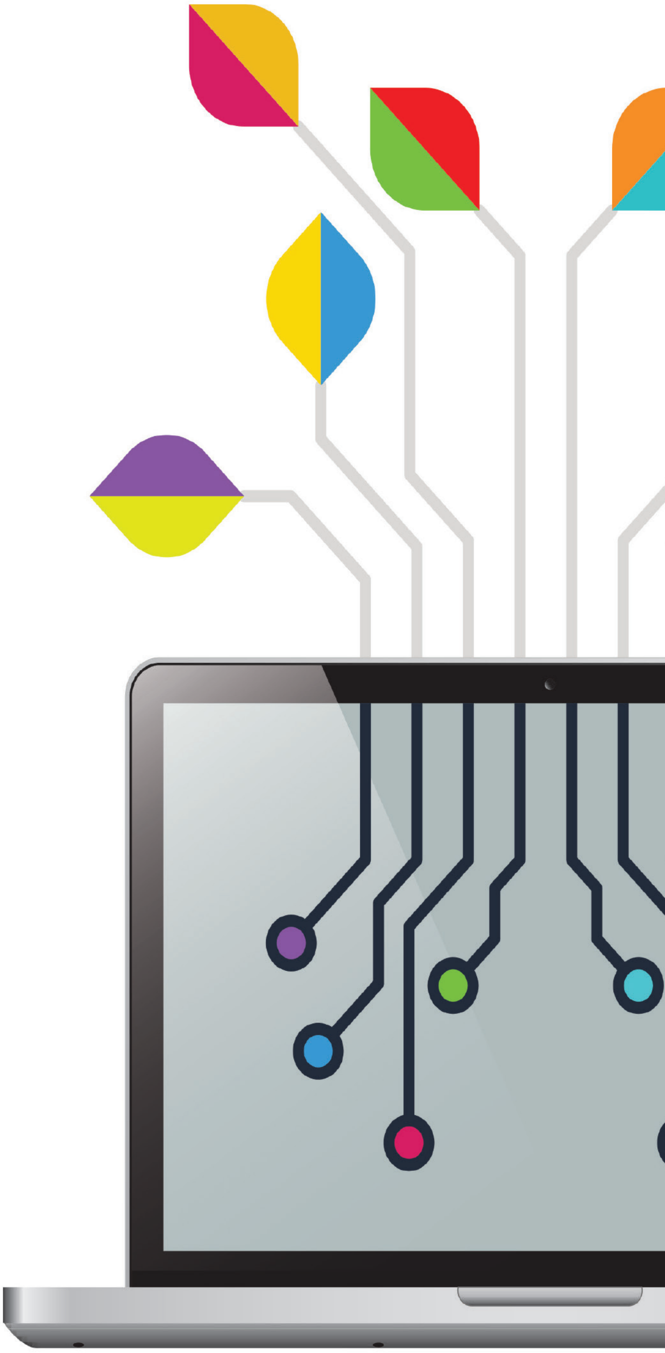
Pearson BTEC Level 3 National in Computing

Unit 1: Principles of Computer Science



Sample Assessment Materials (SAMs)

For use with:

- *Extended Certificate, Foundation Diploma and Extended Diploma in Computing*
- *Diploma in Computing*
- *Diploma in Creative Computing*
- *Diploma in Computer Science*
- *Diploma in Computer Systems and Network Support*
- *Diploma in Business Information Systems*

First teaching from September 2016 Issue 2

Edexcel, BTEC and LCCI qualifications

Edexcel, BTEC and LCCI qualifications are awarded by Pearson, the UK’s largest awarding body offering academic and vocational qualifications that are globally recognised and benchmarked. For further information, please visit our qualifications website at qualifications.pearson.com. Alternatively, you can get in touch with us using the details on our contact us page at qualifications.pearson.com/contactus

About Pearson

Pearson is the world's leading learning company, with 35,000 employees in more than 70 countries working to help people of all ages to make measurable progress in their lives through learning. We put the learner at the centre of everything we do, because wherever learning flourishes, so do people. Find out more about how we can help you and your learners at qualifications.pearson.com

References to third-party material made in this specification are made in good faith. We do not endorse, approve or accept responsibility for the content of materials, which may be subject to change, or any opinions expressed therein. (Material may include textbooks, journals, magazines and other publications and websites.)

Contents

Summary of Pearson BTEC Level 3 Nationals in Computing Sample Assessment Materials for Unit 1: Principles of Computer Science Issue 1 to 2 changes

Summary of changes made between previous issues and this current issue	Page numbers
This Sample Assessment Material has been refocussed to remove reference to particular programming languages:	
Question paper Question 4 (a) Reference to Guvinder's code replaced with Guvinder's pseudocode.	17
Information booklet Reference to code replaced with pseudocode.	25
Mark scheme Question 4 (a) Reference to the sleep function when 'calculating average score' replaced with WAIT.	40

BTEC

Pearson BTEC Level 3 Nationals

Write your name here

Surname	Forename

Learner Registration Number □□□□□□□ Centre Number □□□□□□ Level 3

Computing

Unit 1: Principles of Computer Science

Extended Certificate /Foundation Diploma /Diploma /Extended Diploma in Computing

Diplomas in Computing for Creative Industries, Computer Science, Computer Systems and Network Support, Business Information Systems

Sample assessment material for first teaching September 2016

Time: 2 hours

You must have:
Insert (enclosed)

Total marks

Instructions

- Use **black** ink or ball-point pen.
- **Fill in the boxes** at the top of this page with your name, centre number and learner registration number.
- Answer **all** questions.
- Answer the questions in the spaces provided – *there may be more space than you need.*

Information

- The total mark for this paper is 90.
- The marks for **each** question are shown in grey boxes – *use this as a guide as to how much time to spend on each question.*
- Additional information and stimulus material needed to answer the questions can be found in the additional information booklet.
- You may use a calculator

Advice

- Read each question carefully before you start to answer it.
- Try to answer every question.
- Check your answers if you have time at the end.

