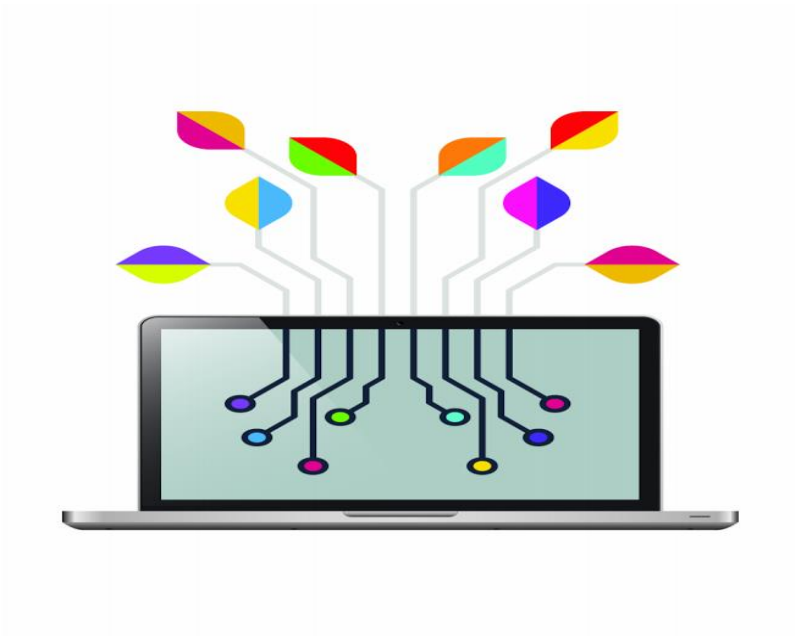
Mark Scheme (Results)

January 2019

BTEC Level 3 National in Computing Unit 1: Principles of Computer Science (31768H)



ALWAYS LEARNING

Pearson

BTEC Qualifications from Pearson

BTEC qualifications from Pearson, the world's leading learning company. We provide a wide range of qualifications including academic, vocational, occupational and specific programmes for employers. For further information visit our qualifications websites at www.btec.co.uk for our BTEC qualifications.

Pearson: helping people progress, everywhere

Our aim is to help everyone progress in their lives through education. We believe in every kind of learning, for all kinds of people, wherever they are in the world. We've been involved in education for over 150 years, and by working across 70 countries, in 100 languages, we have built an international reputation for our commitment to high standards and raising achievement through innovation in education. Find out more about how we can help you and your students at: www.pearson.com/uk

Jan 2019
Publications Code 31768H_1901_MS
All the material in this publication is copyright
© Pearson Education Ltd 2017

General Marking Guidance

- All candidates must receive the same treatment. Examiners must mark the first candidate in exactly the same way as they mark the last.
- Mark schemes should be applied positively. Candidates must be rewarded for what they have shown they can do rather than penalised for omissions.
- Examiners should mark according to the mark scheme not according to their perception of where the grade boundaries may lie.
- All marks on the mark scheme should be used appropriately.
- All marks on the mark scheme are designed to be awarded. Examiners should always award full marks if deserved, i.e. if the answer matches the mark scheme. Examiners should also be prepared to award zero marks if a candidate's response is not worthy of credit according to the mark scheme.
- Where some judgment is required, mark schemes will provide the principles by which marks will be awarded and exemplification may be limited.
- When examiners are in doubt about applying the mark scheme to a candidate's response, the team leader must be consulted.
- Crossed-out work should be marked UNLESS the candidate has replaced it with an alternative response.
- Phonetic spelling should be accepted.

Question Number	Answer	Mark
1a	Award one mark for each point below up to a maximum of 3 marks. • Contains spaces / number is split into sections • Starts with a 0 • Contains brackets /non numerical characters • Will not be used in a calculation	3

Question Number	Answer	Mark
1b	An explanation to contain any two from. • (If stored as a date of birth) it can be compared to the current date (1) • **calculate** their age (date of birth) (1) • age is always accurate/up to date (1) • to apply a correct discount (1) Accept reverse arguments.	2

