2023 HSC Software Design and Development Marking Guidelines

Section I

Multiple-choice Answer Key

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

Section II

Question 21

Criteria	Marks
• Demonstrates a sound understanding of the use of prototyping	2
• Demonstrates some understanding of prototyping	1

Sample answer:

A prototype can be used to show screen design and navigation, to clarify understanding of the system and get client feedback.

Question 22

Criteria	Marks
• Explains the importance of communication over different stages in the software development cycle	3
• Provides ONE benefit of communication, with reference to a stage in the software development cycle OR • Provides some benefits of communication	2
• Demonstrates some understanding of the need for communication in software development	1

Sample answer:

Without a clear understanding of client needs and without ongoing communication during the defining and planning stages, there is a risk that the final product does not meet their needs or that it becomes increasingly expensive to make necessary changes.

Communication with the client during the testing stage is important in the development of relevant test data and for thorough evaluation of the software.

Question 23 (a)

Criteria	Marks
• Provides TWO reasons for the importance of security of this database	2
• Demonstrates some understanding of the importance of data security	1

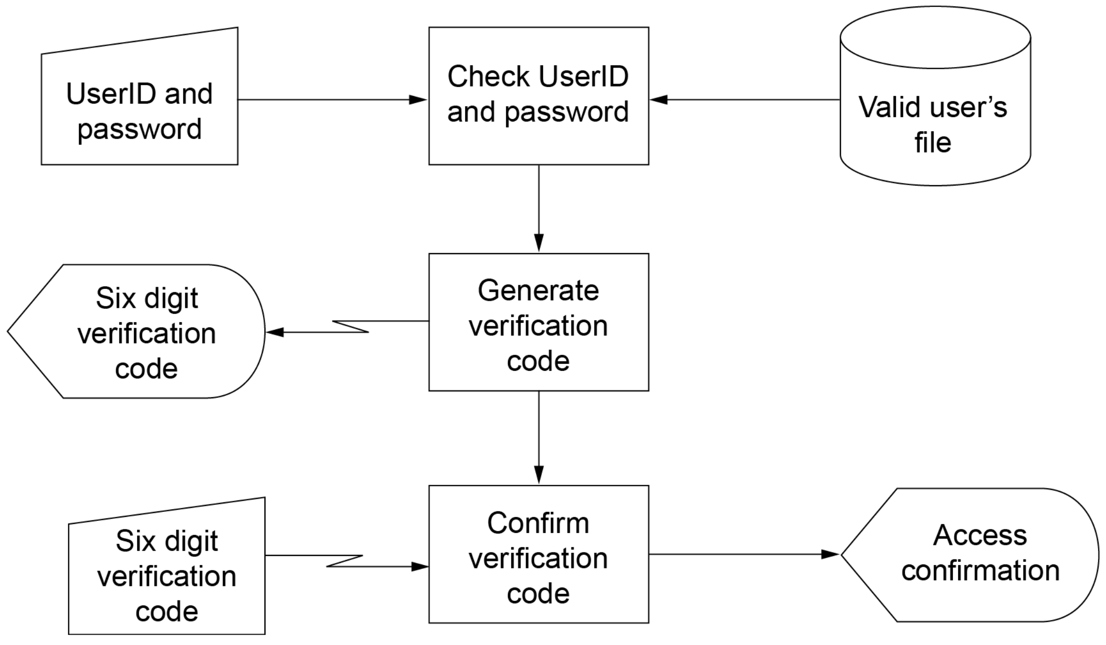
Sample answer:

Unintentional/malicious changes to the data may cause harm to the patients. Privacy of the patient may be compromised if the data is made public.

Question 23 (b)

Criteria	Marks
• Provides a substantially correct system flowchart for the authentication process, using appropriate symbols	3
• Provides a partially correct system flowchart	2
• Provides a diagram that shows some understanding of the process	1

Sample answer:


UserID and password
Check UserID and password
Valid user's file
Six digit verification code
Generate verification code
Six digit verification code
Confirm verification code
Access confirmation

Question 24

Criteria	Marks
• Outlines TWO problems that may arise with outsourcing one of the applications	3
• Outlines ONE problem that may arise with outsourcing one of the applications	2
• Demonstrates some understanding of outsourcing	1

Sample answer:

Outsourced developers may not produce an interface with the same look and feel as other applications in the suite.

