

PiXL Independence:

Computer Science - Answer Booklet

KS4

Practical Programming

Contents:

- I. Multiple Choice Quiz Answers – 10 credits
- II. Exam Question Answers – 10 credits each

i. Multiple Choice Answers

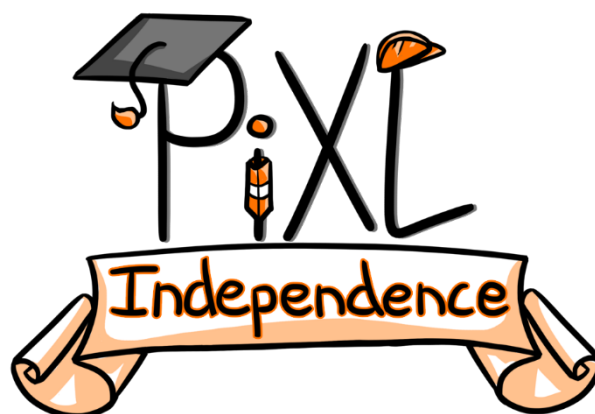
Question	Quiz 1	Quiz 2	Quiz 3	Quiz 4	Quiz 5	Quiz 6	Quiz 7	Quiz 8	Quiz 9	Quiz 10
1	C	D	C	C	E	C	C	D	E	B
2	B	C	B	E	B	B	A	A	C	B
3	C	C	A	B	A	B	C	C	D	B
4	A	B	B	D	A	D	D	D	B	C
5	D	B	D	C	C	B	B	C	B	D
6	B	C	B	D	C	B	A	D	B	E

ii. Exam Question Mark Scheme

1a	[3,9,11,7,2,8,6,17]	1																																				
	[3,9] [11,7] [2,8] [6,17]	1																																				
	[3] [9] [11] [7] [2] [8] [6] [17] List divided into smallest elements	1																																				
	[3,9] [7,11] [2,8] [6,17] First and second item are compared and placed in order in a merged list	1																																				
	[3,7,9,11] [2,6,8,17] Repeated until sorted	1																																				
	[2,3,6,7,8,9,11,17]	1																																				
1b	Compare spear with mace spear is greater so split and take right half	1																																				
	Compare spear with pike spear is greater so split and take right half	1																																				
	Compare spear with spear -Found	1																																				
2 a	pass, name	2																																				
2 b	user, correctLength	2																																				
2 c	True	1																																				
2 d	Local variables are declared and used(accessible) within limited parts of a program, such as a subroutine, procedure or function. [1] Global variables are used(accessible) throughout the entire program.[1]	2																																				
2 e	<ul style="list-style-type: none"> • It is better practice to use local variables whenever a variable is not needed throughout the entire program, as it uses memory only when needed. Alternatively, global variables use memory whenever the program is loaded whether the variable is needed or not. [1] • It is harder to debug errors involving global variables as global variables are liable to be changed within any subroutine, even if this was not initially planned. [1] • Subroutines are easier to use in other programs (as they are re-useable). [1] 	3																																				
3a	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">i</th> <th style="width: 15%;">defence</th> <th style="width: 15%;">attack</th> <th style="width: 15%;">magic</th> <th style="width: 15%;">damage</th> <th style="width: 15%;">health</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">3</td> <td style="text-align: center;">5</td> <td style="text-align: center;">0</td> <td style="text-align: center;">2</td> <td style="text-align: center;">13</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">3</td> <td style="text-align: center;">4</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">12</td> </tr> <tr> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> <td style="text-align: center;">3</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">12</td> </tr> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">12</td> </tr> <tr> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">12</td> </tr> </tbody> </table>	i	defence	attack	magic	damage	health	0	3	5	0	2	13	1	3	4	0	1	12	2	3	3	0	1	12	3	3	2	1	1	12	4	3	2	2	1	12	5
	i	defence	attack	magic	damage	health																																
	0	3	5	0	2	13																																
	1	3	4	0	1	12																																
	2	3	3	0	1	12																																
	3	3	2	1	1	12																																
4	3	2	2	1	12																																	
1 mark per row																																						