Turn over ▶

Paper reference
XXXX/XX
S50696A

1/1/1/1/1

S 5 0 6 9 6 A 0 1 2 0

PEARSON

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

BLANK PAGE

Please refer to the Information Booklet in order to answer Question 1.

Reyha is creating a 2D maze-based computer game.

A design for the Level 1 screen and the Level 1 design criteria are given in Section 1 of the Information Booklet.

1 Lives and score will be given as variables.

(a) Identify **three** other features of the game proposal that would be represented as a variable.

3 marks

1 ....

2 ....

3 ....

The position of sprites onscreen will be set using X and Y co-ordinates.

The player will use the arrow keys to control sprite A.

(b) Produce pseudocode that describes the movement of sprite A when the user presses a key to move it to the right.

3 marks

....

....

....

....

....

....

....

....

....

During play, the player moves sprite A to the door with 3 lives and 13 seconds of time left.

(c) Calculate the points the player will be awarded.

You are advised to show your working.

2 marks

Answer ....

Reyha has written some pseudocode to show how sprite B will be controlled in Level 1.

```
BEGIN
MOVE upwards until touching wall
WHEN touching wall
MOVE downwards
END
```

Check the code against the Level 1 design criteria.

(d) Explain **three** improvements that can be made to the pseudocode to better meet the criteria when programming sprite B.

6 marks

1 ....

....

....

....

2 ....

....

....

....

3 ....

....

....

....

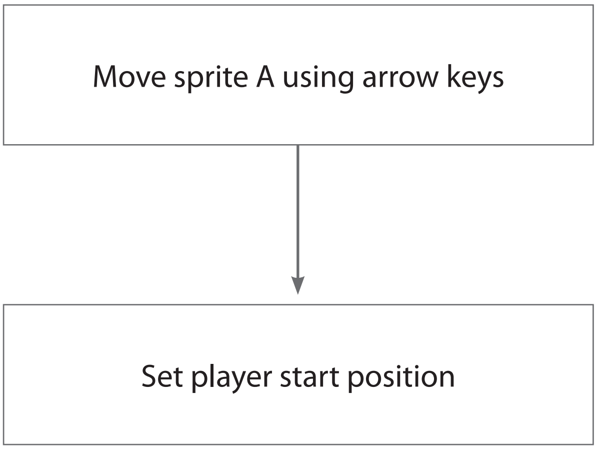
Programmers can use flow charts to plan the logic for their programs.

Reyha’s flow chart must show the logic for:

- actions when sprite A touches an enemy sprite
- checking when the game has ended
- setting the level completion criteria
- checking if the level completion criteria have been met.

(e) Complete the flow chart to show the logic for actions for this screen.

8 marks



Total for Question 1 = 22 marks

Tony has been employed to write a computer program that will store and process information for a small business.

The program will store personal and employment details, including name, date of birth, employee number and pay details.

Tony writes the program using an object-oriented language.

2 (a) Explain how the structure of an object-oriented language would be used to manage the data in the program.

4 marks

....

....

....

....

....

....

....

....

....

....

Tony creates a section of code to calculate how much money should be deducted from a staff member's weekly pay to be paid into the pension scheme.

The table shows an example of data that will be processed.

Name	Maria Cortez
Week beginning	07/09/2016
Hours worked	37
In pension scheme	Yes

Validation checks are used to ensure that data is suitable and to reduce data entry errors.

(b) Explain how a different validation check would be used for each field.

6 marks

Week beginning

....

....

Hours worked

....

....

In pension scheme

....

....

Tony needs to create code to calculate staff annual holiday entitlement.
Holiday entitlement is calculated using the following criteria.

Full-time staff working five days a week are entitled to 28 days' holiday per year.

Part-time staff are entitled to a proportion of 28 days' holiday per year, based on the number of days they work each week.

(c) Develop a function, using pseudocode, that calculates the holiday entitlement for full-time and part-time staff.

4 marks

....

....

....

....

....

....

....

....

(d) Analyse how the features of object-oriented languages will be of benefit to Tony and the company when developing the program.

6 marks

....

....

....

....

....

....

....

....

....

....

....

Total for Question 2 = 20 marks

Megan organises a charity run in a local park.

She needs a computer program to process the runners' data.

Runners can start at any time between 09:00 and 14:00.

Start and finish times will be recorded. The fastest three runners will each receive a prize.

The table shows the data for ten runners.

Runner number	Name	Start time	Finish time	Age	Running club member
1	Jones T	09:30	10:10	32	No
2	Smith J	10:00	10:25	24	Yes
3	Irwin S	09:10	10:00	56	No
4	Patel P	11:00	11:45	52	No
5	Murphy M	10:40	11:40	76	Yes
6	Owen K	12:10	12:40	19	No
7	Drew L	12:55	13:50	45	No
8	Abdi N	10:50	11:18	22	Yes
9	Stein V	10:15	10:50	39	No
10	Tailor B	09:39	10:05	17	Yes

The data for each runner will be stored as a record.

3 (a) Explain how two characteristics of a record make it a suitable structure to store this data.

4 marks

....

....

....

....

....

....

....

....

....

....

Megan writes a bubble sort algorithm to find the three youngest runners.
She stores this test data in an array:

Age = [32 24 56 52 76 19 17]

(b) Demonstrate how the computer processes the age array during the first pass by completing the table.

6 marks

	1	2	3	4	5	6	Comments
Original array	32	24	56	76	19	17	
End of first pass							

Megan wants to know how many runners are members of a running club.

(c) Produce an algorithm that counts the number of running club members and outputs the result at the end of the process.

4 marks

Megan writes a program to process runner data in an event-driven language.

(d) Assess how the features of event-driven languages are suitable for Megan's needs.

8 marks

Total for Question 3 = 22 marks

Please refer to the Information Booklet in order to answer Question 4.

Guvinder is developing a primary school computer game to help children improve their mathematical skills.

The first level of the game will develop addition skills.

4 (a) Assess the efficiency and effectiveness of Guvinder's pseudocode.

