

Cambridge Lower Secondary Computing 8 (2022)

Answers

Answers are given where applicable. Guidance on all activities is available in the Cambridge Lower Secondary Computing Teacher's Guide 8 with Boost Subscription, ISBN 9781398369801.

<https://www.hoddereducation.com/subjects/ict/products/11-14/cambridge-lower-secondary-computing-8-teacher-s-gu>

8.1 Building networks: Network structure and security

Practise (Student's Book page 13)

1 a LAN

b WAN

c PAN

2 b i Networking devices can increase the security risk because there are many more access points to the same network. Each device is potentially at risk of intruders, so the more devices there are on a network, the greater the risk. There are also more users on a larger network, which means that there is a higher chance of user error.

ii Security risks can be prevented by using software such as antivirus, which can find and remove viruses from a network, or users of a network could be trained on ways to keep their devices safe. Note: at this stage, answers may be quite generic; students will investigate how to keep devices safe during the unit.

3 For example:

- Files could be shared with the wrong people on the network; good use of accounts and passwords can prevent this and ensure that the network is well organised to prevent accidental sharing of information.
- A hacker may try to gain access to the network; ensuring that everybody using the network has strong passwords can protect against this.
- If a virus accesses a device, it could spread across the network; antiviruses can help to protect against this.
- The number of devices that people have access to increases the 'surface area' of any attacks to gain access to the network; this means that more people need to be confident in securing their devices as the hackers could have more ways to access the network.

4 For example:

Device	How it changes based on how people use it
Smart speaker	Adjusts volume based on the surrounding noise or the time of day
Smart bulb	Adjusts brightness based on whether anybody is in the room, or the time of day
Smart TV	Switches on and off based on people being in the room

Practise (Student's Book page 20)

For example, students should cover these points as a minimum in their comic strips:

1 A device displaying a piece of data.

2 A device (sender) transferring data to another device (receiver).

3 The receiving device sending the data back to the sending device.

4 The receiving device checking whether the data is the same.

5 If the data is the same, the receiving device is able to use the data.

6 If the data is different, it shows that an error has occurred.

Students may depict the statements above in whatever way they choose, but responses should loosely follow the flowchart in the Student's Book.

Practise (Student's Book page 23)

1–3 Note that students may have chosen entirely different examples and there is sometimes a crossover between spyware and ransomware (the famous WannaCry spy/ransomware, for example). If students have found a 'crossover' threat, then they should find an alternative too.

Practise (Student's Book page 29)

1–2 Example questions and answers:

- *What are the most common ways of transmitting viruses between devices?*
 - Malicious email attachments.
 - Files disguised as something else.
 - Accidentally downloading them from the internet.
- *What does an antivirus program do when it finds a virus?*
 - It quarantines the file so that it cannot be accessed.
 - It lets the user know that a virus has been found.
- *What do firewalls do?*
 - They examine the data entering and leaving the network to check they do not break the rules.
- *What is a trojan?*
 - It is software that pretends to be something that it is not.
 - Trojans track keystrokes and mouse movements – even sometimes webcams.
- *What is adware?*
 - It is a type of spyware that increases the number of adverts that a user has on their computer system; this often means that adverts are displayed to users inside software that they are using, and can sometimes entice users to click on links to insecure or malicious websites or downloads.

Go further (Student's Book page 29)

2 As long as students justify their answers, any of the products is an acceptable suggestion for the youth group, e.g.:

I would recommend that the youth group use Norton 360, as this is the cheapest software that includes the most features. It includes an antivirus, anti-malware and a firewall. It would protect the youth group against the threats of ransomware, spyware, adware, viruses and trojans. If users from the youth group were to use this protection, the network would be less likely to be damaged by threats.

8.2 It's all in the planning: Pseudocode and algorithms

Warm up (Student's Book page 33)

1 a You do not know whether online messages/emails stay private because they are sent from one device to another and could be intercepted or read by another person.

b You assume that messages are private because you send them to your intended recipient. Anyone could read a letter before it is delivered; there are ways to open a letter and read it, such as using steam to unseal the envelope. If you knew that a private letter had been read by someone else, you could feel unsettled.

2 For example: posted letters, email, instant messaging, social media. A letter is kept secure in a sealed envelope; emails and instant messaging are sent to the recipients' account and not to anyone else.

Practise (Student's Book page 37)

1 For example:

a Variables need to be suitably named and linked to the scenario, e.g. 'salary'/'salaryIncrease'.

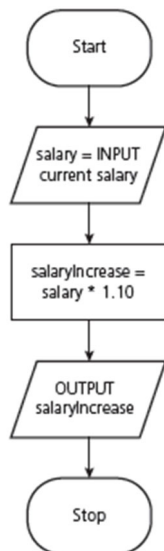
b $\text{salary} * 1.10$

c User enters the salary.

Salary is multiplied by 1.10.

Output the new salary.

2 For example:



3 For example: Pseudocode can be quicker to write out compared with drawing a flowchart. It is therefore easier to edit and improve pseudocode, as you can move the text, edit it and cross it out instead of possibly having to start drawing a flowchart again from scratch.

Practise (Student's Book page 40)

Part 1

4 Similarities and differences concern the characteristics of pseudocode, e.g. the use of uppercase letters, indentation, START, STOP, IF, THEN, ELSE.