3b	<p>Two from:</p> <ul style="list-style-type: none"> • Can reuse subroutines in other programs or parts of the program (only have write once). [1] • Smaller manageable chunks (program broken into). [1] • Improves readability. [1] • Only have to debug once. [1] • Can be called anywhere from program. [1] 	2										
4	<p>Possible solution.</p> <pre> 1 declare work as real # 1 mark-correct datatype 2 declare toys as int # 1 mark-correct datatype 3 declare overtime as real 4 work =input('Enter hours worked') #1 -mark for getting the 3 inputs 5 toys=input('Enter toys made') 6 overtime =('Enter hours overtime') 7 if hours =<8 and hours>5 #1 mark use of selection, 1 mark use of condition 8 pay=hours*0.75 # 1-correct equation #^^^^line above 1mk correct =<8 >5 boolean (and) in condition 9 elif hours>8 # 1 mark use of condition 10 pay=(hours*0.75)+(overtime*toys*1.5)# 1-correct equation 11 else 12 pay=hours*toys*1.5 # 1-correct equation 13 end if </pre>	10										
5a	50 Steve	1										
5b	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Line</th> <th>Program feature</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Data structure</td> </tr> <tr> <td>6</td> <td>Selection</td> </tr> <tr> <td>5</td> <td>Iteration</td> </tr> <tr> <td>3</td> <td>Variable</td> </tr> </tbody> </table> <p>1 mark for each</p>	Line	Program feature	1	Data structure	6	Selection	5	Iteration	3	Variable	4
Line	Program feature											
1	Data structure											
6	Selection											
5	Iteration											
3	Variable											
5c	highScores=[['Johnny',23] ['Paul',19] ['Steve',50] ['Sid',13]] 2D array	1										
5d	<pre> 1 set highScores=[['Johnny',23] ['Paul',19] ['Steve',50] ['Sid',13]] 2 set highscore=0 3 set topgun='' 4 for i=0 to 4 5 if highScores[i,1]>highscore 6 highscore=scores[i,1] 7 topgun=players[i,0] 8 else 9 highscore=highscore 10 topgun=topgun 11 end if 12 next i </pre>	3										
6a	Image will be pixels. [1] Each pixel represented by a binary number. [1]	2										
6b	Colours reduced e.g. 2 similar colours will be represented by one colour. [1] Means less data and therefore smaller file. [1]	2										
6c	Lossless compression. [1] Losing text will make file less readable. [1]	2										
6d	Smaller files mean less storage space. [1] Quicker downloads /internet file transfer. [1]	2										
6e	Sample rate-interval between when sound wave is digitised. [1] Bit depth how many bits represent each sample of the sound wave. [1] Increasing sample rate increases sound quality and file size. [1] Increasing bit depth increases quality and file size. [1]	4										
7a	Comments on code. [1] Meaningful variable names. [1]	2										
7b	E.g. make sure it is text only, make sure it consists of text and numbers	1										
7c	Any four from:	4										

	Auto indentation, colour coding of code, error highlighting, debugging tools, code completion.																
7d	<p>Four from:</p> <ul style="list-style-type: none"> • Compiler creates an executable file. Interpreter does not. [1] • Compiler lists all errors at end of process. [1] • Interpreter gets to first error then stops. [1] • Compiler translates all the code at once. [1] • Interpreter translates code line by line. [1] 	4															
8a	<table border="1"> <thead> <tr> <th>Hexadecimal</th> <th>Denary</th> <th>Binary</th> </tr> </thead> <tbody> <tr> <td>9B₁₆</td> <td>155</td> <td>10011011₂</td> </tr> <tr> <td>C7</td> <td>199₁₀</td> <td>11000111₂</td> </tr> <tr> <td>DD₁₆</td> <td>221</td> <td>11011101</td> </tr> <tr> <td>7A</td> <td>122₁₀</td> <td>01111010₂</td> </tr> </tbody> </table>	Hexadecimal	Denary	Binary	9B ₁₆	155	10011011 ₂	C7	199 ₁₀	11000111 ₂	DD ₁₆	221	11011101	7A	122 ₁₀	01111010 ₂	5
Hexadecimal	Denary	Binary															
9B ₁₆	155	10011011 ₂															
C7	199 ₁₀	11000111 ₂															
DD ₁₆	221	11011101															
7A	122 ₁₀	01111010 ₂															
8b	<pre> 10111011 +01110101 ----- 10011 0000 [1] [1] </pre>	2															
8d	Overflow. Extra bit would have to be stored elsewhere.	1															
8e	01001000 Multiplied by 2	2															
9a	Restrict the format of the input to ensure it is useful.	1															
9b	<p>Two from:</p> <ul style="list-style-type: none"> • Check it's a number [1] • Check it's not negative [1] • Check it's an integer [1] 	2															
9c	<p>One from:</p> <ul style="list-style-type: none"> • Presence check (Not Null) [1] • Pick from a list [1] 	1															
9d	Field is a column heading or category, a particular piece of information e.g. Artist.	1															
	Record is a complete set of data for one entry	1															
9e	Confirming the identity of the user before allowing them to use the program.	1															
9f	<p>Different members of staff might need different levels of access e.g. manager and worker.</p> <p>The database contains sensitive data like bank and personal details. It is important that information cannot be stolen/accessed by unauthorised people.</p>	1 1															
10a	Program works but gives unexpected output.	1															
10b	While username.length>=5 and username.length<=10	1															
10c	Not following the rules of the language e.g. omitting a bracket.	1															
10d	<p>Iterative testing small blocks of code as you go through. [1] Picking up syntax errors and logic errors. [1]</p> <p>Final testing e.g. first alpha and beta. Try to 'break it', followed by testing under realistic conditions [1]. Then does it meet the original brief/success criteria? [1]</p>	4															



Commissioned by The PiXL Club Ltd.

This resource is strictly for the use of member schools for as long as they remain members of The PiXL Club. It may not be copied, sold, or transferred to a third party or used by the school after membership ceases. Until such time it may be freely used within the member school.

All opinions and contributions are those of the authors. The contents of this resource are not connected with, or endorsed by, any other company, organisation or institution.

PiXL Club Ltd endeavour to trace and contact copyright owners. If there are any inadvertent omissions or errors in the acknowledgements or usage, this is unintended and PiXL will remedy these on written notification.