8 marks

Guvinder needs to test his code.

(b) Analyse how he could use compatibility testing to ensure the program works as planned.

6 marks

Guvinder wants to include the computer game in a web application.

(c) Evaluate the implications of Guvinder's decision to implement the computer game in this way.

12 marks

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

....

....

....

....

....

....

....

....

....

....

....

....

....

....

Total for Question 4 = 26 marks

END OF EXAM

TOTAL FOR PAPER = 90 MARKS

Pearson BTEC Level 3 Nationals

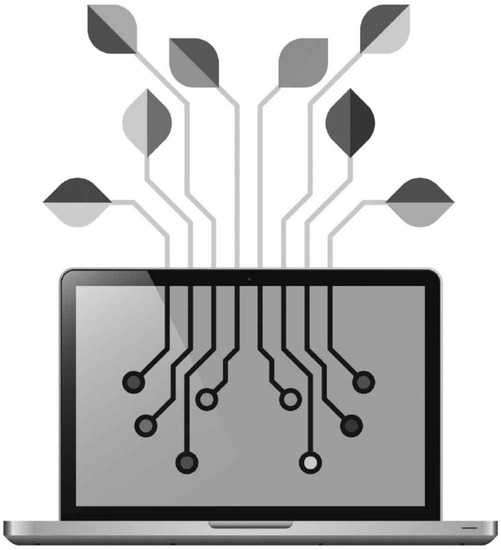
Computing

Information Booklet for Unit 1: Principles of Computer Science

Extended Certificate /Foundation Diploma /Diploma /Extended Diploma in Computing
Diplomas in Computing for Creative Industries, Computer Science, Computer Systems and Network Support, Business Information Systems
Sample assessment material for first teaching September 2016
Time: 2 hours

Instructions:

- You will need the information in this booklet to answer some questions.
- Read the information carefully.
- You must **not** write your answers in this booklet.
- Only your answers given on the question paper booklet will be marked.



Paper reference
XXXX/XX
S50696A

1/1/1/1/1

PEARSON

SECTION 1

The information in this section should be used to answer Question 1.

Figure 1 shows a design for the Level 1 screen from the game.

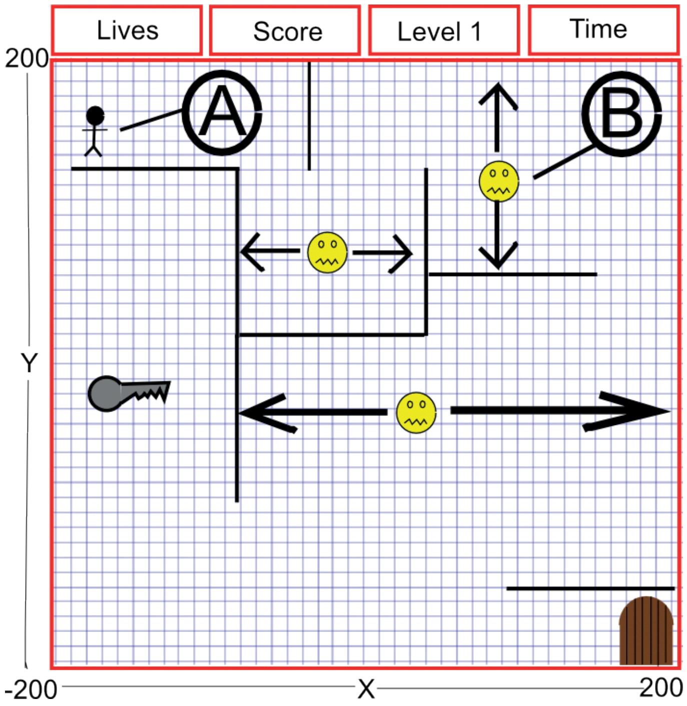


Figure 1

All movable characters and enemies will be referred to as ***sprites***.

The screen for level 1 of the game must meet the following design criteria:

- positions on screen will be set using X and Y co-ordinates
- the player controls the character (sprite A) using the arrow keys
- sprite A will not be allowed to move through walls
- the arrow keys will be assigned to control Sprite A's movement
- sprite A will only move up, down, left or right
- sprite A will move 10 units in a given direction for each key press
- the player starts the game with 3 lives
- a life is lost when sprite A touches an enemy sprite
- after touching an enemy sprite A returns to the "start"
- the enemy sprites will move along given paths
- the enemy sprites will repeat the path until the level is completed
- the player must collect the key and reach the door to move to the next level
- the player will be given 30 seconds to complete this level
- reaching the door will award the player points equal to the number of seconds left multiplied by the number of lives left. Then 50 points are added.
- completing the level with all three lives will award bonus points that will multiply the points awarded for the level by 3.5
- when the timer reaches 0 the player will lose a life and be sent back to the start.

SECTION 2 – Pseudocode

The information in this section should be used to answer Question 1.

The following list of terms are not listed in the unit specification. They will be used in a/some questions that refer to pseudocode in this examination:

MOVE – command to control an onscreen character. This would not determine direction or distance.

SECTION 3 – Programming Code

The following list of programming terms and functions are not listed in the unit specification. They will be used in a/some questions that refer to programming in this examination:

time – a python library that allows programs to use features relating to time and timings

sleep() – a function from the time library that allows the programmer to add a delay before executing more code.

SECTION 4

The information in this section should be used to answer Question 4.

The first level of the game will develop children's skills in addition.

Initial requirements for first level of the maths game are:

- the program will choose two numbers at random
- the random numbers will be between 0 and 9 inclusive
- the two numbers will be shown on the screen
- the user will type in their answer
- the user will get 1 point for every answer they get right
- the level will contain three rounds of questions
- they will answer 10 questions in each round
- the total for each round will be stored
- at the end of the third round, the user's highest score and their average score will be displayed onscreen.