Question Number	Answer	Mark

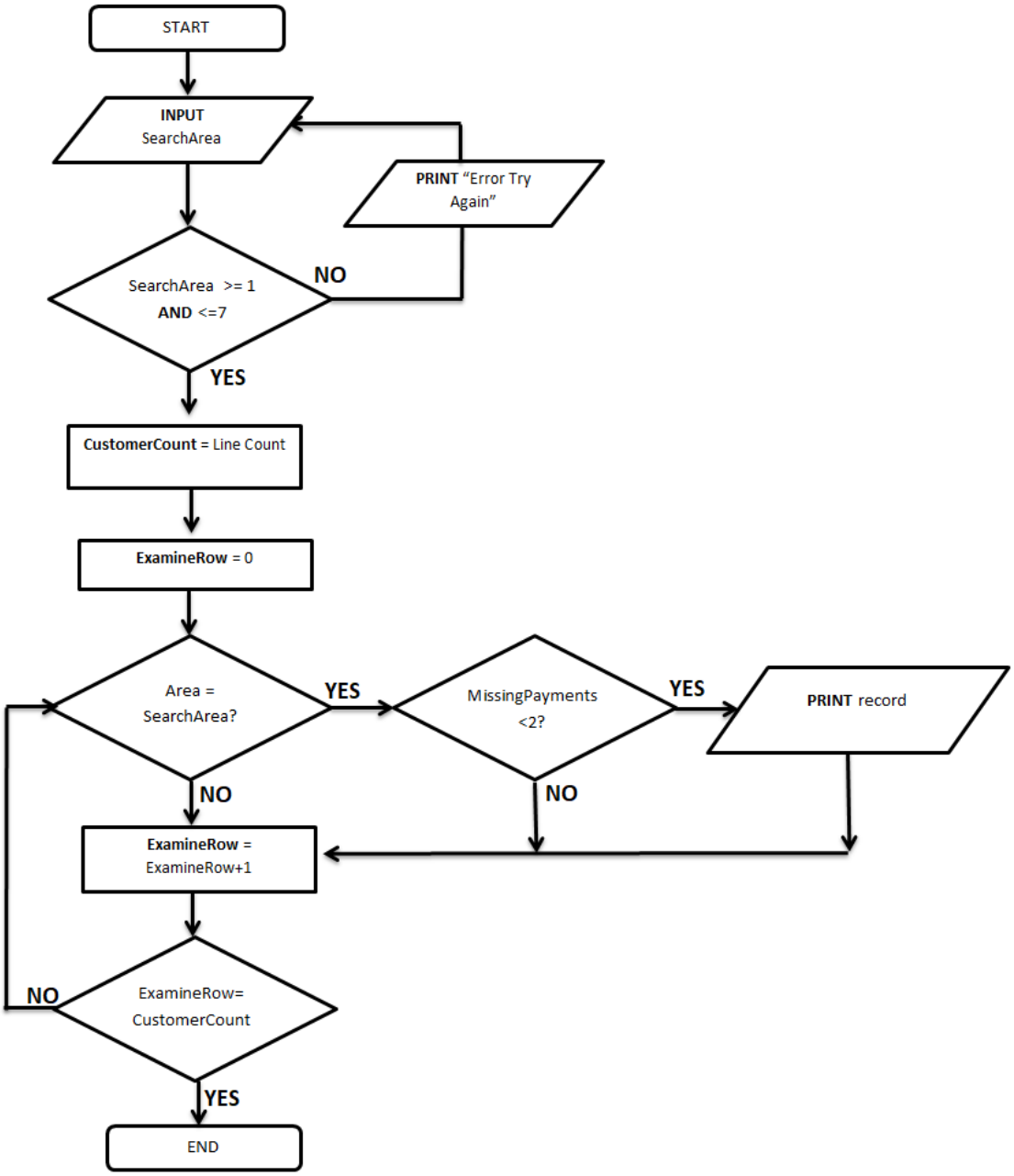
1c	**SOLUTION:**
	BEGIN
	IF missingPayment < 2
	status = "Continue"
	ELSE
	status = "Stop"
	ENDIF
	END
	
	Award 1 mark for:
	• **IF** statement with suitable condition (line 1)
	• correctly setting the status to "continue" (line 2)
	
	Additional Guidance:
	Accept any variable name that implies payments
		2

Question Number	Answer	Mark
1d	A description to containing any four from: • declaring a function create the codes/defines what it will do (1) • (declaring) gives it a name/identifier (1) • (declaring) defines its parameters (1) • calling the function executes the code (1) • (a called function possibly) returns a value (1) • a function is declared once but can be called multiple times (1) **Additional Guidance:** Allow alternatives words for execute e.g. run, apply, perform etc	4

