



General Certificate of Secondary Education

Digital Technology

Unit 4

Digital Development Concepts

[GDG41]

Assessment

**MARK
SCHEME**

- 1 (a) B Data validation ensures that data entered is complete and falls within specified boundaries. [1]
- (b) D Object-oriented programming uses classes and inheritance whereas procedural programming uses procedures and top-down design. [1]
- (c)
- | A | B | C = A OR B | D = NOT(B AND C) |
|---|---|------------|------------------|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 |
- [1]
[1]
[1]
[1] [4]
- (d)
- | Definition | Key Term |
|---|-------------------------|
| Observing key characteristics and trends in the data being considered | Pattern Recognition [1] |
| Removing specific details from a problem which are not required to solve it | Abstraction [1] |
| Breaking large complex problems into smaller problems | Decomposition [1] |
- [3]
- (e) C A dry run is a paper-based exercise and the programmer goes through the solution step by step. [1]
- 2 (a) Any **two** from:
 Provides an (integrated) set of tools for programmers [1] which enables applications to be built from coding to testing [1]
 Provides editing tools [1] Accept examples of editing tools eg. clipboard, use of colour to highlight syntax error or key words, collapsible code, line numbering, code completion tool
 Provides debugging tools [1]
 Provides compiling tools [1]
 Provides GUI features for building forms [1] [2]
- (b) (i) So that it can be executed [1]/understood by the computer [1]/only understands 0s and 1s (binary) [1]/to change to object code [1]
- (ii) Any **three** from:
 Library code is included [1] pre-processing directives included [1]
 Syntax analysis [1] / errors identified [1]
 Semantic analysis [1]
 Machine code is generated [1]
 Lexical analysis [1] [3]

10

Error description	Error type
A keyword has been spelt incorrectly	Syntax Execution Logic [1]
A calculation includes a division by zero	Syntax Execution Logic [1]
An incorrect result is output from a program	Syntax Execution Logic [1]

Candidates circling more than one answer on each line should receive 0 marks for that line

[3]

AVAILABLE MARKS

9

3 (a) (i)

Variable	Data Type
highScore	integer/numeric [1]
grade	character/string [1]

[2]

(ii) Sample answer, code acceptable:

Any **seven** from:

Use of if statement [1]

Comparison (≥ 5000) [1]

Correct range comparison (e.g. ≥ 3000 [1] and [1] ≤ 4999 [1])

Correct grade assignment in 1 case [1]

All four grade assignments correct [1]

Output grade statement [1]

Alternative solutions are acceptable provided the logic is correct. Component marks listed should be applied in the same way to every solution.

These three lines are supplied in the stem of the question

OUTPUT "Enter high score"

INPUT highScore

If highScore > 10000

 grade= 'X'

Else

If highScore \geq 5000 and highScore \leq 10000

 grade='M'

Else

If highScore \geq 3000 and highScore \leq 4999

 grade='L'

Else

 grade='N'

OUTPUT grade

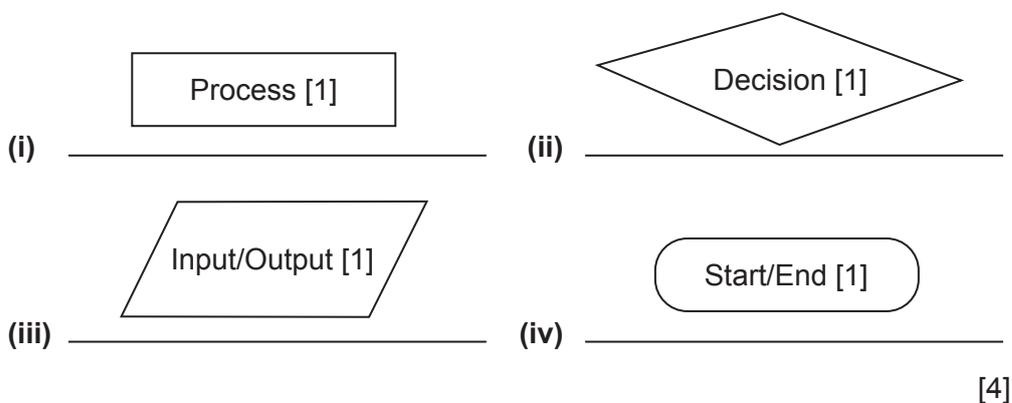
[7]