Section 4 (continued)

This is the pseudocode that Guvinder has written for the first level.

```
1 roundlScore = 0
2 round2Score = 0
3 round3Score = 0
4 roundNumber = 1
5 GOBAL roundNumber
6 GLOBAL round1Score
7 IF roundNumber = 1
8     OUTPUT("Get ready for round 1...")
9     questionNumber = 1
10         WHILE questionNumber <=16
11               numl = RANDOM NUMBER 6 TO 9
12         num2 = RANDOM NUMBER 6 TO 9
13               answer = numl + num2
14               OUTPUT "Round Number 1. Question number " questionNumber
15               OUTPUT("What is " numl "+" num2
16               INPUT guess
17               IF guess = answer
18                     round1lScore = roundlScore +1
19                     OUTPUT("Well done. Your score is now " roundl1Score
20                     questionNumber = questionNumber +1
30               ELSE
31                     OUTPUT("That was not correct. Your score is” roundlScore
32                     questionNumber = questionNumber +1
33         roundNumber = roundNumber +1
34 IF roundNumber = 2
35    OUTPUT("Get ready for round 2...")
36    WAIT 3 seconds
37    GLOBAL round2Score
38    questionNumber = 1
39    WHILE questionNumber <=10
40         numl = RANDOM NUMBER 6 TO 9
41         num2 = RANDOM NUMBER 6 TO 9
42         answer = numl + num2
43         OUTPUT "Round Number 2. Question number2 questionNumber
44         OUTPUT "What is " numl) + " num2
45         INPUT guess
46         IF guess = answer
47               round2Score = round2Score +1
48               OUTPUT "Well done. Your score is now ‟ round2Score
49               questionNumber = questionNumber +1
50         ELSE
51               OUTPUT "That was not correct. Your score is " round2Score
52               questionNumber = questionNumber +1
53    roundNumber = roundNumber +1
54 OUTPUT "Round 1 score = " round1lScore  " Round 2 score ‟= round2Score
55 IF roundNumber = 3
56    OUTPUT "Get ready for round 3..."
57    WAIT 3 seconds
58 GLOBAL round3Score
59 questionNumber = 1
60    WHILE questionNumber <=10
61         numl = RANDOM NUMBER 6 TO 9
62         numl = RANDOM NUMBER 6 TO 9
63         answer = numl + num2
64         OUTPUT("Round Number 3. Question number " questionNumber
65         OUTPUT("What is " numl + num2
66         INPUT guess
67               IF guess = answer
68               round3Score = round3Score +1
69               OUTPUT(‟Well done. Your score is now " round3Score
70               questionNumber = questionNumber +1
71         ELSE
72               OUTPUT("That was not correct. Your score is " round3Score
73               questionNumber = questionNumber +1
74 roundNumber = roundNumber+1
75 OUTPUT "Round 1 score = ‟ round1Scor) " Round 2 score =‟(round2Score + "
   Round 3 score = ‟ round3score
76 OUTPUT"Calculating your average score...2
77 WAIT 1 second
78 OUTPUT "..."
79 WAIT 1 second
80 OUTPUT "..."
81 WAIT 1 second
82 OUTPUT "..."
83 averageScore = (roundl1Score + round2Score + round3Score) /3
```

Unit 1: Principles of Computer Science – sample mark scheme

General marking guidance

- All learners must receive the same treatment. Examiners must mark the first learner in exactly the same way as they mark the last.
- Mark schemes should be applied positively. Learners must be rewarded for what they have shown they can do, rather than be 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 the 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 the learner's response is not worthy of credit, according to the mark scheme.
- Where some judgement is required, mark schemes will provide the principles by which marks will be awarded and exemplification may be limited.
- When examiners are in doubt regarding the application of the mark scheme to a learner's response, the team leader must be consulted.
- Crossed-out work should be marked UNLESS the learner has replaced it with an alternative response.

Specific marking guidance for levels-based mark schemes*

Levels-based mark schemes (LBMS) have been designed to assess learners' work holistically. They consist of two parts: indicative content, and levels based descriptors. Indicative content reflects specific content-related points that a learner might make. Levels-based descriptors articulate the skills that a learner is likely to demonstrate, in relation to the assessment outcomes being targeted by the question. Different rows in the levels, represent the progression of these skills.

When using a levels-based mark scheme, the 'best fit' approach should be used.

- Examiners should first make a holistic judgement on which band most closely matches the learner's response and place it within that band. Learners will be placed in the band that best describes their answer.
- The mark awarded within the band will be decided based on the quality of the answer in response to the assessment focus/objective and will be modified according to how securely all bullet points are displayed at that band.
- Marks will be awarded towards the top or bottom of that band depending on how they have evidenced each of the descriptor bullet points.

Question number	Answer	Mark
1(a)	Award **1** mark for any of the following up to a maximum of **3** marks: - level - time - key collected - position of characters/enemies/sprites.	**(3)**

