Rewarding Learning

General Certificate of Secondary Education
2019

Digital Technology

Unit 4

Digital Development Concepts

[GDG41]

THURSDAY 16 MAY, AFTERNOON

MARK SCHEME

AVAILABLE MARKS

1 **(a)**

Data Item	Data Type
NameOfTest	String/alphanumeric/text
Score	numeric/Integer/Double/Real/float/Int
Average	numeric/Integer/Double/Real/float/Int

[3]

(b) VALIDATION is the action of checking data to ensure it is acceptable. A RANGE check will be required to ensure that the value entered for the Score is valid. A PRESENCE check will ensure that NameOfTest1 is entered. [3]

(c) Sample answer
INPUT NameOfTest4 [1]
INPUT Score4 [1]
Average = (Score1 + Score2 + Score3 + Score4) [1]/[1] 4 [1]
OUTPUT "The average is" Average [1] [6]

(d) Any **three** from:
Clipboard [1]
Colour coded text/colour coded functions (note to examiners for example, syntax errors are underlined in red in the C# coding editor) [1]
Collapsible code sections [1]
Line Numbering [1]
Code Completion Tools [1]
IntelliSense [1]/syntax error assistance
Auto completion/auto indent [1] [3]

(e) **(i)** Any **two** from:
The program must be turned into machine code/0s and 1s/binary [1] so that the computer can understand it [1] so that it can be executed [1] [2]

(ii) Any **two** from:
Processes each statement in the source code [1]
Translates the program into machine code [1]
Checks syntax (of each statement) [1]/syntax analysis
Lexical analysis [1]
Sementic analysis [1]
Include library code [1] [2]

19

				AVAILABLE MARKS
2	**(a)**	Variable kiloMetres/hikingDistance [1] Constant X or 1.61 [1]	[2]	
	(b)	Any **two** from: Variables: a memory location [1] given a name [1] used to hold data [1] which can change value [1] Constants value that cannot be changed [1] while the program is running [1] given a name [1] used to hold data [1] a memory location [1]	[4]	
	(c)	Sample answer accept algorithm, python, c# or VB For i=1 to 5 Input miles totalmiles=totalmiles+miles end For kilometres=totalmiles*1.61 output “The total distance in kilometres is” kilometres Any assignment or initialisation [1] Input [1] Running total [1]/total Correct calculation [1]/conversion Output [1] Loop [1] Condition or For [1]	[7]	13
3	**(a)**	B Test case 2	[1]	
	(b)	A Test case 1	[1]	
	(c)	C Test case 3	[1]	3

4

Definition	Program Construct
Executing all lines of code in a program	SEQUENCE [1]
Executing lines of code based on the outcome of an IF-Statement	SELECTION [1]
Executing lines of code repeatedly	ITERATION [1]

[3] 3

		AVAILABLE MARKS
5	Syntax errors Any **two** from: code does not conform to the (grammar) rules of the language [1] program cannot run with syntax errors/syntax errors must be corrected before the program can run [1] suitable example, e.g. leave out a semi-colon or mis-spell a keyword [1] occurs at compile time [1] Logic errors Any **two** from: causes unexpected behaviour [1] causes incorrect output [1] bug in the program [1] suitable example, e.g. incorrect condition in an If-statement [1] Occur at runtime [1]/during execution Program will still run [1] [4]	4
6	**Level 0 [0]** Answer is not worthy of credit. **Level 1 ([1]–[2])** The candidate refers to one [1] or two [2] of abstraction and decomposition. 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 one [3] or two [4] of abstraction and decomposition. 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 abstraction and decomposition. 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: Reference to Abstraction – filtering out details about the problem that will not be required for the solution Reference to Decomposition – break the complex problem down into smaller more manageable problems called sub-problems egs of abstraction, decomposition and advantages. [6]	6

AVAILABLE MARKS

7 **(a)** American Standard Code for Information Interchange [1]

(b)

Statement	True or False
The original ASCII code table used seven bits to represent text.	TRUE [1]
There were 64 characters in the original ASCII code table.	FALSE [1]
Unicode was created to increase the number of characters that could be represented in a computer system.	TRUE [1]
Unicode incorporates the ASCII character set	TRUE [1]

[4] 5

8 **(a)** conversion work accept divide by two or place value [1]
00010101 [1] [2]

(b) conversion work (headings or divide by 16) [1]
1 mark for 15 = F [1]
accurate answer – 1F [1] [3]

(c) colour codes on a computer system [1] [1]
reducing memory [1]
XML/XTML characters [1]
memory addresses/locations [1]
unicode characters UT [1]
Assembly language [1]

(d) **(i)** (1)10000010
correct answer (not including overflow) [1]
indicating overflow [1]
correct use of carry anywhere [1] [3]

(ii) magnitude of number [1] exceeds the range [1]
value generated by the addition [1] is outside of the range/exceeds that which can be stored by the bit pattern [1] [2]