Part 2

1–2 The pseudocode is asking the user 'What is the capital of England?' and setting the answer as London. If the answer is equal to the answer entered by the user, then the output is correct. If it is not equal to the answer, then the output is incorrect.

4 Preference will be individual, e.g.:

- Pseudocode takes less time to write out; flowcharts take longer due to drawing shapes.

- It is easier to see the flow of a program algorithm as a flowchart as the steps are clearer.
- Pseudocode is similar to how a program may be written, e.g. in a programming language such as Python, so the writing stage can be easier.
- The shapes in a flowchart aid understanding of the selection and help to ensure that both the True and False outputs have been considered.

Practise (Student's Book page 42)

Part 1

1 The last data item to be checked will be the data item being searched for:

a apple, banana, pear

b apple, banana, pear, grape, melon

c apple, banana, pear, grape, melon, lemon, orange, mango, papaya

d apple, banana, pear, grape, melon, lemon, orange

2 An advantage is that you could find the item you are looking for quickly if it is one of the first few data items in the list.

A disadvantage is if the data item being searched for is in the last few data items in the list; then the search would take longer to locate the searched-for item.

Part 2

1 Ensure that students are examining all items in turn, from left to right, up to and including the searched-for data item.

2 A linear search can be used on any list of items, e.g. images, text, letters or numbers, as the list does not need to be in any order. As long as a comparison can take place, a linear search can be carried out.

Practise (Student's Book page 46)

Part 1

1–2 For example solutions to each pseudocode challenge, refer to the program files below:

- **PseudocodeChallenges1(Teacher).py**
 - Conditional statement: if the light level is equal to 50.
 - Test values could include 40, 50, 60; one higher, one lower and then the value used in the conditional statement (50).
- **PseudocodeChallenges2(Teacher).py**
 - Conditional statement: if the light level is not equal to 120.
 - Test values could include 50, 120, 160; one higher, one lower and then the value used in the conditional statement (120).
- **PseudocodeChallenges3(Teacher).py**
 - Conditional statement: if the light level is less than 100.
 - Test values could include 50, 100, 150; one higher, one lower and then the value used in the conditional statement (100).
- **PseudocodeChallenges4(Teacher).py**
 - Conditional statement: if the light level is greater than 60.
 - Test values could include 40, 60, 100; one higher, one lower and then the value used in the conditional statement (60).
- **PseudocodeChallenges5(Teacher).py**
 - Conditional statement: if the light level is equal to 200.

- Test values could include 50, 200, 220; one higher, one lower and then the value used in the conditional statement (200).

Part 2

1–2 The errors to be fixed are that the conditional statement needs to be less than 90 (< 90) and the image displayed should be changed to SAD.

3 For example:

- comparison operator (<, >, ==, !=, <=, >=)
- value to be compared in conditional statement (50)
- image to be output (heart)
- spelling (e.g. 'dispay' should be 'display')
- indentation (the output moved in under the conditional statement)
- colon at the end of the conditional statement (:)
- – uppercase letters for image output (HEART)
- – correct code for light sensor in MicroPython:

```
if display.read_light_level() > 50
```

and to display an image:

```
display.show(Image.HEART)
```

4 Students should complete the 'Expected outcome' column before testing the program on the micro:bit.

Test number	Data entered	Expected outcome	Pass/fail
1	light level = 0	Sad face	
2	light level < 90	Sad face	
3	light level > 90	Nothing	

5 By testing on the micro:bit, students are able to complete the final column, stating whether the program passed or failed.

Test number	Data entered	Expected outcome	Pass/fail
1	light level = 0	Sad face	Pass
2	light level < 90	Sad face	Pass
3	light level > 90	Nothing	Pass (however if the value detected first was less than 90 then the screen would not clear)

Practise (Student's Book page 49)

1 If the A or B button is pressed on the micro:bit, then the output will be run, displaying the text 'Access Granted'. The output will also run if both the A and B buttons are pressed together.

2 The Boolean operator needs to be changed from OR to AND.

3–4 The A and B buttons need to be pressed at exactly the same time to activate the output with the Boolean operator. This may cause some tests to fail. It may then require a visual check of the program code.

5 The data type generated is Boolean as the output is only ever TRUE or FALSE.

6–7 Example test plan:

Test number	Data entered	Expected outcome	Pass/fail
1	A button pressed	Nothing	Pass
2	B button pressed	Nothing	Pass
3	A and B buttons pressed	'Access Granted' displayed	Pass

8 Students may consider the different inputs they could use and think about using the A or B button to send their messages more quickly.

Practise (Student's Book page 52)

1 If the user presses the A button on one of the micro:bits, then it will send a radio signal to the other micro:bit in the same radio group. The micro:bit will display the numbers 8, 5, 12, 12, 15. Each number represents the position of a letter in the alphabet: the word is 'HELLO'.