By outsourcing the application, it may be more difficult to ensure the security of the project, since sensitive company information may need to be shared.

Question 25

Criteria	Marks
• Explains how features of software help maintain market share	4
• Explains how a feature of software helps maintain market share OR • Outlines features of software that helps maintain market share	3
• Identifies features of software that help maintain market share OR • Outlines a feature that helps maintain market share	2
• Provides some relevant information	1

Sample answer:

Software designed by dominant developers typically has consistent design across all of their applications. This leads to familiarity that encourages their customer base to continue to use their software as new products become available.

The software reflects current practice with up-to-date interfaces and functionality because the developers have the resources to regularly update their software, offering an enhanced user experience.

Answers could include:

Software produced by dominant developers can exchange data more easily through the use of compatible files which become industry standard, making it harder for other developers to compete.

Question 26 (a)

Criteria	Marks
• Demonstrates a sound understanding of why both Read statements are needed	2
• Demonstrates some understanding of a read statement	1

Sample answer:

Line 9 provides a priming read from the file and thus enables the EOF condition in line 10 to be tested, allowing for an empty file. Line 14 provides subsequent records for processing and provides the terminating condition when there are no more records.

Question 26 (b)

Criteria	Marks
• Provides TWO reasons why the code is easy to maintain • Includes examples from the code	3
• Provides ONE reason why the code is easy to maintain, using an example from the code OR • Provides TWO reasons why the code is easy to maintain	2
• Demonstrates some understanding of code maintenance	1

Sample answer:

The use of meaningful variable and module names, such as UserID and CheckPasswordCorrect, assists maintenance programmers to quickly understand the purpose of the code.

Including calls to subroutines in lines 5 and 12 enables maintenance programmers to easily locate the detailed logic that may need to change.

Question 26 (c)

Criteria	Marks
• Provides a thorough explanation of the need for both module and system testing in this scenario	4
• Demonstrates a sound understanding of the need for module and system testing	3
• Demonstrates some understanding of module and/or system testing	2
• Demonstrates some understanding of testing	1

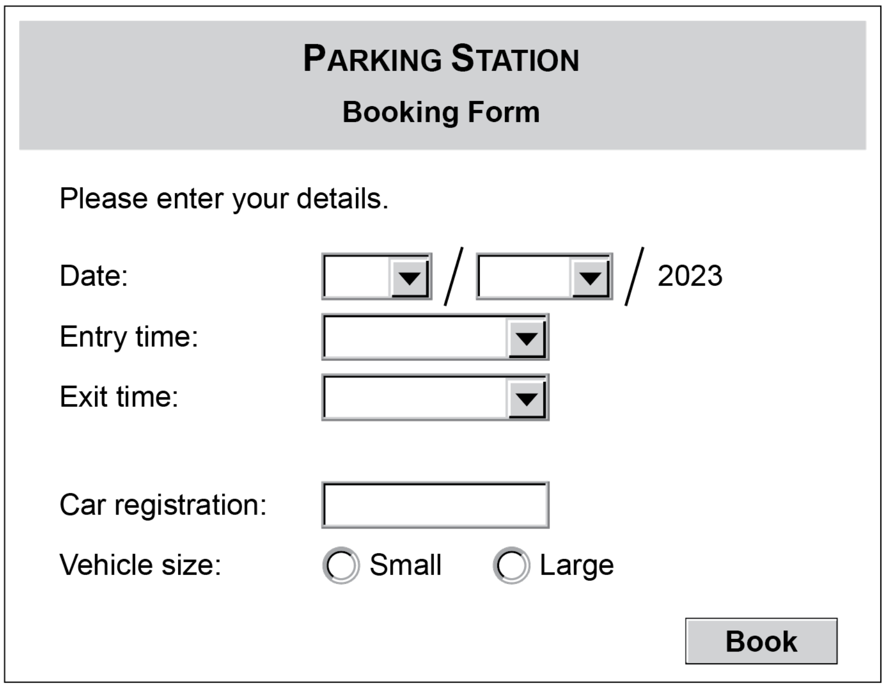
Sample answer:

Module testing will confirm that the module works, including the logic of the called modules with parameters being passed appropriately, for example when calling CheckPasswordCorrect with its three parameters. During system testing, live data will provide expected volumes of data which may show potential issues with performance and response times when accessing both files. Live data also ensures that unexpected combinations of data are well handled compared to the minimal conditions when using selected test data. It also tests for missing files or inappropriate file types or record formats.