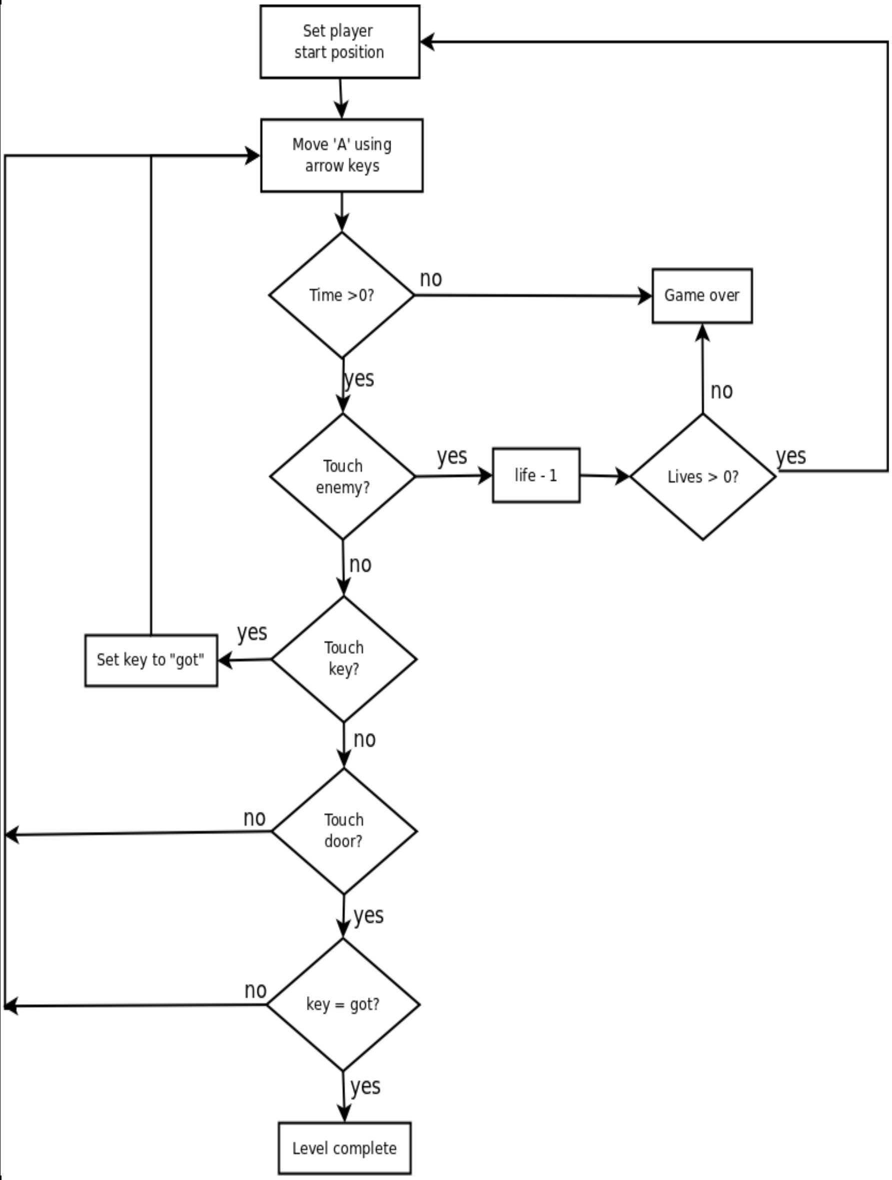
Question number	Answer	Mark

1(b)	**Example answer**
	BEGIN
	IF NOT touching wall
	        When right arrow pressed
	        Change X by (+)10
	END
		(3)

	Award **1** mark for each related descriptive point up to a maximum of **3** marks. - Correct logic for sprite/character A not touching the wall. - Input identified as right arrow key. - X position is changed by a positive value of 10. Accept any other relevant phrasing/wording. Additional guidance: Mark for the logic may be awarded for any correct logical solution, e.g. to not move if touching the wall, else execute the move. Learner response may or may not have begin/end.	

Question number	Answer	Mark
1(c)	Award **2** marks for correct answer only. Award **1** mark only for showing an awareness of correct method (bracketed aspects of method). Method: ((13*3)+50)*3.5 Answer: 311.5	**(2)**

Question number	Answer	Mark
1(d)	Award **1** mark for identification and **1** additional mark for appropriate expansion up to a maximum of **2** marks each. • Setting a condition to detect the bottom wall (1), to stop enemy sprite B moving downwards (1). • The code should contain a loop (1), so that the enemy sprite B will repeat the action until the level is complete (1). • Include nested logic (1) to identify which wall is being touched (1). • Indent subordinate actions (1) to ensure correct action/reaction to events (1). • Replace upwards/downwards with 'change y' (1) to specify how many units it should move (in each loop of the instruction) (1). Accept any other relevant phrasing/wording.	**(6)**

Question number	Answer	Mark
1(e)	Example flow chart response:  Set player start position Move 'A' using arrow keys Time >0? no Game over yes Touch enemy? yes life - 1 Lives > 0? no yes no Touch key? yes Set key to "got" no Touch door? no yes key = got? no yes Level complete	**(8)**

Question number	Answer	Mark
1(e) (cont)	Award **1** mark for any of the following up to a maximum of **6** marks: • correct logic for checking time left (time >0 yes, continue to other actions/no, go to game over) • correct logic for touching an enemy sprite (no, continue/yes, flows to lose life) • correct logic for loose life (e.g. life = life -1, life -1, life -=1 etc.). • correct logic for check number of lives (life>0. If 0 end game if>0 loop to set player position) • correct logic for 'touching key' (no, continue/yes, set 'have key' to yes and loop back to moving around maze) • correct logic for 'touching door' (no, loop back to move around maze/yes, check if have key = yes) • correct logic for 'have key' (no, loop back to move around maze/yes, level complete). Award **1** mark for each of the following up to a maximum of **2** marks: • correct use of BCS flow chart symbols • flow chart is logically organised allowing accurate representation of the flow of data/actions. Accept any other relevant phrasing/wording.	

Question number	Answer	Mark
2(a)	Award **1** mark for each related descriptive point up to a maximum of **4** marks. • A class (1) would be used to group all the related staff data (name, DOB etc.) (1). • An object (1) would be generated for each staff member, using the class as a template (1). Accept any other relevant phrasing/wording.	**(4)**

<table>
<tr><th>Question number</th><th>Answer</th><th>Mark</th></tr>
<tr><td>2(b)</td><td>Award <b>1</b> mark for identification and <b>1</b> additional mark for appropriate expansion, up to a maximum of <b>2</b> marks each.
<table>
<tr><th>Field</th><th>Validation check techniques</th></tr>
<tr><td>Week beginning</td><td>Data type check (1) ensures that the data entered is in a suitable/valid date format (1).
Constraints (1) check that the date entered is a Monday (start of week)/date is not in the future (1).</td></tr>
<tr><td>Hours worked</td><td>Data type check (1) ensures data entered is numeric (1).
Constraints (1) check that it is entered as a two-digit number (1).
Range (1) check number fits in a predetermined range, e.g. the minimum/maximum hour allowed (1).</td></tr>
<tr><td>In pension scheme</td><td>Boolean (1) data has only two possible values yes/no (1).</td></tr>
</table>
Accept any other relevant phrasing/wording.
Additional guidance:
<ul>
<li>allow only one example for each type of validation check, e.g. allow only one of length check, presence check for constraint.</li>
<li>award each validation check only once – learners should provide a different check for each example.</li>
</ul></td><td><b>(6)</b></td></tr>
</table>

