NSW EDUCATION STANDARDS AUTHORITY

2017 HSC Software Design and Development Marking Guidelines

Section I

Multiple-choice Answer Key

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

Section II

Question 21

Criteria	Marks
• Outlines features of the information required	3
• Outlines a feature of the information required OR identifies information	2
• Identifies one piece of information	1

Sample answer

A testing report will include a listing of test data used. This will show the actual data values as well as associated expected outputs and purpose of each data set. The report will also include a statement of where requirements may not have been fully met or where changes could be made.

Question 22

Criteria	Marks
• Provides similarities and/or differences between storyboards and screen designs	3
• Identifies features of storyboards and/or screen designs	2
• Identifies a feature of storyboards or screen designs	1

Sample answer:

Both provide information about the user interface.

Storyboards provide information about changes to screens and navigation between screens, while screen designs provide more detailed information about screen layouts and elements.

Question 23 (a)

Criteria	Marks
• Provides an explanation relevant to the scenario	3
• Identifies benefits of the phased method OR • Outlines a benefit of the phased method	2
• Shows some understanding of an installation method	1

Sample answer

The phased installation method could allow users to become familiar with the interface of, for example, the sales module before using other modules. The retail store does not need to purchase all modules at once, as they are separate.

Question 23 (b)

Criteria	Marks
• Describes how a relevant system modelling tool would be used, with reference to the modules in the scenario	3
• Describes how a relevant system modelling tool is used	2
• Identifies a relevant system modelling tool	1

Sample answer

A structure diagram can be used to show how modules, such as sales and stock, can be accessed and how they share data.

Question 23 (c)

Criteria	Marks
• Provides descriptions of appropriate techniques relevant to the context	3
• Provides a description of an appropriate technique OR identifies two relevant techniques	2
• Identifies a relevant technique or tool	1

Sample answer

The team leader could use project management tools that include a Gantt chart to determine and monitor the sequence of completion of modules.

The leader should appropriately allocate module completion tasks to team members.

Question 24

Criteria	Marks
• Describes how the use of CASE tools benefits a developer	3
• Identifies ways OR outlines one way in which the use of CASE tools benefits a developer	2
• Identifies one way in which the use of CASE tools benefits a developer	1

Sample answer:

CASE tools can save the developer time and effort. For example, drawing software can be used to produce modelling tools such as data flow diagrams.

Code generators can automatically produce code that is free of syntax errors.

CASE tools can be used to rapidly create large amounts of test data.

Version control software helps ensure all team members are working with the same version of specifications, modules, data etc.

Question 25

Criteria	Marks
• Provides outlines of two factors to be considered by a developer	3
• Provides an outline of one factor OR identifies two factors	2
• Shows some understanding of programming languages	1

Sample answer:

A developer needs to consider if the execution of software will be predetermined or depend on user actions, influencing the choice between event-driven or sequential languages.

The developer needs to consider their own level of expertise in relevant languages.

Question 26

Criteria	Marks
• Provides an explanation of relevant features of the algorithm	3
• Shows understanding of maintenance of the algorithm	2
• Identifies one feature that facilitates maintenance	1

Sample answer:

Blank lines separate logical sections of the algorithm, such as input, calculations and output, making them easier to locate.

Values that may change, such as NormalHours, are stored in a variable and initialised near the start of the algorithm so they are easy to find and need only be altered once, minimising the risk that some changes are overlooked.

Question 27

Criteria	Marks
• Provides an explanation of the effects of cloud computing on developers and users	3
• Provides a description of an effect of cloud computing on developers or users	2
• Shows some understanding of cloud computing	1

Sample answer:

Developers can develop applications that can be stored in the cloud and be accessible to users from multiple locations.

Users can access data from multiple devices. They generally do not know where it is stored. Storage is virtually unlimited for both developers and users.