Question 27 (a)

Criteria	Marks
• Produces a well-designed interface which uses appropriate screen design elements	3
• Produces an interface with some appropriate elements	2
• Provides a diagram that demonstrates some understanding of the system	1

Sample answer:



PARKING STATION

Booking Form

Please enter your details.

Date: / / 2023

Entry time:

Exit time:

Car registration:

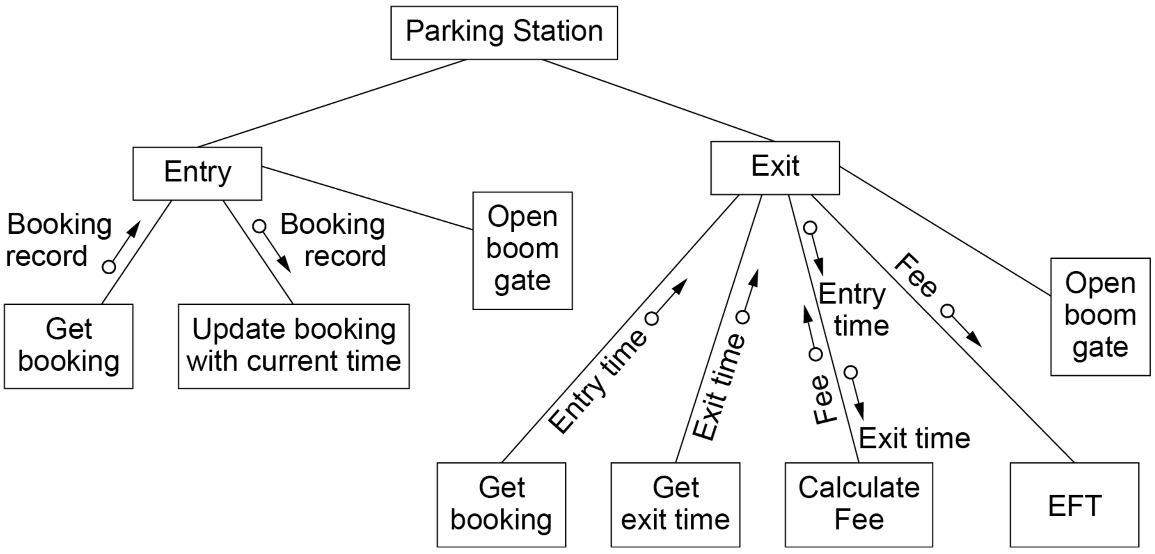
Vehicle size: Small Large

Book

Question 27 (b)

Criteria	Marks
• Produces a substantially correct structure chart that represents the system	4
• Produces a structure chart that represents most components of the system	3
• Produces a chart with some relevant components	2
• Provides a diagram that demonstrates some understanding of the system	1

Sample answer:



Question 27 (c)

Criteria	Marks
• Describes how the system can be evaluated with reference to TWO appropriate quality assurance criteria	3
• Shows some understanding of quality assurance	2
• Provides some relevant information	1

Sample answer:

The functionality of the system could be evaluated by viewing samples of transactions and checking that the correct fees were charged, based on the recorded entry and exit times.

The ease of use of the system could be evaluated by presenting a sample of users with a survey requiring them to comment on their experiences of the process and analysing the responses.

Question 28 (a)

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

Sample answer:

OK	ProductCode	Len	Extracted character	Count	Display
F	p3	2			
T			p		
F					Not uppercase
			p	1	
F			3	2	Not all lowercase
					Invalid code
F	Pad		P		
T			P	1	Not all lowercase
F			a	2	
					Invalid code
T	Q				

Note: X is not considered

Question 28 (b)

Criteria	Marks
• Identifies TWO errors and suggests ways to correct them	3
• Identifies ONE error and suggests a way to correct it OR • Identifies TWO errors	2
• Identifies an error	1

Sample answer:

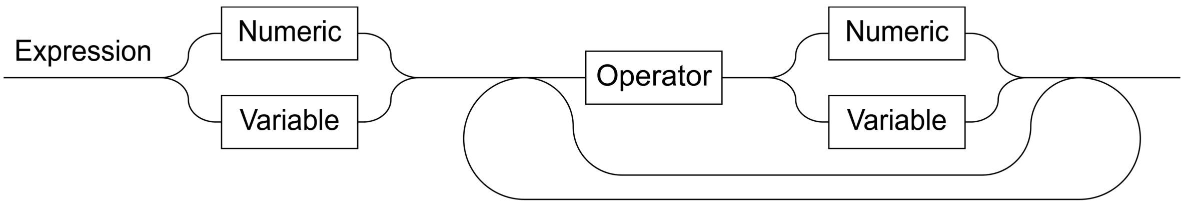
At line 110 the FOR/NEXT loop counter should start at 2.

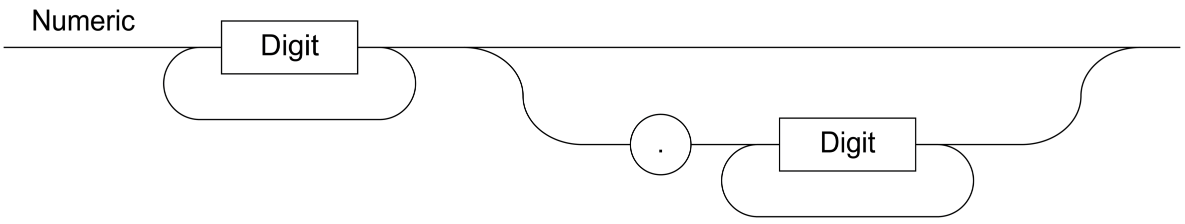
The value of Len is not being updated for the next ProductCode. Move line 40 to line 55.

Question 29

Criteria	Marks
• Provides substantially correct railroad diagram(s)	4
• Provides railroad diagram(s) that satisfies most of the requirements	3
• Provides a partially correct railroad diagram OR • Provides a substantially correct EBNF	2
• Demonstrates some understanding of the syntax rules for expressions	1

Sample answer:





Question 30

Criteria	Marks
• Provides a substantially correct algorithm that addresses the following features: – Display menu screen – Get a non-zero ButtonClick – Perform correct subroutine – Loop until Quit (9) is selected	3
• Provides an algorithm that attempts to address some of the features	2
• Demonstrates some understanding of the problem	1

Sample answer:

```
BEGIN Menu
    REPEAT
        Display MenuScreen
        REPEAT
            Get ButtonClick
        UNTIL ButtonClick <> 0
        CASEWHERE ButtonClick is
            1           :   Screen1
            2           :   Screen2
            3           :   Screen3
            4           :   Screen4
            Otherwise   :   Quit
        ENDCASE
    UNTIL ButtonClick = 9
END Menu
```

Question 31

Criteria	Marks
• Provides a sound explanation of why linking is necessary	3
• Shows some understanding of linking	2
• Provides some relevant information	1

Sample answer:

A compiler produces object code from the provided source code, including calls to any library routines or externally compiled subroutines. The linking process locates the executable code for each of these external routines and links them with the generated object code, to create one complete self-contained executable file.

Question 32 (a)

Criteria	Marks
• Provides THREE items of test data for different aspects of the algorithm, with corresponding reasons for inclusion	3
• Provides TWO items of test data for different aspects of the algorithm, with some justification	2
• Demonstrates some understanding of test data relevant to this algorithm	1

Sample answer:

Test data	*Reason for inclusion*
01/15/2022	Check that the month is valid (1−12)
31/06/2023	Check correct number of days in month
19/04/2022	Verify that correct long date is produced

Question 32 (b)

Criteria	Marks
• Provides a substantially correct algorithm that addresses the following features: – Validates the inputted date – Uses appropriate data structures and library routines – Outputs the correct long date – Provides meaningful error messages	5
• Provides a substantially correct algorithm that addresses most of the features	4
• Provides an algorithm that addresses some of the features	3
• Provides an algorithm that attempts to address one of the features	2
• Demonstrates some understanding of the problem	1