- (b) Any **four** from
 code can be re-used [1]
 shortening development time [1]
 code can be tested/debugged in sections [1]
 so errors can be identified quickly [1]
 more robust software [1] [4]

13

- 4 (a) (i) real/double/float (numeric acceptable) [1]
- (ii) use the list or array name [1] followed by the index/position of the element in the array/list [1]
 [1] for a suitable example, e.g. temperatures [2]. [2]
- (iii) contents are stored together in memory [1] therefore faster processing/
 updating/deleting of data [1] [2]

(b)



- (c) Any **nine** from:
 Start/End Symbol [1]
 Declare variables in a process box [1]
 Initialise variables in a process box [1]
 Check counter value ≤ 7 in decision box [1]
 Loop indicated after decision box with Yes and No routes correct [1]
 Add temperatures [counter] to running total (total=total [1] +temperatures [counter]) in a process box [1]
 Increment counter in process box [1]
 Calculate average as total/7 in process box [1]
 Output average in an input/output box [1] [9]

- (d) Any **four** from:
 Comparison of two adjacent elements [1]
 A swapped item is added to the sorted sublist [1] in the correct position [1]
 A sorted sublist is created [1]
 OR
 Sample code or pseudo code for the Insertion sort showing:
 2 loops [1]
 Position of minimum [1]
 If statement [1]
 Swap numbers [1] [4]

22

Statement	True/False	
When searching through large amounts of data the binary search algorithm is less efficient than the linear search algorithm	False	[1]
The binary search algorithm requires data to be sorted	True	[1]
The binary search algorithm examines all data items in a list until the target value is found or until there are no more data items to examine	False	[1]
The binary search algorithm starts by finding the mid-location in a list of data items	True	[1]

[4]

- (b) Any **two** from:
 Create a new file [1] and suitable explanation or example [1]
 Open a file [1] and suitable explanation or example [1]
 Read data from a file [1] and suitable explanation or example [1]
 Write data to a file [1] and suitable explanation or example [1]
 Append data to a file [1] and suitable explanation or example [1]
 Delete data from a file [1] and suitable explanation [1]
 (2 × [2])

[4]

8

- 6 (a) Conversion work – accept divide by two or place value [1]
 101001 [1]

[2]

- (b) Conversion work – accept power of 2 or place value [1]
 Conversion work – addition of results [1]
 251 [1]

[3]

- (c) Any **three** from:
 Conversion work – split bit pattern [1]
 5A [1]
 Conversion work 5 [1]
 Conversion work A [1]

[3]

- (d) Any **three** from:
 Base 16 or binary conversion [1]
 34 [1]
 Hex conversion 3 [1]
 Hex conversion 4 [1]

[3]

- (e) (i) The magnitude of the number is greater than the maximum [1] number that can be represented by the computer [1]

[2]

(ii)

	1	1	1	0	0	0	1	1
+	0	0	1	1	1	1	1	1
(1)	0	0	1	0	0	0	1	0

- Correction result (not including overflow [1])
 Correct overflow [1]
 Overflow identified [1]

[3]

- (f) The 7-bit ASCII table could represent **128** [1] characters whilst the 8-bit ASCII table could represent **256** [1] characters. The **ASCII** [1] characters set is a subset of the **UNICODE** [1] character set.

[4]

20

AVAILABLE MARKS

7 Level 0 [0]

Answer is not worthy of credit.

Level 1 ([1]–[2])

The candidate refers to one [1] or two [2] of selection and iteration in the correct context. The candidate makes limited use of spelling, punctuation and grammar. The meaning of the text is not always clear. The candidate displays a limited form and style appropriate to the question. The organisation of the answer is limited.

Level 2 ([3]–[4])

The candidate describes how one [3] or two [4] of selection and iteration are used in program control. The candidate makes satisfactory use of spelling, punctuation and grammar. The meaning of the text is usually clear. The candidate demonstrates a satisfactory form and style appropriate to the question. The organisation of the answer is satisfactory.