Question 28 (a)

Criteria	Marks
• Provides a substantially correct desk check	3
• Provides a partially correct desk check	2
• Shows some understanding of desk checks	1

Sample answer:

Low	*High*	*Found*	*Middle*	*Display*
1	8	False	4	
4	6		6	
	5		5	
4			4	
4			4	

Infinite loop

Question 28 (b)

Criteria	Marks
• Provides changes to address both errors	2
• Provides a change to address one error	1

Sample answer:

Line 8 should be Set High to Middle – 1
Line 13 should be Set Low to Middle + 1

Question 28 (c)

Criteria	Marks
• Provides a reason why a binary search is not suitable AND how a linear search can be used	3
• Provides a reason why a binary search is not suitable OR how/why a linear search can be used	2
• Shows some understanding of linear or binary searches	1

Sample answer:

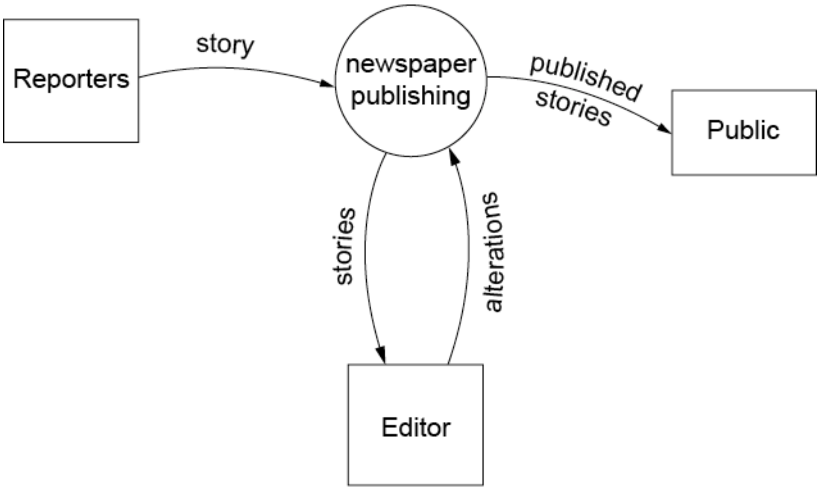
The array can be searched using a linear search method that steps through the whole array, reporting every time 42 is encountered.

Even though the array is sorted, a binary search would not be suitable as it would search for 42 in either the left or right half of the array, and thus not find all the 42s.

Question 29

Criteria	Marks
• Provides a correct context diagram	3
• Provides a substantially correct context diagram	2
• Shows some understanding of relevant system documentation	1

Sample answer:



Question 30

Criteria	Marks
• Provides a detailed description of the fetch–execute cycle	3
• Provides features of the fetch–execute cycle	2
• Shows some understanding of the fetch–execute cycle	1

Sample answer:

A copy of the next instruction in RAM is placed in the instruction register.
The program counter is incremented by the number of bytes in the instruction.

The instruction is decoded and carried out. Results from calculations are stored.

Question 31

Criteria	Marks
• Provides a correct ROBO program	3
• Provides a ROBO program with some correct statements	2
• Shows some understanding of the problem	1

Sample answer:

```
BEGIN
    REP 5 # FD 100; LT 90; FD 50; BK 100; RT 90; #;
END
```

Question 32

Criteria	Marks
• Provides a substantially correct algorithm that incorporates the following features – A loop with an appropriate terminating condition – A selection structure to determine divisibility – Calculation of tallies – Appropriate output	4
• Provides a partially correct algorithm that addresses the main features of the problem	3
• Provides an algorithm that attempts to address some features of the problem	2
• Shows some understanding of the problem	1

Sample answer:

```
BEGIN
    Tneither = 0
    T2 = 0
    T3 = 0
    Tboth = 0
    Input N
    WHILE N>=0
        CASEWHERE
            Divisible(N,3) AND Divisible (N,2): add 1 to Tboth
            Divisible(N,3): add 1 to T3
            Divisible(N,2): add 1 to T2
            OTHERWISE: add 1 to Tneither
        ENDCASE
        Input N
    ENDWHILE
    Output Tneither “were divisible by neither”
    Output T2 “were divisible by 2 but not 3”
    Output T3 “were divisible by 3 but not 2”
    Output Tboth “were divisible by both”
END
```

Question 33 (a)

Criteria	Marks
• Outlines relevant responsibilities of the developer	3
• Identifies relevant responsibilities OR outlines a relevant responsibility of the developer	2
• Identifies a relevant responsibility of the developer	1

Sample answer:

Data about purchases and credit card details must be secured by encryption of passwords and suitable firewalls.