Question Number	Answer	Mark
1e	Award **one** mark for identification and **one** additional mark for appropriate expansion up to 3 marks. Repeated code only need to be written once (1) but can be executed multiples times (1) which increases the performance of the code (1) Functions are self-contained blocks (1) so they can be shared with others (1) reducing development time (1) Functions are self-contained blocks (1) and can be individually tested (1) which makes debugging/maintenance easier (1) Functions can be stored in separate files (1) which can be maintained/tested/debugged separately (1) without affecting the main body of the code (1) Reduces errors in the code (1) as code only needs to be adapted once (1) rather than several times (1)	3

Question Number	Answer	Mark
1f		8

EXAMPLE SOLUTION:



Level	Mark	Descriptor
Level 0	0	No rewardable material.
1	1-3	Structure of the algorithm uses some appropriate hierarchies/subdivision but clarity and/or readability is limited. Variable/object/process names are inappropriate and/or inconsistent Use of logical operations and sequence/structure of processes demonstrate limited accuracy. There is limited use of accepted conventions A partial and/or highly inefficient solution has been achieved.

2	4-6	Structure of the algorithm uses mostly appropriate hierarchies/subdivision to provide some clarity and readability. Variable/object/process names are mostly appropriate but there is some inconsistency Use of logical operations and sequences/structure are mostly accurate with only minor errors. Accepted conventions have been applied but there are some inconsistencies. An almost complete/inefficient solution has been achieved.
3	7-8	Structure of the algorithm uses appropriate and consistent hierarchies/subdivision providing clarity and readability. Variable/object/process names are appropriate and used consistently Use of logical operations and sequences/structures are accurate throughout. Accepted conventions have been used consistently A full and efficient solution been achieved.

Question Number	**Answer**	**Mark**
2a	An explanation to contain **three** from: • A block of code (1) • Separate from the main programme code (1) • Designed to complete a specific task (1) • That will be **called and executed** (1) • Before returning to the main body of code (1) • May not return a value (unlike a function) (1) **Additional Guidance:** Allow Sub Routines / Functions as an alternative for a "block of code."	3

Question Number	Answer	Mark
2b	An explanation to contain any **four** from: • the FOR loop repeats a set number of times / 3 times (1) • the loop will continue if the password entered is incorrect (1) • the break statement will exit the loop (early) (1) • if the correct password is entered (in fewer than three attempts) (1) • the user does not have to enter the password again when correct (1)	4

Question Number	Answer	Mark

2c	**EXAMPLE SOLUTION:**
	
	BEGIN
	INPUT newPassword
	WHILE length of newPassword <8 **THEN**
	  INPUT newPassword
	ENDWHILE
	
	PRINT "Accepted"
	
	END
	
	One mark for each of:
	• Prompts / allows user to enter password (1)
	• Correct check for password length (1)
	• Suitable structure for printing the "accepted" message (1)
	• Correct use of a loop (1)
		4

Question Number	Answer	Mark
2d		5

EXAMPLE SOLUTION (RECURSIVE):

Stage 1:

Unsorted List						
44	75	23	12	55	43	33

Stage 2:

Unsorted Subset A					Unsorted Subset B	
23	12	43	33	44	75	55
				Pivot		

Stage 3:

Unsorted Split Subset A (1)	
23	12
Pivot	

Unsorted Split Subset A (2)	
43	33
Pivot	

Unsorted Subset B	
75	55
Pivot	

Stage 4:

Sorted Split Subset A (1)	
12	23

Sorted Split Subset A (2)	
33	43

Sorted Subset B	
55	75

Stage 5:

Merged Subsets & Final Sorted List						
12	23	33	43	44	55	75

EXAMPLE PARTIAL SOLUTION (INPLACE):

44	75	23	12	55	43	33	44 is greater than 33 so they are swapped over
S>						<E	
33	75	23	12	55	43	44	43 is bigger than 33 so no swap needed
E>					<S		
33	75	23	12	55	43	44	55 is bigger than 33 so no swap needed
E>				<S			
33	75	23	12	55	43	44	12 is smaller than 33 so they are swapped over
E>			<S				
12	75	23	33	55	43	44	75 is bigger than 33 so they are swapped over
	S>		<E				
12	33	23	75	55	43	44	23 is smaller than 33 so they are swapped over
	E>	<S					
12	23	33	75	55	43	44	Pointers meet. 33 is now in the correct place
		<E					
		S>					