2–4 For example:

micro:bit 1 student A

Test number	Data entered	Expected outcome	Pass/fail
1	A button pressed	micro:bit screen cleared and message sent to other micro:bit	Pass
2	Message received on second micro:bit	8, 5, 12, 12, 15 displayed on the micro:bit in turn	Pass

micro:bit 1 student B

Test number	Data entered	Expected outcome	Pass/fail
1	A button pressed	micro:bit screen cleared and message sent to other micro:bit	Pass
2	Message received on second micro:bit	8, 5, 12, 12, 15 displayed on the micro:bit in turn	Pass

5 42 is set in the example, however each pair will have their own radio group set.

6 For example: 'microbit' = 13, 9, 3, 18, 15, 2, 9, 20.

7 For example:

Test number	Data entered	Expected outcome	Pass/fail
1	A button pressed	micro:bit screen cleared and message sent to other micro:bit.	Pass
2	Message received on second micro:bit	8, 5, 12, 12, 15 displayed on the micro:bit in turn	Pass
3	B button pressed	No output on either micro:bit	Pass

8 The 'Scenario' asks students to create a program to send a message securely from one micro:bit to another; use of the radio function will enable this.

Practise (Student's Book page 56)

Part 1

1–4 The hidden word is *bye*.

Part 2

1–3 Each person's program will differ depending on their name.

Go further (Student's Book page 57)

1–3 Students' checklists should include the following checks.

- Spelling is correct.

- Brackets appear in the correct places.
- Brackets are opened and closed.
- A colon is placed at the end of a conditional statement.
- **If** appears at the start of a conditional statement.
- Quotation marks are placed around any strings.

4 For example:

Test number	Input	Expected outcome	Pass/fail	Error found
1	A button is pressed	Scroll the text Hello, World across the micro:bit display	Line 6 syntax error	Display spelled incorrectly
2	A button is pressed	Scroll the text Hello, World across the micro:bit display	Line 6 syntax error	Missing quotation mark at the end of the text to be displayed

Test number	Input	Expected outcome	Pass/fail	Error found
3	A button is pressed	Scroll the text Hello, World across the micro:bit display	Line 6 syntax error	Missing bracket at the end of the text to be displayed
4	A button is pressed	Scroll the text Hello, World across the micro:bit display	Line 5 syntax error	Line 5 has missing end bracket and colon
5	B button is pressed	Display a heart	Line 7 syntax error	Missing bracket
6	B button is pressed	Display a heart	Line 8 syntax error	Missing capital letter on 'Image' and all capitals on 'HEART'
7	A and B button pressed	Play music 'power up'	Line 10 syntax error	Music should be in lowercase not capitals

5 As the testing is completed, students may find and fix one error at a time. This means that students are developing the program code and then continuing to test it. It allows the program to improve and be tested continuously.

Challenge yourself (Student's Book page 58–59)

1–3 Testing can be done by looking for errors in the way the program code is written or by using a test plan to test different inputs and what you expect to happen.

4 The conditional statement can have only one of two outputs: TRUE or FALSE. This is a Boolean output.

5 Example checklist:

- Spelling is correct.
- Brackets appear in the correct places.
- Brackets are opened and closed.
- A colon is placed at the end of a conditional statement.
- **If** appears at the start of a conditional statement.
- Quotation marks are placed around any strings.
- Uppercase letters are used for any images that are output.
- Each section of code is written as expected, e.g. using full stops and underscores in the correct places in the conditional statement for the shake input:

```
if accelerometer.was_gesture('shake'):
```

- The radio network is set between 0 and 255.
- The 'radio' library has been imported.
- The program starts with:

```
from micro:bit import *
```

- The lines to run if the conditional statement is TRUE or FALSE are indented.

6 For example: Testing is important throughout the development process, and it starts when the algorithm is created. It is easier to find and fix an error in a smaller amount of code or a section of an algorithm instead of at the end, when there is more to examine to find the error.

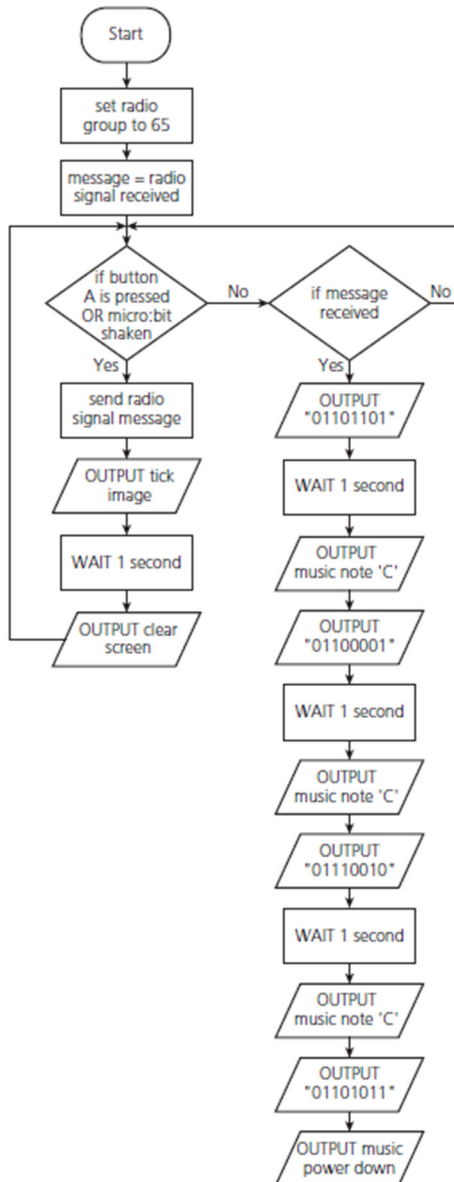
Final project (Student's Book page 59)

1 For example: Inputs will be the A button or the micro:bit being shaken.

2 Example solution:

- In the flowchart example, using the A button OR the micro:bit being shaken will use a Boolean operator.
- The use of ASCII will be seen in the string to output each individual binary number.
- There will be a sound in between numbers to tell the user where there is a break between letters and another one at the end to tell them when the message is complete.