Level 3 ([5]–[6])

The candidate fully describes how both selection and iteration are used in program control. The candidate uses a good standard of spelling, punctuation and grammar. The meaning of the text is always clear. The candidate demonstrates a good standard of form and style appropriate to the question. The organisation of the answer is good.

Answers may include:

Selection

Making use of conditions

Selecting statements that are to be executed based on the evaluation of conditions

Implemented using if statements

Not all statements are executed in an if statement

Suitable example

Iteration

Making use of conditions or counters

Repeating statements that are to be executed based on the evaluation of conditions or counters

Implemented using for, while, repeat loops

Reference to bounded or unbounded iteration

Suitable example

[6]

6

AVAILABLE
MARKS

8 (a) (i) **Boolean/Bool/char/string** [1]

(ii) Any **one** from:
 Boolean/Bool : only two states Y/N [1]
 Char: made up of a single character 'Y/'N' to represent Yes or No [1]
 String: a string of length 1 to represent Yes or No [1]

(iii) Range check [1]

(b) Sample answer accept code or algorithm

Input [1]
If statement: **If** *requiredTemp* <18 [1] OR [1] *requiredTemp* > 24 [1]
 (or alternative)
 Error message [1]
 Use of loop [1]
 Loop condition [1]
 Set valid [1]

Alternative solutions are acceptable provided the logic is correct.
 Component marks listed should be applied in the same way to every solution.

AVAILABLE MARKS

<pre>do valid=true OUTPUT prompt INPUT <i>requiredTemp</i>. if <i>requiredTemp</i> <18 OR <i>requiredTemp</i> > 24 valid=false OUTPUT error message end if clear error message WHILE valid=false</pre>	<pre>Valid=false while(valid!=true) OUTPUT prompt INPUT <i>requiredTemp</i>. if <i>requiredTemp</i> >=18 and <i>requiredTemp</i> <= 24 valid=true OUTPUT error message end if clear error message</pre>
--	---

[8]

(c) Any **two** from:
 Length [1] plus explanation or example [1] (e.g. could be used to ensure the system code is exactly 4 letters long)
 Split [1] plus explanation or example [1] (e.g. could be used to check individual characters or substrings in the system code)
 Substring [1] plus explanation or example [1] (e.g. could be used to check individual characters or substrings in the system code)
 Substring searching [1] plus explanation or example [1] (e.g. could be used to find particular characters or substrings in the system code)
 Change to lower/uppercase [1] plus explanation or example [1] (e.g. could be used to change data entered by user into uppercase for storing)
 Concatenate [1] plus explanation or example [1]
 (2 × [2])

[4]

15

			AVAILABLE MARKS
9	<p>(a) Testing and making changes [1] at different stages in development [1] [2] Repeated testing [1]</p> <p>(b) (i) Any two from: Carried out on one module of code [1] Makes use of the module specification [1] Detailed testing of scope [1] [2]</p> <p>(ii) Any two from: Carried out on completed system/after integration testing [1] Help ensure correct outputs produced/user requirements met [1] Use black box testing methods [1] [2]</p> <p>(iii) Any two from: Used to ensure that the units of the system work together [1] Focus on data passed between units [1] Tests the interfaces between units/modules [1] [2]</p> <p>(c) Any two from: Valid [1] Invalid [1]/erroneous Null [1] Extreme [1] [2]</p>	<p>[2]</p> <p>[2]</p> <p>[2]</p> <p>[2]</p> <p>[2]</p>	10
10	<p>(a) Any two from: To improve the product being developed [1] To identify problems with the product early in the development process [1] To gain feedback from the end user during development [1] [2]</p> <p>(b) Any two from: To enable improvement to be made at an early stage [1] Changes are less costly if made early in the development process [1] Allows continuous user feedback [1] [2]</p> <p>(c) (i) Compare with the user requirements [1]</p> <p>(ii) Any two from: Perform black/white box testing [1] Design a thorough test plan [1] Test the system using high volumes of incorrect data/invalid/ exceptional data [1] [2]</p>	<p>[2]</p> <p>[2]</p> <p>[1]</p> <p>[2]</p>	7
Total			120