The software must meet the clothing store's requirements, be free of malware and be compatible with the store's existing systems.

Question 33 (b) (i)

Criteria	Marks
• Provides a correct IPO chart including linking between inputs, processes and outputs, and some detail of the processing involved	4
• Provides a substantially correct IPO chart	3
• Provides an IPO chart containing a variety of relevant items	2
• Shows some understanding of the problem	1

Sample answer:

Input	*Process*	*Output*
Item, quantity	Linear search on StockList() to find the itemname index	Index
Item, quantity	Compare quantity input with quantity in StockList() If quantity input > number in stock	Message to user ‘Not enough in stock’
Item, quantity	If quantity input <= number in StockList() Update numinstock in StockList() Add entry to end of CartList() array Update total price (quantity ordered * price per item)	Updated CartList() Updated StockList()
Continue/checkout	If continue, allow customer to select more items	Total Price

Question 33 (b) (ii)

Criteria	Marks
• Provides a substantially correct algorithm that correctly accesses both arrays and incorporates the following features: – a loop to allow for multiple purchases – determination of the index of the StockList() array for the item – an appropriate comparison to check whether there are enough items in stock – a new record in the CartList array – updated NumInStock field – calculation and display of the total price	5
• Provides an algorithm which correctly accesses array elements and addresses most of the main features of the algorithm	4
• Provides an algorithm which attempts to access an array and addresses some of the main features of the algorithm	3
• Provides an algorithm with an attempt to address an aspect of the problem	2
• Provides an algorithm that shows some understanding of the problem	1

Sample answer:

```
BEGIN Purchase
    CartIndex = 1
    Total = 0
    Get Finished
    WHILE Finished Is False
        Get Item
        Get NumberR
        FOR StockIndex = 1 to Maximum StockIndex
            IF Item = StockList(StockIndex). ItemName THEN
                IF NumberR > StockList(StockIndex).NumInStock THEN
                    Display “Insufficient stock”
                ELSE
                    CartList(CartIndex).ItemName = Item
                    CartList(CartIndex).NoRequired = NumberR
                    Increment CartIndex
                    StockList(StockIndex).NumInstock = StockList(StockIndex).NumInstock – NumberR
                    Total = Total + Stocklist(StockIndex).PricePerItem  x  NumberR
                ENDIF
            ENDIF
        NEXT StockIndex
        Get Finished
    ENDWHILE
    Display Total
END
```

Section III

Question 34 (a)

Criteria	Marks
• Identifies the essential features of polymorphism	2
• Shows some understanding of polymorphism	1

Sample answer:

Polymorphism is the ability of a method to process objects differently depending on the class of the object, its data type or number of parameters.

Question 34 (b) (i)

Criteria	Marks
• Shows a good understanding of the differences between the types of declaration, with examples from the code	3
• Shows an understanding of the differences between the types of declaration	2
• Shows some understanding of private or public declarations OR • Identifies an attribute with a public declaration and an attribute with a private declaration	1

Sample answer:

For example, the itemNumber attribute in the Pet class is private and therefore cannot be altered by any method within the Accessory class, whereas the getPrice method is public and therefore can be used from anywhere in the program.

Question 34 (b) (ii)

Criteria	Marks
• Provides an answer that demonstrates inheritance and the assignment of appropriate values to attributes	2
• Provides an answer that assigns values to attributes	1

Sample answer:

```
d.colour = “black”
d.male = TRUE
d.microchipped = TRUE
d.supplier = “puppy farm”
```

Question 34 (b) (iii)

Criteria	Marks
• Extends the code to include the attribute and a correct method	4
• Extends the code to include the attribute and the method	3
• Extends the code to include the attribute or the method	2
• Shows an understanding of what is required for this scenario	1

Sample answer:

```
class Toy {
        is a Accessory

        private –

        public –
            Toy()
            petType: string
            discount()
                IF numberOfItems > 3 THEN
                    price = price * 0.9
                ENDIF
            END discount
    }
```

Question 34 (c)

Criteria	Marks
• Explains how the paradigms can be used to develop most parts of the software	4
• Explains how the paradigms can be used to develop some parts of the software	3
• Shows an understanding of one of the paradigms with reference to the scenario OR • Shows some understanding of both paradigms	2
• Shows some understanding of a paradigm	1

Sample answer:

The management of the database could be carried out in OO. Clients would be objects within a class with attributes containing their details such as name and address and methods such as bookTour().