3 Example:



4 Example: HELP = 01001000 01000101 01001100 01010000

5 Students should save their flowcharts as different versions to show the stages of development.

6–7 For example: You can test your program throughout the different stages of the program development. The testing needs to be structured as a test plan to document what tests have been carried out, what has been found and any improvement or amendments required.

Example test plan:

Test	Input	Expected outcome	Pass/fail
1	A button pressed on micro:bit 1	Message displayed on micro:bit 2	Pass
2	micro:bit 1 shaken	Message displayed on micro:bit 2	Pass
3	A button pressed on micro:bit 1 and shaken	Message displayed on micro:bit 2	Pass

8–9 Students will need to explain how they tested at different stages of the program development, when creating both the algorithm and the program code. This can be done using the test plan and a checklist of possible errors.

10 For example: A linear search looks at each data item in turn, from start to finish or until it has found the item it is looking for. A computer could use it when trying to locate in the ASCII table the letter or symbol that a binary number represents.

8.3 Let's talk technology: What's going on inside

Practise (Student's Book page 66)

1 Students' pages should contain the following information.

- Operating system: the software that provides users with a method of interacting with a digital device.
- Utility software: the software that helps users to manage their digital device.
- Students may include images associated with Windows, Android, iOS or any other alternative; some students may show a command-line based interface, which is also acceptable, and images may be icons representing the interface or an example of the interface.
- Purpose of an operating system: provides an interface; manages memory, the device and security.
- Examples of utility programs:
 - Antivirus: lets the user know whether there are any harmful programs on their computer and helps them to remove them.
 - Firewall: prevents unauthorised access to the user's computer.
 - Data compression: reduces the size of files to help save storage space on the computer.
 - Disk defragmentation: reorganises files on the computer's storage area to make them quicker to access.
 - Screensavers: animated images or backgrounds that appear on screen when the computer is not being used for a period of time.
 - Backup: used to make a copy of a selected file.

2 a Operating systems vary and may include:

- iOS (iPhone)
- Android (Samsung Galaxy)
- Windows 11 (PCs).

b Students who opt to investigate their mobile devices could return an extensive range of examples, from sounds and notifications to control centres and flight mode. For example: on a device such as an iPhone, utility applications may include: Notifications, Sounds & Haptics, Screen Time, Display & Brightness, Passwords; on an Android device they could include tools such as Set Do Not Disturb Schedule or Turn on Wi-Fi.

Practise (Student's Book page 68)

1 a School = 01010011 01100011 01101000 01101111 01101111 01101100

b Computer = 01000011 01101111 01101101 01110000 01110101 01110100 01100101 01110010

2 For example: Tara = 01010100 01100001 01110010 01100001

Computational thinking – pattern recognition (Student's Book page 68)

1 a Hello friend.

b My age is 13.

2 Responses here will vary but should be a number of characters long, such as the ‘Hello friend’ example from part 1a.

3 a Allow appropriate time in class for students to attempt to convert their partner’s message back into words using the ASCII conversion chart.

- b – All uppercase letters start with 0100; all lowercase start with 0110.
- There is only a 1 binary digit change in the code as you move from one alphabetic character to the next.
 - Lower- and uppercase representations all begin with 01; special characters all begin with 001.
 - All numerical characters start with 0011 and the remaining four digits contain the binary representation for that digit, e.g. 2 in ASCII is 00110010.

Practise (Student’s Book page 71)

1

Hundred thousands	Ten thousands	Thousands	Hundreds	Tens	Ones
100 000	10 000	1 000	100	10	1

2

	Hundred thousands 100 000	Ten thousands 10 000	Thousands 1 000	Hundreds 100	Tens 10	Ones 1
a					7	9
b				1	4	6
c			1	2	2	4
d	1	0	0	0	3	7

3 Encourage students to use subscripts, to show number bases in all representations.

a 94_{10}

b 63_{10}

c 1011_{10} (one thousand and eleven)

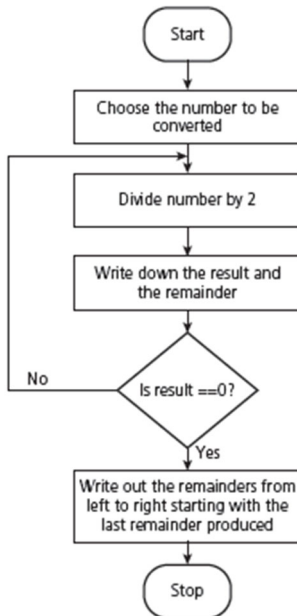
d 13_{10}

e 131_{10}

f 111_{10} (one hundred and eleven)

Practise (Student’s Book page 73)

1 For example:



2 For example:

- First, write down the place values for a binary number.
- Examine the highest place value and check whether this value can be subtracted from the original decimal number without creating a negative number.
- If it can, place a 1 underneath the place value and keep a note of the result of your calculation.
- If it cannot, place a 0 underneath the place value.
- Go to the next place value, and repeat the process to see whether that value can be subtracted from the result, without creating a negative number in the answer.
- If it can, place a 1 underneath the place value and keep a note of the result of your calculation.
- If it cannot, place a 0 underneath the place value.
- Continue repeating this process until you have examined all the place value.