Question number	Answer	Mark

2(c)	**Example answer**
	BEGIN
	IF FullTime == "True":
	    Holidays = 28
	ELSE:
	    Holidays = 28 / 5 * (days per week)
	END
		(4)

	Award **1** mark for each related descriptive point up to a maximum of **4** marks. • Correct logic for checking if staff are full-time or part-time (e.g. FullTime == True) (1). • Appropriate variable name declared for ‘holidays’ and consistently used throughout the code (1). • For full-time staff – variable for holidays set to 28 (1). • For part-time staff: correct formula holidays = 28 / 5 *(days per week) (1). Accept any other relevant phrasing/wording. Additional guidance: • the response should show a clear understanding between comparison (e.g. FullTime == (is equal to)) and setting a value (e.g. Holidays = 28) this may be through clearly written instructions in the pseudocode or by using common programming conventions, i.e. 1 equals sign for assign and 2 or 3 for ’is equal to’ • credit alternative solutions that use correct logic and would produce the expected outcome.	

Question number	Indicative content
2(d)	Answers will be credited according to the learner's demonstration of knowledge and understanding of the material, using the indicative content and levels descriptors below. The indicative content that follows is not prescriptive. Answers may cover some/all of the indicative content but should be rewarded for other relevant answers. **General** • Modular structure makes it easier to test and maintain. • Easy to modify and use existing code as new objects can be created with small differences to existing ones. **Data hiding** • Data declared as 'private' inside a class is hidden from all other classes. • This protects the data and improves the reliability of the program and the security of the data by ensuring it isn't accessed and/or modified by other parts of the program. **Encapsulation** • Data and functions are grouped together, which aids maintenance and development by ensuring associated functions and processes are together so a programmer does not have to search for related parts of the code when changes need to be made. • Encapsulation also allows the programmer separate levels of abstraction in one group. **Inheritance** • Properties defined by an existing class can be passed to another class, in order to share common characteristics. • By placing inheriting classes in a common tree, code can be passed down and only characteristics that are unique to this class needs to be defined. • Reduces the amount of code that needs to be created: o improves efficiency and speed of a program o reduces potential errors o aids debugging. **Reusability** • Code that has already been created can be called and used in other parts of code. • This reduces the size of the code saving development time and aiding accuracy and efficiency of the program.

Mark scheme (award up to 6 marks) refer to the guidance on the cover of this document for how to apply levels-based mark schemes*.

Level	Mark	Descriptor
Level 0	0	No rewardable material.
Level 1	1–2	Technical vocabulary is used but it 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 scenario.
Level 2	3–4	Accurate technical vocabulary is used to support arguments but not all are relevant to the issues of the question. There is consideration of relevant issues using logical chains of reasoning but it does not reflect on their relative importance to the given scenario.
Level 3	5–6	Fluent and accurate technical vocabulary is used to support arguments that are relevant to the issues of the question. There is a balanced and wide-ranging consideration of relevant issues, using coherent and logical chains of reasoning that show a full awareness of their relative importance to the given scenario.

Question number	Answer	Mark
3(a)	Award **1** mark for identification and **1** additional mark for appropriate expansion up to a maximum of **2** marks each. • It can contain a mix of data types (1) so an appropriate type can be assigned for each field (1). • Each field can be assigned an identifier (1) so it can be used/accessed directly (1). • Each field (in the record) is usually stored in consecutive memory locations (1), making processing more efficient (1). Accept any other relevant phrasing/wording.	**(4)**

Question number	Indicative content
3(b)	Answers will be credited according to the learner's demonstration of knowledge and understanding of the material using the indicative content and levels descriptors below. The indicative content that follows is not prescriptive. Answers may cover some/all of the indicative content but should be rewarded for other relevant answers.

Example solution

Item in array	1	2	3	4	5	6	Comments
Original array	32	24	56	76	19	17	
	24	32	56	76	19	17	1&2 are compared item 2(24) is lower than item 1(32) so items are swapped
	24	32	56	76	19	17	2&3 are compared Item 3(56) is higher than item 2(32) no swap is made
	24	32	56	76	19	17	3&4are compared Item 4(76) is higher than item 3(56) no swap is made
	24	32	56	19	76	17	4&5 are compared item 5(19) is lower than item 4(76) so items are swapped
	24	32	56	19	17	76	5&6 are compared item 6(17) is lower than item 5(76) so items are swapped
End of first pass	24	32	56	19	17	76	At the end of the first pass the data is still not full sorted so additional passes would be required

Mark scheme (award up to 6 marks) refer to the guidance on the cover of this document for how to apply levels-based mark schemes*.

Level	Mark	Descriptor
Level 0	0	No rewardable material.
Level 1	1–2	Technical vocabulary is used but is not used appropriately. The explanation shows only a superficial understanding of the identified processes. Data is applied correctly to parts of the processes but may contain significant errors. Processes are not always applied accurately and/or efficiently and show a limited understanding of the relationship between the stages.

Level	Mark	Descriptor
Level 2	3–4	Accurate technical vocabulary is used to explain how data is processed but the importance of some stages on the final outcomes is not made clear. There are minor errors in the data entered and used. Appropriate methodologies are applied, showing an understanding of the logical chain of events in the processes but there are some minor inaccuracies in their application.
Level 3	5–6	Fluent and accurate technical vocabulary is used to explain how all stages process data, to achieve the intended outcomes. Data is entered and used accurately at all stages. Appropriate methodologies are applied accurately and efficiently at all stages of the processes, to create accurate and reliable outcomes.