	Award one mark for each of the following up to a maximum of 5 marks: **Sorting Process (recursive):** • Identifying 44 as the initial pivot • Placing unsorted numbers **less than** pivot/split point placed on the left • Placing unsorted numbers **greater than** pivot/split point placed on the right • Correctly identifying correct pivots in split subsets • Correctly sorting data in split subsets **Sorting Process (inplace):** • Identify 44 as the initial start pointer • Identify 33 as initial end pointer • Evidence of comparison of initial start and end pointers and swapping values • Correctly swapping start and end **pointers** throughout their solution • Correctly adjusting start and end **pointers** based on sort process **Additional Guidance** - Marks should be awarded for correct sorting. Therefore if the previous step is incorrect (e.g. incorrect pivot is chosen) but the candidate has followed through and applied their error on the next steps correctly then marks should be awarded. If an inplace solution a full sort does not need to be provided. Full marks can be awarded for demonstrating an understanding of the process.	

Question Number	Answer	Mark
2e	A description to contain any **four** from: • make use of 2 loops (1) • loops are nested (1) • inner loop will iterate code a set number of times (1) • to check each adjacent value / swap if needed (1) • outer loop will make sure there is the correct number of passes / until data is sorted (1)	4

Question Number	Answer	Mark
3a	An explanation of the term encapsulation, such as: Joins/wraps data and functions together (1) to form a new entity / self-contained unit (1) that is run using an identifier/name (1) but the contents are kept hidden from outside an object (1)	4

Question Number	Answer	Mark
3b	**Possible responses:** • This is an action / block of code that will be performed / executed after data validation has been completed. • **No Action** - If the validation rules have been met then Theo may decide to take no further action and let the code continue. • **Messages / Dialogue Boxes** – A message be displayed to the user to either confirm successful data entry or to provide advice to the user on how to fix the errors. For example if they have not entered a customer surname or they have entered a date that goes out of bounds. • **Visual Clues** - For example a green tick could appear to confirm successful entry of the time or a red cross to indicate that an error has been made.	8

	• **Audio clues** – For example sharp negative sounds can be played to confirm the time is missing or a happy positive sound can be played to confirm all user inputs are complete. • **Automatic Enforcement** – Code can be setup that will change the format of the data input. For example the date 19/1/2019 could automatically be changed to 19/01/19 or if the surname is entered is lower case the first letter can be capitalised. • **Data Logging** - If an error was found then this could be logged into a separate file to allow the developer to view common faults so they can change the design of the program in future to make it better.	

Level	Mark	Descriptor
Level 0	0	No rewardable material.
1	1-3	Technical vocabulary is used but is not used appropriately to support arguments in relation to the issues of the question. Issues are identified but chains of reasoning are not made leading to a superficial understanding of the relative importance of issues to the context of the scenario. Does not link arguments to the given scenario.
2	4-6	Accurate technical vocabulary is used to support arguments but not all arguments are relevant to the issues of the question. A consideration of relevant issues using logical chains of reasoning but does not reflect upon their relative importance to the context of the scenario. Considers the various elements of the question and but does not always link arguments to the given scenario.
3	7-8	Fluent and accurate technical vocabulary is used to support arguments that are relevant to the issues of the question. A balanced consideration of relevant issues using coherent and logical chains of reasoning that shows a full awareness of their relative importance to the context of the scenario. Carefully considers the various elements of the question and links arguments to the given scenario.

Question Number	Answer	Mark
3c	**Possible responses:** **Sequence:** • Sequence can be used to execute code in order. • Sequence can be used to enter the table booking information from the customer in the order listed in the requirements. • The code could be structured into blocks which each block running a different search (1, 2 or 3) depending on what tables are available. The different blocks could then be executed in sequence. **Conditional:** • Conditional can be used to check for different conditions within the code. • A check can check if a table preference has been selected. If not then this can be set to general tables. • When a table is found the customer can then be asked if they wish to accept. If yes then the details can be stored / an order number generated. If not then the next search can run. • An else statement can be used at the end of the code if no table is found that prints a message. **Iteration:** • Iteration can be used when code needs to loop. • Within each search (1, 2, and 3) code can be iterated to check the availability of each table being searched. For example if window tables are selected then code be used to check the availability of table 1 and then the same code could loop to check the availability of table 2, 5 and 6. • Code could be used on the booking form to continually iterate code to check if the data required to search for a table has been entered before running code to find available tables.	10

Level	Mark	Description
0	0	
1	1-4	Technical vocabulary is used but is not used appropriately to support arguments in relation to the issues of the question.