3 a

129 / 2	=	64	Remainder	1
64 / 2	=	32	Remainder	0
32 / 2	=	16	Remainder	0
16 / 2	=	8	Remainder	0
8 / 2	=	4	Remainder	0
4 / 2	=	2	Remainder	0
2 / 2	=	1	Remainder	0
1 / 2	=	0	Remainder	1

1000001₂

b

45 / 2	=	22	Remainder	1
22 / 2	=	11	Remainder	0
11 / 2	=	5	Remainder	1
5 / 2	=	2	Remainder	1
2 / 2	=	1	Remainder	0
1 / 2	=	0	Remainder	1

00101101₂ (encourage students to use 8 bits)

c

98 / 2	=	49	Remainder	0
49 / 2	=	24	Remainder	1
24 / 2	=	12	Remainder	0
12 / 2	=	6	Remainder	0
6 / 2	=	3	Remainder	0
3 / 2	=	1	Remainder	1
1 / 2	=	0	Remainder	1

01100010₂ (encourage students to use 8 bits)

4 For example:

a

128	64	32	16	8	4	2	1
Can 128 be subtracted from 129?	Can 64 be subtracted from 1?	Can 32 be subtracted from 1?	Can 16 be subtracted from 1?	Can 8 be subtracted from 1?	Can 4 be subtracted from 1?	Can 2 be subtracted from 1?	Can 1 be subtracted from 1?
Yes, result = 1	No	No	No	No	No	No	Yes, result = 0
1	0	0	0	0	0	0	1

b

128	64	32	16	8	4	2	1
Can 128 be subtracted from 45?	Can 64 be subtracted from 45?	Can 32 be subtracted from 45?	Can 16 be subtracted from 13?	Can 8 be subtracted from 13?	Can 4 be subtracted from 5?	Can 2 be subtracted from 1?	Can 1 be subtracted from 1?
No	No	Yes, result = 13	No	Yes, result = 5	Yes, result = 1	No	Yes, result = 0
0	0	1	0	1	1	0	1

c

128	64	32	16	8	4	2	1
Can 128 be subtracted from 98?	Can 64 be subtracted from 98?	Can 32 be subtracted from 34?	Can 16 be subtracted from 2?	Can 8 be subtracted from 2?	Can 4 be subtracted from 2?	Can 2 be subtracted from 2?	Can 1 be subtracted from 0?
No	Yes, result = 34	Yes, result = 2	No	No	No	Yes, result = 0	No
0	1	1	0	0	0	1	0

Practise (Student's Book page 77)

1

Input A	Input B	Output X
0	0	0
0	1	1
1	0	1
1	1	1

2

Input A	Output X
0	1
1	0

Practise (Student's Book page 79)

1 Students' answers will vary considerably, but encourage all students to investigate a minimum of two devices, e.g.:

Digital device	RAM available
Laptop	16 Gb
Tablet	4–8 Gb
PC	16–64 Gb
Mobile phone	4 Gb

2 For example (using the table in question 1):

a Mobile phone, tablet, laptop, PC.

b The fastest would be the PC. It has the highest amount of RAM, which means that it can store more programs and data currently in use, so the processor spends less time accessing the code and data needed and loading it from the device's hard disc drive into RAM.

Practise (Student's Book page 84)

For example:

A pick-and-place robot used in large warehouses to move items from one location to another to fulfil customers' orders: The robot uses vision technology to identify, grab and move objects from one location to another while avoiding obstacles. It has a pre-programmed path that it is expected to follow but, using data collected from cameras, it identifies obstacles and is able to work out how to manoeuvre around them.

Advantages/disadvantages should reference some of the following points:

Advantages	Disadvantages
<ul style="list-style-type: none"> ● Cost effective ● Reduced time to complete tasks ● High-quality work ● Ability to work in dangerous environments 	<ul style="list-style-type: none"> ● Potential job losses ● Initial high cost of investment ● Need to retrain staff ● Need for support/maintenance staff ● Over-reliance on technology

Practise (Student's Book page 85)

1–2 Responses will vary and may include examples similar to the one provided, e.g. the use of AI and AR to support trekking expeditions or to suggest potential routes home during inclement weather or peak travel times. Students' suggestions may include the use of AR glasses; others may be able to identify applications where AR is used to project images onto the user's real-life environment using mobile-phone technology, e.g. in games such as Pokémon GO.

AR in education could include: allowing users to visit unusual or dangerous locations, e.g. volcanoes/the top of Mount Everest/space; allowing users to experience different times in history.

AR in entertainment could include: laser-tag and other games; apps that allow users to try new hairstyles or outfits; sports; holographic TVs.