(iii) the overflow digit is (dropped/ignored/lost) [1] the result may be incorrect [1] [2]

(e)

P	Q	R = P and Q	S = R or Q
0	0	0	0
0	1	**0** [1]	**1** [1]
1	0	**0** [1]	**0** [1]
1	1	1	1

(accept false for 0 and true for 1) [4] 17

AVAILABLE MARKS

9 **(a)** **(i)** Any **two** from:
use a variety of test data [1]
Normal/erroneous/extreme [1]
use high volumes of test data [1]
use test data that covers all pathways in the program [1] [2]

(ii)

Data item	Data type	Sample valid data
frameSize	String [1]	A1
numberOfFrames	Numeric/Integer/ int [1]	25 (Any number between 10–40) [1]
discountDue	Real/Double/float [1]	10.00 (must be less than 20.00) [1]

[5]

(b) **(i)** Sample answer accept code or algorithm

```
do
  valid=true
  OUTPUT prompt
  INPUT numberOfFrames
   if numberOfFrames <10 OR numberOfFrames > 40
      valid=false
      OUTPUT error message
   end if
  clear error message
WHILE valid=false
```

or

```
Valid=false
WHILE valid!=true
INPUT numberOfFrames
if numberOfFrames >=10
AND numberOfFrames <=40
  valid=true
ELSE
OUTPUT error message
END WHILE
```

use of loop [1]
set valid [1]
input [1]
if numberOfFrames <10 [1] OR [1] numberOfFrames > 40 [1]
or alternative above
reset valid [1]/correct condition in loop
error message [1] [8]

AVAILABLE MARKS

(ii)

Test number	Item to be tested	Reason for test	Test data	Expected outcome
1.	numberOfFrames	Extreme value	10	Value accepted [1]
2.	numberOfFrames	Valid data/value [1]	39	Value accepted
3.	numberOfFrames	Invalid data/value [1]	45	Error message/ value rejected [1]
4.	numberOfFrames	NULL data	Press enter key [answer must be placed in the box. Blank answer not acceptable [1]	Error message/ value rejected [1]

[6]

(c)

```
cost = numberOfFrames      * priceOfFrame                [1]
IF numberOfFrames          >=20 [2] or >20 [1]           [2]
    discountDue = cost     * 0.05                        [1]
END IF
totalCost = cost – discountDue                           [1]
```

[5]

26

10 (a) (i) BLACK BOX TESTING
Any **two** from:
Design or structure being tested is unknown by the person conducting the test [1]
Tester approaches the system as a black box which they cannot see into [1]
Identify errors relating to the interface/or how it operates [1]
Test for missing/incorrect functions/data structures [1] /tests functionality
Focuses on inputs/outputs [1] [2]

(ii) WHITE BOX TESTING
Any **two** from:
Testing team allowed to see the underlying structure [1]/code/logic
Tester needs to have a detailed knowledge of the development [1]/code
Specific inputs are selected to ensure the appropriate output is produced [1]
Tested using a process called dry run [1] Trace tables used to examine any variable [1]
Finds/logic/dataflow errors [1] [2]

	AVAILABLE MARKS

(b)

Statement	Tick (✓)
System testing is carried out as the individual components are being developed.	
System testing is carried out when all the individual components have been developed.	✓
Integration testing is used to ensure that all units of code are working together as expected.	✓
Integration testing is used to ensure that each individual unit of code is working as expected.	

[2] 6

11 (a) Any **two** from:
to improve the product being developed [1]
to gain feedback from the end user during development/following development [1]
is solution complete[1]
to check if user requirements are met [1] [2]

(b) Any **two** from:
client [1]
(end) user [1]
developer/author of code/programmer [1] [2]

(c) (i) Any **two** from:
can **compare** completed system or design for system to user requirements [1] to ensure all requirements met [1]
use as a checklist [1]

(ii) Any **two** from:
provides evidence of how system performs or works [1]
provides evidence of changes or improvements made [1]
provide evidence that system meets or does not meet the user requirements [1]

(iii) Compare finished product to original plan (e.g. wireframes/algorithms/flowcharts) [1]
to ensure all met before development starts [1] [6] 10

12 (a)

Definition	Search Technique
A search technique which requires a sorted list of items.	Linear Search/**Binary Search** [1]
A search technique which examines every item in a list until the required item is found.	**Linear Search**/Binary Search [1]

	AVAILABLE MARKS

(b) (i) Any **two** from:
less comparisons [1] reduces search list [1]
less passes through the list [1]
finds mid point of the data [1] reference to linear search of all items [1] [2]

(ii) ClassroomTemperatures

22	18	23	17	20

Pass 1

18	22	17	20	23

Pass 2

18	17	20	22	23

Pass 3

17	18	20	22	23

Pass 4

17	18	20	22	23

[4] 8

Total **120**