9.2.1	Structure chart	H5.2
13	1	9.2.3	Error types	H4.3
14	1	9.1.2	Client–developer communication	H5.1
15	1	9.2.3	Event-driven vs sequential	H1.2
16	1	9.2.2	Sorting	H4.2
17	1	9.2.2	Searching arrays	H4.2
18	1	9.2.2	Data validation routine	H4.2
19	1	9.2.2	Effect of algorithm on array	H4.2
20	1	9.2.2	Matching algorithms to stated requirements	H5.2

Section II

Question	Marks	Content		Syllabus outcomes
21	3	9.1.1	Developer responsibilities	H3.1
22 (a)	3	9.2.5	Maintenance (reasons)	H4.2
22 (b)	3	9.2.3	Maintainable code	H4.2
23	4	9.2.3	Translation / client and developer perspectives	H1.2
24	3	9.2.3	Syntax (EBNF)	H4.2
25 (a)	3	9.1.2	Software development approaches	H1.2
25 (b)	2	9.1.2	Installation	H1.2
25 (c)	3	9.2.4	Systems testing	H4.3
25 (d)	3	9.1.1, 9.2.3	Impact of hardware on solution	H2.2
26	4	9.1.2	CASE tools / productivity	H5.3
27 (a)	4	9.2.1	Documentation IPO diagram	H5.2
27 (b)	3	9.2.1	Data structure	H5.2
27 (c)	3	9.3	Project management – Gantt Chart	H5.1

Question	Marks	Content	Syllabus outcomes
28 (a)	3	9.2.1 Algorithm desk check	H4.2
28 (b)	3	9.2.2 Error detection	H4.3
29	3	9.2.2 Data types	H1.3
30	5	9.2.2 Algorithm design	H4.2
31 (a)	2	9.2.3 CPU registers (accumulator, program counter)	H1.1
31 (b)	3	9.2.3 Assembler desk check	H1.1

Section III

Question	Marks	Content	Syllabus outcomes
32 (a)	2	9.4.1 OOP abstraction	H1.2
32 (b) (i)	3	9.4.1 OOP inheritance	H1.2
32 (b) (ii)	3	9.4.1 OOP produce code	H4.2
32 (c)	3	9.4.1 LOGIC Artificial Intelligence / pattern matching	H2.2
32 (d) (i)	2	9.4.1 LOGIC analyse a query	H4.2
32 (d) (ii)	3	9.4.1 LOGIC code a query	H4.2
32 (e)	4	9.4.1 Paradigms / productivity	H1.2
33 (a)	2	9.4.2 Data representation (negative integers)	H1.3
33 (b)	3	9.4.2 Data representation (IEEE754)	H1.3
33 (c)	2	9.4.2 Boolean representation of circuit	H5.2
33 (d)	3	9.4.2 Design a logic circuit	H5.2
33 (e) (i)	3	9.4.2 Data stream – interpret	H1.1, H1.3
33 (e) (ii)	3	9.4.2 Data stream – produce	H1.1, H1.3
33 (e) (iii)	4	9.4.2 Data stream – algorithm	H1.1, H1.3