<table>
<tr><th>Question number</th><th>Answer</th><th>Mark</th></tr>
<tr><td rowspan="2">3(c)</td><td>

Example algorithm

```
count = 0
FOR item in Running club member:
    IF item == “yes”:
        count = count +1
PRINT count
```

</td><td rowspan="2"><b>(4)</b></td></tr>
<tr><td>

Award **1** mark for each related descriptive point up to a maximum of **4** marks.

- Suitable ‘count’ variable declared and set to 0 (1).
- Logical use of repetition to check all items in the list (1).
- Correct logic for adding 1 to count for each occurrence of (1).
- Appropriate and logical structure used (1).

Additional guidance:

Also accept:
count+=1
ADD 1 to count

To award final mark the algorithm should use correct indentation to show hierarchy of operation, e.g. IF should be indented under FOR to show that it is performed as part of the FOR loop. PRINT/OUTPUT result should be fully left aligned so as to show it will print after the loop has finished and not every time a new item is added.

</td></tr>
</table>

Question number	Indicative content
3(d)	Answers will be credited according to the learner's demonstration of knowledge and understanding of the material using the indicative content and levels descriptors below. The indicative content that follows is not prescriptive. Answers may cover some/all of the indicative content but should be rewarded for other relevant answers. Features of event driven languages: **Events** • The program responds to individual user actions, such as the entry of data, user pressing a button, etc. • As actions can be assigned to particular events, they are suitable for tasks where the computer only needs to respond to user inputs. Good for use with data processing programs. • The assignment of events makes them suitable for creating graphical user interfaces and buttons that can be clicked, to action regularly performed tasks. • The development of a GUI (Graphical User Interface) based solution may be overly complicated for a simple data handling application. The development and design of the GUI, is likely to extend the development time of a program. **Event handlers** • The event handler is a section of code that denotes the action that will be carried out when a particular event occurs. • Each event (such as 'sort data') will be written as a separate function/subroutine. The modular nature of the code makes debugging and testing easier, as code can be tested in separate 'chunks'. • As events do not affect another if a section of code is not tested correctly, an error may not be noticed. The program may appear to work until incorrect code is run. **Event loops** • Used to monitor the events the program must respond to. Constantly runs, waiting for the predefined event to be triggered. • Constant running uses some CPU processing time even when the program is not actively being used. **Service orientated processing** • Allows execution of code to be linked to particular services that may run in the background. • Allows programs to be linked to a service that it may use, giving the event driven programs greater flexibility in the way they are used. • The drawback of linking a program to a service, is that the program will always run in the background using up system processing time and memory. • In the given scenario, it is unlikely that this type of application would be used, as the program would only be loaded and used infrequently.

Question number	Indicative content
	Time driven • Allows functions in a program to be scheduled, to activate at particular times or at particular intervals. • Allows the programmer to add functions that do not need to be monitored/activated by the user, or tasks that are best scheduled when the system will not be in use by the user. • If the program was to handle an extremely large data set, sorts etc., could be scheduled as a 'batch-process' overnight. As the set of data in the given scenario is very small, it is not a useful feature. **Trigger functions** • Trigger functions are designed to assign a particular event with an action.

Mark scheme (award up to 8 marks) refer to the guidance on the cover of this document for how to apply levels-based mark schemes*.

Level	Mark	Descriptor
Level 0	0	No rewardable material.
Level 1	1–2	Technical vocabulary is used but it 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 scenario. Arguments are not linked to the given scenario.
Level 2	3–5	Accurate technical vocabulary is used to support arguments but not all are relevant to the issues of the question. There is consideration of relevant issues, using logical chains of reasoning but it does not reflect on their relative importance to the given scenario. Various elements of the question are considered but it does not always link arguments to the given scenario.
Level 3	6–8	Fluent and accurate technical vocabulary is used to support arguments that are relevant to the issues of the question. There is a balanced and wide-ranging consideration of relevant issues, using coherent and logical chains of reasoning that show a full awareness of their relative importance to the given scenario. Various elements of the question are carefully considered and arguments are clearly linked to the given scenario.

Question number	Indicative content
4(a)	Answers will be credited according to the learner's demonstration of knowledge and understanding of the material, using the indicative content and levels descriptors below. The indicative content that follows is not prescriptive. Answers may cover some/all of the indicative content but should be rewarded for other relevant answers. Evaluation of the given code, with consideration of the efficiency and effectiveness of the code in relation to: Variables – use of global variables versus local variables. Structure – 'play' function contains all code to execute the game. Consideration of breaking down the code in to functions/subroutines that control individual features: • generate random number • prompt user and check answer against generated number • keeping track of score. The programmer calls global variables using the 'global' command. Generally not viewed as best practice, as it permanently alters the global variable (problematic if the game had a 'play again' option). Better if the variables were passed onto the function, so that they could be used locally without affecting the main global value. Repeated code – the programmer could make better use of subroutines to reduce the amount of code that has to be compiled and run. The scores could be added to separate locations in a list/array rather than as separate variables so that they can be manipulated more efficiently later (when calculating average); reduces memory used and improves the efficiency of the code. Accuracy – the code performs the expected mathematical operations (generate a random addition sum) as outlined in the given criteria. The total score for each round is displayed, but the code does not specifically output a message stating what the 'highest' score was. The program does not fully meet some of the design criteria provided. The use of WAIT when 'calculating average score' would make the computer look like it is thinking; adds unnecessary waiting time for the user. The code is likely to return the average with a large number of decimal places, as the code just prints the answer of the calculation (the computer would store the full result). Displaying a large number of decimal places may not make sense to the user. The programmer could use built in functions to either display only a small number of decimal places or use a function such as ROUND to display only a whole number. The output is all text based, which may not be appropriate for the age group using it.