The logic paradigm would be suitable for matching clients to products as facts would be used to store a client's special interests and wishlist items, while an inference engine would take care of the actual matching process.

Question 34 (d) (i)

Criteria	Marks
• Shows a good understanding of backward chaining with reference to this scenario	3
• Shows a good understanding of chaining	2
• Shows some understanding of the problem	1

Sample answer:

The program will break down the rule first_cousin(X, Y), then break down the rules grandparent(X, Y), and \+sibling(X, Y).

The program would find facts to determine that Alice and Gemma have the same grandparent (Geoffrey) and they are not the same person. As there are no facts to show that they are siblings, it will evaluate the query as true.

Question 34 (d) (ii)

Criteria	Marks
• Provides an answer that establishes the required rule	2
• Shows an understanding of logic paradigm syntax	1

Sample answer:

```
greatgrandparent(X, Y) :- parent(Z, Y), grandparent(X, Z)

second_cousin(X, Y) :- greatgrandparent(Z, X), greatgrandparent(Z, Y), X ≠ Y, \+sibling(X, Y),
\+first_cousin(X, Y)
```

Question 35 (a)

Criteria	Marks
• Explains how subtraction works AND why binary representation of a negative number is required	3
• Explains how subtraction works OR why binary representation of a negative number is required	2
• Shows some understanding of the problem	1

Sample answer:

Computers can only add. The number to be subtracted is converted to a negative number and added.

2's complement has only one way of expressing zero, has a greater range than sign-modulus or 1's complement and produces the correct answer.

Question 35 (b)

Criteria	Marks
• Provides a comparison with reference to both the exponent and mantissa	3
• Provides a comparison with reference to either the exponent or the mantissa OR correctly provides the differences	2
• Shows limited understanding of floating point numbers	1

Sample answer:

MySystem has a greater range of values (bigger and smaller) because it has a bigger exponent but IEEE-754 has greater precision (more decimal places) because it has a bigger mantissa.

Question 35 (c) (i)

Criteria	Marks
• Provides a substantially correct truth table	3
• Provides a partially correct truth table	2
• Shows some understanding of truth tables	1

Sample answer:

Carry in	*A*	*B*	*X*	*Y*	*Z*	*Sum*	*Carry*
0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	0
0	1	0	1	0	0	1	0
0	1	1	0	0	1	0	1
1	0	0	0	0	0	1	0
1	0	1	1	1	0	0	1
1	1	0	1	1	0	0	1
1	1	1	0	0	1	1	1

Question 35 (c) (ii)

Criteria	Marks
• Explains the purpose of the OR gate in this context	2
• Shows understanding of adders or half adders	1

Sample answer:

The OR gate allows for a carry of 1 from either of the two half adders in the circuit. Also, when 3 bits are added, the result should only contain two bits.

Question 35 (d)

Criteria	Marks
• Relates the features of the circuit to its purpose	3
• Shows some understanding of the features of a flip-flop circuit	2
• Identifies a purpose	1

Sample answer:

A flip-flop circuit is used in the storage of a bit in memory.

Having the output from the first NAND gate feed into the second NAND gate, and the output from the second into the first, allows the circuit to maintain its state until deliberately reset.

Question 35 (e)

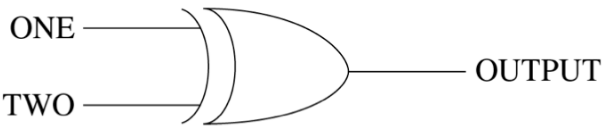
Criteria	Marks
• Provides a correct circuit design	2
• Shows some understanding of the problem	1

Sample answer:

Truth table (optional):

Input ONE	*Input TWO*	*Output*
0	0	0
0	1	1
1	0	1
1	1	0

Circuit:



Question 35 (f)

Criteria	Marks
• Provides similarities and demonstrates a sound understanding of the reasons for differences between the data streams	4
• Provides some explanation for a limited number of similarities or differences	3
• Lists similarities AND differences	2
• Shows some understanding of data streams	1

Sample answer:

Similarities:
The need for start and stop bits to discriminate between data and idle bits.
Both may require identification of the source and destination.