Go further (Student's Book pages 86–87)

1–2 For example:

pitter patter, pitter patter, listen to the rain

pitter patter, pitter patter on the windowpane

could be compressed to:

pitter patter, pitter patter, listen to the rain, pitter patter, pitter patter on the windowpane
∞ listen to ① rain ∞ on ① windowpane

3–4 For example:

Symbol	Text represented
∞	pitter patter pitter patter
①	the

Challenge yourself (Student's Book pages 87–88)

For example:

Input A	Input B	Input C	Output X = NOT A	Output Y = B AND C	Output Z = X OR Y
0	0	0	1	0	1
0	0	1	1	0	1
1	0	0	0	0	0
1	0	1	0	0	0
0	1	0	1	0	1
0	1	1	1	1	1
1	1	0	0	0	0
1	1	1	0	1	1

Final project (Student's Book pages 89–91)

Part 1

1 Students' presentations should contain the following information.

a *Systems software* refers to computer programs that are designed to manage or control the computer's hardware while the user completes tasks using other programs known as *applications software*. Systems software can be divided into two main categories:

- Operating systems: The software that manages all the computer hardware and software; they can also provide an interface between the user and the computer (examples include Windows, iOS, Android).
- Utility programs: The programs the user must select to complete a specific task (examples include antivirus, firewall, data compression, disk defragmentation, screensavers, backup).

b ROM (read-only memory): Memory used to hold instructions that are important to the digital device but not to the user.

RAM (random access memory): Memory that is used to store the programs and data currently in use, e.g. when you load a word-processing application and open a file.

ROM often holds an important program called the *boot-up program*. This is the program that locates the computer's operating system in secondary storage and loads it into RAM so that the processor can provide the user with a way of using the digital device.

For the telecommunication company's goggles, RAM would store any data being processed about the call being made, e.g. the number being called and the images and sound associated with the call.

Main features:

RAM	ROM
<ul style="list-style-type: none"> ● Volatile; this means that the data is lost when the computer is switched off ● Can be read from and written to ● Holds the computer programs and data when the computer is running ● If many programs are running on a computer at the same time, this can slow the computer down because too many data items and instructions are being moved into and out of RAM 	<ul style="list-style-type: none"> ● Non-volatile; this means that the data is not lost when the computer is switched off ● Can be read from but cannot be written to (the user is unable to change the contents of ROM) ● Holds programs and instructions that the computer needs to be able to start up ● The contents of ROM are programmed by the computer manufacturer

c AI is about using complex computer programs and data to make decisions. It does not always include imagery. The user can interact with the AI tool in many ways, e.g. with text, images, video, sound.

With AR, the user interacts with a digital image that is projected onto their own real-world environment. The AR application may use AI to help with its interaction with the user.

d For example: *Machine learning* is when an AI system can use data to make decisions without the help of a human.

i Digital assistants in homes/smart streaming services where suggestions are made about shows the user might like.

ii Social-media applications, e.g. Pokémon GO; also in shopping and travel apps to help the user to make decisions.

2 a Responses should include reference to the following points.

Each digital device generates a special combination of 1s and 0s to represent each character available to the user on the device's keyboard. When a keyboard character is pressed, the correct combination of 1s and 0s are generated so that the processor knows which letters/symbols have been produced. 1s and 0s are generated using electrical pulses. This is necessary as the processor does not understand the letters of the computer keyboard; only 1s and 0s.

b Student responses here will vary depending on student logins to school computing systems. Encourage students to show only their username (not their password). Ask students to write out their username in its usual format and then show underneath how each character could be represented in ASCII format, e.g.:

Username	A	D	1	0	9				
ASCII	01000001	01000100	00110001	00110000	00111001				

c 128 in binary is 10000000₂; in ASCII it is 00110001 00110010 00111000.

Part 2

1

A	B	V = A OR B
0	0	0
0	1	1
1	0	1
1	1	1

2

A	B	V = A AND B
0	0	0
0	1	0
1	0	0
1	1	1

3

A	V = NOT A
0	1
1	0

Part 4

1 a **Step 1:** Divide 198₁₀ successively by 2 until the result of the division is 0.

198 / 2 = 99, remainder is 0

99 / 2 = 49, remainder is 1

49 / 2 = 24, remainder is 1

$24 / 2 = 12$, remainder is 0

$12 / 2 = 6$, remainder is 0

$6 / 2 = 3$, remainder is 0

$3 / 2 = 1$, remainder is 1

$1 / 2 = 0$, remainder is 1

Step 2: Read from the bottom (MSB) to the top (LSB) as 11000110.

So, 11000110 is the binary equivalent of decimal number 198.

b Step 1: Divide 68_{10} successively by 2 until the result of the division is 0.

$68 / 2 = 34$, remainder is 0

$34 / 2 = 17$, remainder is 0

$17 / 2 = 8$, remainder is 1

$8 / 2 = 4$, remainder is 0

$4 / 2 = 2$, remainder is 0

$2 / 2 = 1$, remainder is 0

$1 / 2 = 0$, remainder is 1

Step 2: Read from the bottom (MSB) to the top (LSB) as 1000100.

So, 1000100 is the binary equivalent of decimal number 68.