		Issues are identified but chains of reasoning are not made leading to a superficial understanding of the relative importance of issues to the context of the scenario.
2	5-7	Accurate technical vocabulary is used to support arguments but not all arguments are relevant to the issues of the question. A consideration of relevant issues using logical chains of reasoning but does not reflect upon their relative importance to the context of the scenario.
3	8-10	Fluent and accurate technical vocabulary is used to support arguments that are relevant to the issues of the question. A balanced and wide ranging consideration of relevant issues using coherent and logical chains of reasoning that shows a full awareness of their relative importance to the context of the scenario.

Question Number	Answer	Mark
4a	**Possible responses:** **General:** • The program will make use of a user interface and therefore the program will respond to user interacts. • Code within event driven languages is split into blocks / functions and a main loop can constantly listen for different events and instantly trigger the correct handler (to run a particular block / function) at any time. • Procedural languages work best then there are longer pieces of code/processes and are not suited to running lots of small independent tasks (e.g. that is required of a user interface). • Object orientated languages are well suited when multiple programmers are working on the same project but when each section does not rely on another section. In this case each section of code will rely on other sections which event driven languages can handle. • Object orientated languages generally require more code to be written for small projects such as this one. They are better suited to projects that have lots of repeated code but this project will not have lots of repeated code and therefore it will require lots more code to be written. **Events:** • Events could be setup for each button on the user interface. • A mouse / get focus event could be setup for when the user clicks on the individual buttons with the mouse. • A keyboard event could be setup to allow the user to navigate the interface with the tab key or assigned keyboard shortcuts. • A hover event could be setup to display messages when the user moves the mouse cursor over individual items on the user interface. **Event Handlers:** • These will be assigned to the events and will run when an event takes place. • For example when Poppy clicks on button 1, the handler will take the user to a different form/interface that allows them to enter their subjects. • For example when the user clicks on button 4, a dialogue box will appear asking the user which	6

	day they would like to print a revision timetable for. **Trigger Functions:** • These will be used to decide what code / functions to run. • When an event takes place on the user interface, the trigger function will therefore decide which handler to run and then execute that specific code associated to that handler. **Service Orientated Processing:** • These are processes that will run in the background. There will not really be a lot of use for this as the program is very small and will not be performing lots of tasks at once. • She could make use of service orientated processing by keeping the program open while she is revising. This will then keep a log of the number of minutes she is revising. **Time Driven:** • A time driven function can be used to remind the user of the time every x minutes during a revision period so they do not lose track of time. • A time driven function can be used to be able to remind the user when they need to start and stop revising a particular subject. • The user could enter the time when they start revising and a time driven function can be used to remind them when it's time for them to take a break.	

Level	Mark	Description
0	0	
1	1-2	Technical vocabulary is used but is not used appropriately to support arguments in relation to the issues of the question. Issues are identified but chains of reasoning are not made leading to a superficial understanding of the relative importance of issues to the context of the scenario.
2	3-4	Accurate technical vocabulary is used to support arguments but not all arguments are relevant to the issues of the question. A consideration of relevant issues using logical chains of reasoning but does not reflect upon their relative importance to the context of the scenario.
3	5-6	Fluent and accurate technical vocabulary is used to support arguments that are relevant to the issues of the question.

		A balanced and wide ranging consideration of relevant issues using coherent and logical chains of reasoning that shows a full awareness of their relative importance to the context of the scenario.

Question Number	Answer	Mark
4b	**Possible responses:** **General:** • Abstraction involves identifying the information that is needed to solve a problem and filtering out information that is not needed. **Identifying the general themes:** • Poppy could have used abstraction to identify the general overarching themes/areas. • For example: 1. Subjects (that needs to be revised) 2. Days (when she wants to revise) 3. Time (in each day she wants to revise) 4. Revision time length 5. Rest breaks • This will allow Poppy to focus on the key/essential structures of the program without getting 'side tracked' by the complexities of the problem. **Pattern generalisation:** • When the themes have been defined, Poppy can use pattern generalisation to look for patterns, similarities and connections between them. • For example: 1. An array has been setup for each day of the week which contains a list of subjects. Therefore the number of subjects in each array can be automatically counted rather than being manually entered in section D. 2. The revision time is calculated in section E. The break time is calculated in section E. However both of these values are required to calculate the revision end time. **Efficiency of code:** • Poppy's algorithm is very efficient and reuses values and calculations from earlier in the code. For example: 1. The numberofSubjects variable calculated in section C is then reused in section D. 2. The revisionTime variable which is calculated in section E is then reused in section F.	8

	• This will mean the program code will perform better and run on computers that have a lower specification / fewer resources. • Removing unnecessary elements will also speed up development time. As there is likely to be less code this will decrease the amount of debugging that is required. Less code will also make maintaining the code easier in the future.	