Differences:
The data stream coming from the drone will be much larger as it contains the data for images, while commands sent to the drone require only a few bits.

Error checking for the images is not essential as they are sent frequently and minor errors in the image will not matter, while the commands sent to the drone must be correct to ensure the drone responds correctly or it may crash.

2017 HSC Software Design and Development Mapping Grid

Section I

Question	Marks	Content	Syllabus outcomes
1	1	9.1.2 Approaches	H1.2
2	1	9.2.3 Control structures	H4.2
3	1	9.2.3 User documentation	H5.2
4	1	9.2.3 Syntax diagrams	H2.1
5	1	9.1.1 Licences	H3.1
6	1	9.2 Stages of development	H1.2, H4.2
7	1	9.2.3 Error types	H4.2
8	1	9.2.1 Defining the problem	H3.2, H4.2
9	1	9.2.1 Data types	H4.2
10	1	9.2.1 Quality criteria	H3.2, H5.1
11	1	9.2.2 Sentinel values	H4.2
12	1	9.2.2 Event driven vs sequential	H1.2
13	1	9.2.3 Translation	H1.2
14	1	9.2.2 Search methods	H4.2
15	1	9.2.2 Sequential / relative files	H1.2, H4.2
16	1	9.2.2 System documentation	H5.2
17	1	9.2.2 Algorithm output	H4.2
18	1	9.2.2 Algorithm comparison	H4.2
19	1	9.2.2 Subroutine call	H4.3
20	1	9.2.2 Local/global variables	H4.2, H4.3

Section II

Question	Marks	Content	Syllabus outcomes
21	3	9.2.4 Testing report	H5.2
22	3	9.2.2 System documentation	H5.2
23 (a)	3	9.1.2 Installation	H1.2
23 (b)	3	9.2.2 System documentation	H5.2
23 (c)	3	9.3 Project management	H5.1
24	3	9.1.2 CASE tools	H1.2
25	3	9.2.2 Choosing a language	H2.1
26	3	9.2.5 Maintenance	H4.2, H4.3
27	3	9.1.2 Emerging technologies	H2.2
28 (a)	3	9.2.2 Desk check	H4.2
28 (b)	2	9.2.2 Error correction	H4.2
28 (c)	3	9.2.2 Search methods	H4.2
29	3	9.2.1 System documentation	H5.2
30	3	9.2.3 Fetch – execute cycle	H1.1, H1.3

Question	Marks	Content	Syllabus outcomes
31	3	9.2.3 EBNF Statements	H2.1, H4.2
32	4	9.2.2 Algorithm design	H4.2
33 (a)	3	9.1.1 Responsibility of developer	H3.1
33 (b) (i)	4	9.2.1 System documentation	H5.2
33 (b) (ii)	5	9.2.2 Algorithm design	H4.2

Section III

Question	Marks	Content	Syllabus outcomes
34 (a)	2	9.4.1 Polymorphism	H2.1
34 (b) (i)	3	9.4.1 OOP declarations	H4.2
34 (b) (ii)	2	9.4.1 OOP attributes	H4.2
34 (b) (iii)	4	9.4.1 OOP subclass	H4.2
34 (c)	4	9.4.1 Paradigm comparison	H5.3
34 (d) (i)	3	9.4.1 Chaining	H4.2
34 (d) (ii)	2	9.4.1 Logic rule	H4.2
35 (a)	3	9.4.2 Binary subtraction	H1.1, H1.3
35 (b)	3	9.4.2 Floating point	H1.1, H1.3
35 (c) (i)	3	9.4.2 Truth table	H1.3, H5.2
35 (c) (ii)	2	9.4.2 Adder circuit	H1.1, H1.3
35 (d)	3	9.4.2 Flip-flop circuit	H1.1, H1.3
35 (e)	2	9.4.2 Circuit design	H1.1, H1.3
35 (f)	4	9.4.2 Data streams	H1.1