Sample answer:

```
BEGIN date_conversion

    Get InDate
    MM = Extract(InDate, 4, 2)
    Month = Value (MM)
    DD = Extract(InDate, 1, 2)
    Days = Value (DD)
    YYYY = Extract(InDate, 7, 4)

    validmonth = CheckMonth(Month)
    IF validmonth THEN
        validday = CheckDays(Month, Days)
        IF validday THEN
            Display DD + " " + MonthName(Month) + " " + YYYY
        ELSE
            Display "Your days" + DD "are not valid for your entered month"; MM
        ENDIF
    ELSE
        Display "Your entered month must be a valid number between 1 and 12"
    ENDIF
END date_conversion

BEGIN CheckMonth(MM)
    flag = 1
    IF MM < 1 OR MM > 12 THEN
        flag = 0
    ENDIF
    RETURN flag
END CheckMonth

BEGIN CheckDays(Month, Days)
    flag = 1
    IF Days > DaysinMonth(Month) OR Days < 1 THEN
        flag = 0
    END IF
    RETURN flag
END CheckDays
```

Section III

Question 33 (a) (i)

Criteria	Marks
• Provides the facts that need to be defined	2
• Provides some relevant information	1

Sample answer:

gpop(f_city, g_city)
hcrime(g_city, f_city)

Question 33 (a) (ii)

Criteria	Marks
• Provides the correct rule	2
• Provides some relevant information	1

Sample answer:

Better_City(X,Y) :- gpop(X,Y), hcrime(Y,X)

Question 33 (a) (iii)

Criteria	Marks
• Explains why the logic paradigm is more appropriate than the imperative paradigm for developing the system	3
• Shows some understanding of the logic and/or imperative paradigm	2
• Provides some relevant information	1

Sample answer:

If the imperative paradigm is used, the developer needs to specify the code for each process in solving queries, requiring a lengthy series of IF statements. It is cumbersome to add new cities and incorporate new queries. Therefore, using the imperative paradigm makes initial coding and subsequent maintenance tedious.

Using a logic paradigm means all the developer needs to do is to define appropriate rules and facts. The processing in the inference engine provides the necessary logic to solve queries, making the initial development and maintenance easier.

Question 33 (b)

Criteria	Marks
• Explains a benefit of instantiation and includes a relevant example	3
• Demonstrates some understanding of instantiation	2
• Provides some relevant information	1

Sample answer:

Instantiation can be helpful in game development as there is usually a need to create many similar objects of the same class. For example, a background might need multiple instances of a tree class, such as a pine tree and wattle tree. This helps reduce development time as there is no need to write code for each individual tree.

Question 33 (c)

Criteria	Marks
• Demonstrates a thorough understanding of factors that influence the choice of paradigm	4
• Demonstrates a sound understanding of factors that influence the choice of paradigm	3
• Demonstrates some understanding of paradigm selection	2
• Provides some relevant information	1

Sample answer:

Type of Problem: If the project involves the simple definitions of facts and rules, the logic paradigm is the obvious choice.

Programmer Expertise: Developers often have a paradigm of preference and skill, so to move to other paradigms might require a steep learning curve. However, their choice to use this paradigm may be affected by the nature of the project.

Other modules: Depending on the paradigm already used in the development of other modules in the project, it might restrict the possible choice of paradigm for this new module.

Question 33 (d)

Criteria	Marks
• Explains why the OOP paradigm is appropriate for this software, with reference to OOP concepts	3
• Describes why ONE concept in the OOP paradigm is relevant to this software	2
• Provides some relevant information	1

Sample answer:

Inheritance: There is a superclass called 'dog' with subclasses for each life stage. Many attributes and methods common to each subclass are inherited from the superclass.

Polymorphism: Each instance of a subclass behaves slightly differently to the same input (a command). For example, a dog will react differently to a command given by its owner and by a stranger.

These concepts make OOP appropriate as there will not be a need to write code defining the attributes and methods of each subclass. In addition, it gives the programmer the scope to have similar methods react differently to the same input.

Question 33 (e)

Criteria	Marks
• Explains how both heuristics and goals could be used in the software for this scenario	3
• Describes how goals and/or heuristics could be used in the software for this scenario	2
• Demonstrates some understanding of goals or heuristics	1

Sample answer:

With changing conditions, it is impractical to code a set path from the kitchen to the desired table. Therefore, a combination of goals and heuristics should be used in this case as the path changes based on the conditions found. The goal that would be coded is the map location of the table. Heuristics could be included to find a reasonable path that leads to the goal in a short time frame, as needed in a restaurant. This path would change when conditions cause an obstacle to block it.