Level	Mark	Description
0	0	
1	1-4	Technical vocabulary is used but is not used appropriately to support arguments in relation to the issues of the question. Issues are identified but chains of reasoning are not made leading to a superficial understanding of the relative importance of issues to the context of the scenario.
2	5-6	Accurate technical vocabulary is used to support arguments but not all arguments are relevant to the issues of the question. A consideration of relevant issues using logical chains of reasoning but does not reflect upon their relative importance to the context of the scenario.
3	7-8	Fluent and accurate technical vocabulary is used to support arguments that are relevant to the issues of the question. A balanced and wide ranging consideration of relevant issues using coherent and logical chains of reasoning that shows a full awareness of their relative importance to the context of the scenario.

Question Number	Answer	Mark
4c	**Possible responses:** • **Structure** – The algorithm is generally well structured and follows standard programming conventions that should be able to be easily implemented in any programming language. • There is good use of indentation throughout the algorithm so it's easier to see which statement belongs to which condition. • **Comments** - The use of comments makes it easier to see the purpose of each block of code so it's easier to cross reference this to the requirements. • Further comments may have added extra clarity within each section. • **Blocks** – The use of blocks makes the algorithm easier to navigate. Each block stands as a self-contained unit which will help if debugging is required. • **Variables** –Sensible variables names have been used and have been used consistency throughout the algorithm which will help when the actual code is produced. • Standard naming conventions have been used (e.g. camelCase) which are standard practice and generally universally understood. • Some variable names are particularly long which may increase errors, especially the ones that appear several times in the code. • **Efficiency –** The algorithm is generally very efficient and uses iteration to repeat code to reduce the amount of code needed. • **Errors:** • Section D - An incorrect operator is. A divide operator has been used and it should be a multiply operator. • Section E contains an infinite while loop. This is because it is set to repeat the code until revisionTime is no longer greater than 45. However this variable is not changed throughout the code and therefore the loop will continue to loop forever. • Section G - The revisionTime variable is correctly calculated in section D. However the same variable is used in section E to calculate the number of rest breaks. In order to calculate this, the variable is changed. Therefore when this variable is printed in section G, an inaccurate value is shown to the user.	12

	• Section G does not show the number of rest breaks needed. • **Overall** the algorithm uses standard programming conventions throughout so it will be readable for any programmer. • There are some errors with the logic which will therefore create some inaccuracies within the data processed. • This could therefore increase debugging time.	

Level	Mark	Descriptor
Level 0	0	No rewardable material.
1	1-4	Technical vocabulary is used but is not used appropriately to support arguments in relation to the issues of the question. Issues are identified but chains of reasoning are not made leading to a superficial understanding of the relative importance of issues to the context of the scenario. No conclusion is presented or is generic.
2	5-8	Accurate technical vocabulary is used to support arguments but not all arguments are relevant to the issues of the question. A consideration of relevant issues using logical chains of reasoning but does not reflect upon their relative importance to the context of the scenario. An attempt at a conclusion is presented that links arguments to the given scenario but is not justified in that it does not reflect the careful consideration of both sides of the argument.
3	9-12	A balanced explanation of both sides of the argument is presented using fluent and accurate technical vocabulary. The points made are discussed in a balanced way that reflects their relative importance to the given scenario. A clear and justified conclusion is presented that reflects a thorough consideration of both sides of the argument. Fluent and accurate technical vocabulary are used to support arguments that are relevant to the issues of the question. A balanced and wide ranging consideration of relevant issues using coherent and logical chains of reasoning that shows a full awareness of their relative importance to the context of the scenario. A fully justified conclusion is presented that links arguments to the given scenario and that reflects the careful consideration of both sides of the argument leading to a reasoned decision.

For more information on Edexcel qualifications, please visit our website
www.edexcel.com

Pearson Education Limited. Registered company number 872828
with its registered office at Edinburgh Gate, Harlow, Essex CM20 2JE