2 a

128	64	32	16	8	4	2	1
0	1	0	0	1	1	0	1

$$64 + 8 + 4 + 1 = 77$$

b

128	64	32	16	8	4	2	1
1	0	0	1	1	0	1	0

$$128 + 16 + 8 + 2 = 154$$

3

128	64	32	16	8	4	2	1
0	0	1	1	0	1	1	0

= 54 in denary

Using the place-value method:

Write out the headings (always use 8 bits).

Write the 1s and 0s underneath the headings (always start from the right-hand side (LSB) in case

the number was not written using 8 bits.

Add up the value of the place-value headings with a 1 underneath them to calculate the denary

value of the number being represented.

Part 5

1–2 Student responses will vary, but could include the following examples.

Industry

- Some companies use AR and mobile-phone apps or special glasses to overlay digital information onto a staff member's real-life view; this can increase their productivity as their live targets are updated regularly.
- Some also use AR to generate designs for new products to allow designers to interact with and test the new product before it is produced.

- Robots are used in many production lines, e.g. car manufacturing (but students can list many more in their presentations).

Advantages of AR

- Provides users with more experience with products (e.g. they can try them in their own homes).
- Brings the store/business to the customer.
- Makes the company stand out from others in a more creative way.

Disadvantages of AR

- May need to publish regular updates for new products.
- Can be expensive to develop.
- Not all customers will be able to use the technology.

Education

- Robots used to support students in developing mathematical thinking skills.
- Telepresence robots to support remote education.
- AR and VR enhance learning experiences by helping students to visualise learning; some use VR comic books/animations as a learning tool and game-based learning to capture students' interest.

Advantages of AR

- Allows students to learn in an interactive environment.
- Provides a media-rich learning experience (fun element).
- Allows students to interact with experiences that are not normally available.

Disadvantages of AR

- Budgets can be limited in schools.
- Applications may not provide the learning experiences the school needs.
- There may not be enough equipment for all students.

Entertainment

- • Humanoid robots are often used to carry out tricks either with remote controls or pre-programmed tasks; some toys based on robotic technology are used for entertainment.
- AR is widely used in social-media apps and filters, and many games now use AR technology.
- AR is used in many TV shows or in public relations/advertising on TV where a unique symbol or character is identified on the user's mobile-phone camera, and this can be used to activate a task or another application.

Advantages of AR

- Provides users with an interactive experience.
- Increases the fun element of gaming.
- Can provide users with a more realistic experience in their own reality (living room/garden/school, etc.).

Disadvantages of AR

- Software can require a lot of additional storage.
- May need a lot of RAM to run the application.
- May need additional hardware, and this can be expensive.

8.4 Testing conditions: Developing games

Learn (Student's Book pages 98–99)



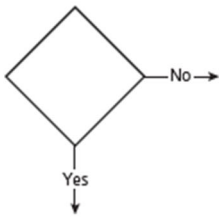
Terminator – used at the start and end of a flowchart to signify the start and end.



Process – used when something needs to happen, e.g. an action or calculation.



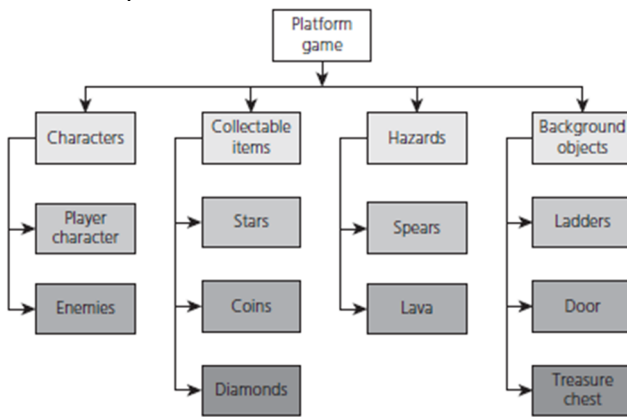
Input/output – used when data needs to be input into or output from the program.



Decision – used when the path of a program will be determined by the check of a condition.

Practise (Student’s Book page 100)

1 For example:



2–3 For example: The player’s character is not visible in the image in the ‘Practise’ box, so students may not identify this. Some students may include enemies as a hazard as they will need to be avoided by the player. There is not a set order in which the elements of the game should be created. There are, however, some tasks that should come before others, e.g. creating the background first helps with placement and movement of the other elements.

Practise (Student’s Book page 102)

1–2 If the user enters a height of 140, then the program will show ‘Welcome aboard! Enjoy the ride.’ This is because the result of the condition check is FALSE in the **if statement**; therefore it performs the **else** selection.

4 The output should be ‘Welcome aboard! Enjoy the ride.’

5 The output should be ‘Sorry, you are too short for this ride.’

Practise (Student's Book pages 103–104)

1–2 Any value between 5 and 20, e.g. 15.

4 If running correctly, the program should display the following.

- Temperature less than 5: 'Wear your gloves and scarf.'
- Temperature greater than 20: 'Take your sunhat.'
- Temperature of 10: 'You might need your coat.'