Question 34 (a)

Criteria	Marks
• Completes the table correctly	2
• Provides some relevant information	1

Sample answer:

Hexadecimal representation	*Binary representation*
B	1011
9	1001
1E	00011110

Question 34 (b)

Criteria	Marks
• Provides a correct binary calculation with working and shifting shown	3
• Provides a substantially correct binary calculation	2
• Demonstrates some understanding of binary multiplication	1

Sample answer:

These bits were shifted two places

$$\begin{array}{r} 110110\ \times \\ 101 \\ \hline 110110 \\ \boxed{110110}00 \\ \hline 100001110 \end{array}$$

Question 34 (c)

Criteria	Marks
• Provides a comparison that demonstrates a sound understanding of both representations	3
• Shows some understanding of integers and/or floating point numbers	2
• Provides some relevant information	1

Sample answer:

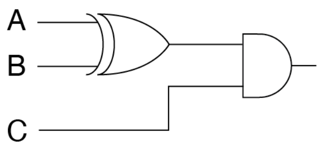
Signed 32-bit integer representation is not applicable to positive numbers less than 1 or greater than $2^{15} - 1$. Within that range, integer values are represented exactly.

Using the same number of bits, single precision floating point representation allows much larger (10^{38}) and very small (10^{-38}) numbers but with limited accuracy (approximately 6 decimal places).

Question 34 (d)

Criteria	Marks
• Provides a correct circuit with logic gates	3
• Provides a substantially correct circuit OR • Provides a correct truth table OR • Provides a correct Boolean expression	2
• Provides some relevant information	1

Sample answer:



Question 34 (e) (i)

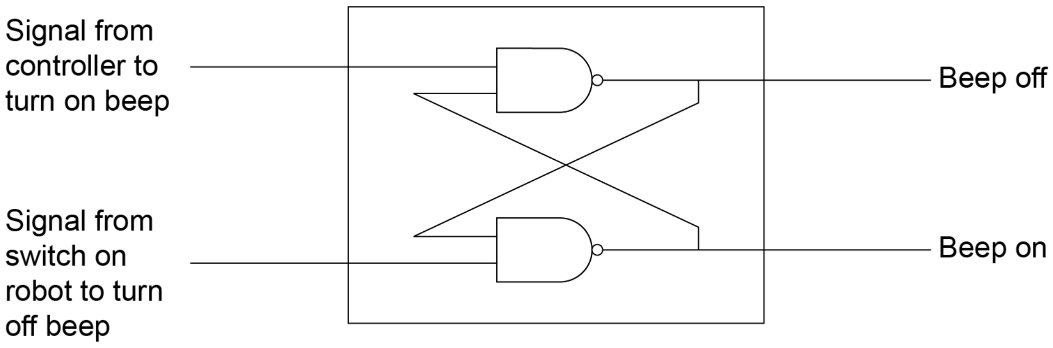
Criteria	Marks
• Explains how a flip-flop can be used in the circuit	3
• Demonstrates some understanding of the use of a flip-flop in the circuit	2
• Provides some relevant information	1

Sample answer:

A flip-flop continues to output a signal once it receives input on the SET line, until a signal on the RESET line is received.

The SET input comes from the controller resulting in a signal to produce a continuous beep. The RESET input comes from the switch pressed by the staff member causing the beep to stop.

Answers could include:


Signal from controller to turn on beep
Signal from switch on robot to turn off beep
Beep off
Beep on

Question 34 (e) (ii)

Criteria	Marks
• Provides the correct data stream	2
• Demonstrates some understanding of the relevant data stream	1

Sample answer:

101 0110 1000

Question 34 (e) (iii)

Criteria	Marks
• Provides a substantially correct algorithm that addresses the following features: – identifies the relevant robot and table number – loops through the array to determine the coordinates – generates the data stream – outputs the data stream	4
• Provides a substantially correct algorithm that addresses some of the required features	3
• Provides an algorithm that attempts to address some of the features	2
• Demonstrates some understanding of the problem	1