Mark scheme (award up to 8 marks) refer to the guidance on the cover of this document for how to apply levels-based mark schemes*.		
Level	**Mark**	**Descriptor**
Level 0	0	No rewardable material.
Level 1	1–2	Technical vocabulary is used but it 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 scenario. Arguments are not linked to the given scenario.
Level 2	3–5	Accurate technical vocabulary is used to support arguments but not all are relevant to the issues of the question. There is consideration of relevant issues, using logical chains of reasoning but it does not reflect on their relative importance to the given scenario. Various elements of the question are considered but it does not always link arguments to the given scenario.
Level 3	6–8	Fluent and accurate technical vocabulary is used to support arguments that are relevant to the issues of the question. There is a balanced and wide-ranging consideration of relevant issues, using coherent and logical chains of reasoning that show a full awareness of their relative importance to the given scenario Various elements of the question are carefully considered and arguments are clearly linked to the given scenario.

Question number	Indicative content
4(b)	Answers will be credited according to the learner’s demonstration of knowledge and understanding of the material, using the indicative content and levels descriptors below. The indicative content that follows is not prescriptive. Answers may cover some/all of the indicative content but should be rewarded for other relevant answers. **Compatibility** The program should be tested on as many different computers as possible. Tested on computers with different operating systems or different versions and set-ups of the same operating system. Alternatively, test on different devices to check compatibility with different hardware settings and components. Having access to a wide range of platforms, to form a meaningful test may be unrealistic. So may distribute to others to test in order to get as many different testing platforms as possible. Testing may be time consuming because of the need to test on as wide a range of devices as possible, which may impact on development time. Selecting other users that can provide suitable testing and feedback, would require careful planning. Possibly test that it runs specifically on the hardware and software platform used by the school. May create an emulated/virtualised version of the systems in the school on which to test the program. Emulation/virtualisation may be quite time consuming because of the need to reconfigure the machine every time it is tested on a different platform.

Mark scheme (award up to 6 marks) refer to the guidance on the cover of this document for how to apply levels-based mark schemes*.

Level	Mark	Descriptor
Level 0	0	No rewardable material.
Level 1	1–2	Technical vocabulary is used but it 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 scenario.
Level 2	3–4	Accurate technical vocabulary is used to support arguments but not all are relevant to the issues of the question. There is consideration of relevant issues, using logical chains of reasoning but it does not reflect on their relative importance to the given scenario.
Level 3	5–6	Fluent and accurate technical vocabulary is used to support arguments that are relevant to the issues of the question. There is a balanced and wide-ranging consideration of relevant issues, using coherent and logical chains of reasoning that show a full awareness of their relative importance to the given scenario.

Question number	Indicative content
4 (c)	Answers will be credited according to the learner's demonstration of knowledge and understanding of the material, using the indicative content and levels descriptors below. The indicative content that follows is not prescriptive. Answers may cover some/all of the indicative content but should be rewarded for other relevant answers. **Power** • Use of a language such as HTML/HTML5 would give only limited functionality/computational power, owing to it being a markup language rather than a true programming language. • To successfully implement the package as a web-based product, may need to incorporate different web development language such as Java Script and other support, such as API usage in HTML5 to execute multimedia functionality and additional features. **Performance** • Consider the language/scripting used and how this is implemented. • Using server-side processing means many of the tasks in the web program would be executed by the server. May cause issues in relation to the speed at which a program appears to respond to user interaction. • Could make use of scripting to split certain tasks between the client and server, to improve performance (although performance may be more dependent on the power of the client machine). Executing too many tasks on the server could overload it, especially if a large volume of traffic is expected. • Use of additional client programs, installed locally on the client machine, which interact with the server via the web interface, is one way of splitting the load between the server and client. Using this approach requires competence in a range of different programming and scripting languages. This may increase the development time required. Users may be unwilling to install untested/unknown software. • Use of additional clients may also increase the scope of testing needed, such as additional compatibility and functionally factors that will need testing. **Platform independence** • Using a web-based solution would remove many compatibility issues. Due to the independent nature of many web languages, the operating system that the user has installed should have no impact on being able to access the program. Pupils in the school could access and use the program, using any internet-enabled device in school or at home. • The use of a web platform should reduce the development time, as different versions would not need to be produced for different operating systems. However, the program would need to be tested in different browsers, especially if using plug-ins/extensions.

Question number	Indicative content
	Security • Security of the data held on the system consideration, especially if any user data is stored on the remote server. • If program held and executed on a remote server then, adequate protection of the data being held (such as user names, passwords, etc.) would need to be in place, to avoid threats (viruses, hackers, etc.). • The use of an encrypted connection (https instead of http) would also need to be established in order to protect sensitive data. **Protocols** • Setting up sophisticated security protocols may require additional programmers and data security experts' involvement. Impacts on development time and costs. • Webpages use a standard protocol (http) to communicate between systems, allowing for a platform independent solution. Additional webpage functionality may need to make use of other protocols that enable the use of some web services. Impacts on development time and cost of adding extra services. • The use of web services that may use less common protocols, may impact on the end user as they may need to install additional software, in order to use the product. Users with limited computing capability may find accessing the product difficult, so the product is less likely to reach its intended audience.

Mark scheme (award up to 12 marks) refer to the guidance on the cover of this document for how to apply levels-based mark schemes*.

Level	Mark	Descriptor
Level 0	0	No rewardable material.
Level 1	1–4	Technical vocabulary is used but it 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 scenario. No conclusion is presented or is generic.
Level 2	5–8	Accurate technical vocabulary is used to support arguments but not all are relevant to the issues of the question. There is consideration of relevant issues using logical chains of reasoning but it does not reflect on their relative importance to the given 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.

Level	**Mark**	**Descriptor**
Level 3	9–12	Fluent and accurate technical vocabulary is used to support arguments that are relevant to the issues of the question. There is a balanced and wide-ranging consideration of relevant issues, using coherent and logical chains of reasoning that show a full awareness of their relative importance to the given 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 about Edexcel, BTEC or LCCI qualifications
visit qualifications.pearson.com

BTEC is a registered trademark of Pearson Education Limited

Pearson Education Limited. Registered in England and Wales No. 872828
Registered Office: 80 Strand, London WC2R 0RL
VAT Reg No GB 278 537121

ISBN 978-1-4469-3041-0

9 781446 930410 >