Sample answer:

```
BEGIN DirectRobot(RobotID, TableNo)
    GetCoords (TableNumber, X, Y)
    Output Bin(RobotID,3), Bin(X,4), Bin(Y,4)
END DirectRobot

BEGIN GetCoords (TableNumber, X, Y)
    For i = 1 to 10
        For j = 1 to 10
            IF Room(i,j) = Tablenumber THEN
                X = i
                Y = j
            END IF
        NEXT
    NEXT
END GetCoords
```

2023 HSC Software Design and Development Mapping Grid

Section I

Question	Marks	Content	Syllabus outcomes
1	1	9.2.3 Documentation (developer)	H5.2
2	1	9.1.2 CASE tools	H5.3
3	1	9.2.3 User and developer documentation	H5.1
4	1	9.2.3 Errors / translation	H4.2
5	1	9.2.4 Post implementation review	H5.1
6	1	9.2.3 Syntax diagrams	H4.2
7	1	9.2.2 Global variables	H1.3
8	1	9.2.3 Software development cycle	H4.2
9	1	9.1.1 Software piracy and copyright	H3.1
10	1	9.2.2 Data structures	H1.3
11	1	9.2.3 Algorithm efficiency	H4.3
12	1	9.2.3 Error detection methods	H4.3
13	1	9.2.1 Documentation DFD	H5.2
14	1	9.2.3 Fetch / execute cycle	H1.1
15	1	9.2.2 Library routines	H4.3
16	1	9.2.3 Control structures / logical operators	H4.3
17	1	9.2.2 Sentinel values	H4.2
18	1	9.2.2 Sorting methods	H4.2
19	1	9.2.3 Translation	H1.1
20	1	9.2.2 Match algorithm to purpose	H4.3

Section II

Question	Marks	Content	Syllabus outcomes
21	2	9.1.2 Software development approaches	H4.2
22	3	9.2.1, 9.2.4, 9.3 Software development cycle	H6.1, H6.3
23 (a)	2	9.1.1 Security and privacy	H3.1
23 (b)	3	9.2.1 Systems documentation	H5.2
24	3	9.1.2 Outsourcing	H6.3
25	4	9.1.1 Software market	H3.1
26 (a)	2	9.2.2 Priming read	H4.3
26 (b)	3	9.2.3 Writing for maintenance	H4.3
26 (c)	4	9.2.4 Testing	H4.2, H4.3
27 (a)	3	9.1.1 Interface design	H3.2, H6.4
27 (b)	4	9.2.1, 9.2.2, 9.3 Structure chart	H6.2
27 (c)	3	9.2.1, 9.2.4 Quality assurance	H5.1
28 (a)	3	9.2.2, 9.2.3 Desk check	H4.2

Question	Marks	Content	Syllabus outcomes
28 (b)	3	9.2.3 Debugging	H4.3
29	4	9.2.3 Syntax diagrams	H4.2
30	3	9.2.3, 9.3 Algorithm design	H4.3
31	3	9.2.3 Linking	H4.2
32 (a)	3	9.2.4 Test data	H4.3
32 (b)	5	9.2.3, 9.3 Algorithm design	H3.2, H4.3

Section III

Question	Marks	Content	Syllabus outcomes
33 (a) (i)	2	9.4.1 Facts and rules	H4.2
33 (a) (ii)	2	9.4.1 Facts and rules	H4.2
33 (a) (iii)	3	9.4.1 Programming paradigms	H1.2
33 (b)	3	9.4.1 OOP paradigm	H2.1
33 (c)	4	9.4.1 Programming paradigms	H1.2, H2.1
33 (d)	3	9.4.1 OOP concepts	H2.1, H2.2
33 (e)	3	9.4.1 Logic programming	H2.1, H4.2
34 (a)	2	9.4.2 Binary and hexadecimal representation	H1.3
34 (b)	3	9.4.2 Binary arithmetic	H1.3
34 (c)	3	9.4.2 Floating point representation	H1.3
34 (d)	3	9.4.2 Circuits	H1.1
34 (e) (i)	3	9.4.2 Flip–flops	H1.1, H2.2
34 (e) (ii)	2	9.4.2 Data stream	H1.3
34 (e) (iii)	4	9.4.2 Data stream algorithm	H1.1, H4.1, H4.3