Learn (Student's Book pages 113–114)

- A: Score = 45
Question set 1 receives 30 points as both answers are correct.
Question set 2 receives 15 points for one correct answer.
- B: Score = 10
Question set 1 receives –5 points as both answers are incorrect.
Question set 2 receives 15 points for one correct answer.
- C: Score = 5
Question set 1 receives –5 points as both answers are incorrect.
Question set 2 receives 15 points for one correct answer.
Question set 3 receives –5 points as both answers are incorrect.
- D: Score = 60
Question set 1 receives 30 points as both answers are correct.
Question set 2 receives 30 points as both answers are correct.
- E: Score = –5
Question set 1 receives –5 points as both answers are incorrect.
- Draw students' attention to the pseudocode, which gives a framework for them to follow when developing the code for their own questions.
- Remind students that they will be writing the code for the World Room as a sub-program.

Practise (Student's Book pages 117–119)

7 When students test their program, they need to print the different variables in their program to check that it works correctly (this is suggested in the task). Creating a test plan would not be appropriate as numbers are being randomly generated and the user does not provide the data being entered.

8 They needed to use the 'random' library because it generates a random number that helps to develop the game and gives the user an opportunity to gather more points. Without this library, students would not be able to generate the random number needed.

Computational thinking – algorithms (Student's Book page 119)

1 a The program should output 'x is less than y'.

b The program should output 'x is same as y'.

c The program should output 'x is less than y'.

2 a The program should output 'Good morning'.

b The program should output 'Good afternoon'.

c The program should output 'Good morning'.

d There are a range of answers, e.g. $x = 4$, $y = 7$, $z = 12$.

Challenge yourself (Student’s Book page 121)

1 List of shapes could include:

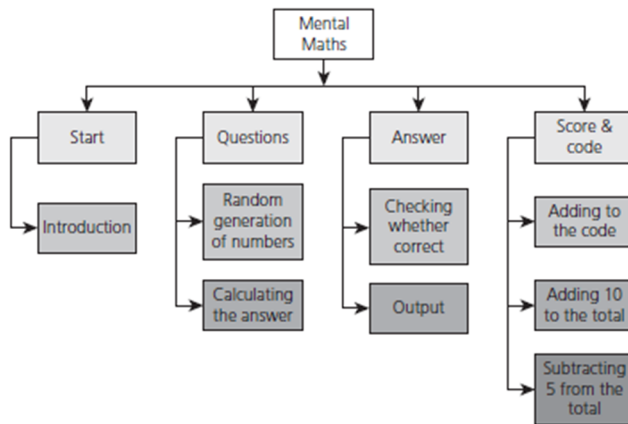
- triangles for the ears
- squares and rectangles for the legs/paws
- rectangle for the body
- squares for the eyes
- rectangle for the nose
- line for the mouth.

2–3 For example:

Order	Part of the image
1	Main body
2	Rear left leg
3	Left ear
4	Right ear
5	Rear right leg
6	Front legs
7	Left eye
8	Right eye
9	Nose
10	Mouth

Final project (Student’s Book pages 122–125)

1 For example:



2 For example:

Variable name	Data type	Justification
num1	Integer	It will be a whole number randomly generated between 1 and 10.
num2	Integer	It will be a whole number randomly generated between 1 and 10.
answer	Float	The division question may not give a whole number.
score	Integer	It will be a whole number as the scores for correct and incorrect answers are whole numbers.

11–12 For example:

secretCode	userGuess	score	Output
comp	comp	150	Congratulations! You cracked the code and achieved a high score.
comp	comp	50	Well done. You managed to crack the code, but your score is not so good.
comp	science	70	Oh dear! You didn't manage to crack the code or achieve a good score.
comp	science	120	Well done. A good score but you didn't manage to crack the code.

13–15 For example:

Test number	Test data	Expected outcome	Pass/fail
1	2 correct answers Bonus number within 10 of the generated number Treasure chest code = coat	Score = 40 Oh dear! You didn't manage to crack the code or achieve a good score.	Pass
2	4 correct answers Bonus number within 10 of the generated number Treasure chest code = comp	Score = 80 Well done. You managed to crack the code, but your score is not so good.	Pass
3	1 correct answer Bonus number guessed Treasure chest code = temp	Score = 100 Oh dear! You didn't manage to crack the code or achieve a good score.	Pass
4	2 correct answers Bonus number guessed Treasure chest code = cone	Score = 200 Well done. A good score but you didn't manage to crack the code.	Pass
5	4 correct answers Bonus number guessed Treasure chest code = comp	Score = 400 Congratulations! You cracked the code and achieved a high score.	Pass

8.5 Data input: Data collection and validation

Warm up (Student's Book pages 128–129)

Part 1

1–2 Image 1 is a computer-generated model that shows what the finished product might be like. If, however, the user were to start amending data used to create that model, e.g. by adding colour/changing the number of doors, the model would become a simulation. Data simulations may produce a visual output, but they are also about the relationships between data sets and how one affects the other. Changing the data and experimenting with the model turns it from a model into a simulation.

Image 2 is also a model (until input data is changed to see how that might affect the shape of the graph, e.g. the data input into the model could be altered in some way, such as the price of cars being increased, which could lead to a reduction in cars sold!).

The key difference is that data models contain data and sometimes formulae or algorithms that represent real-life scenarios as they currently exist.

A model can become a simulation when input data is altered to show how it affects the outcome from the model.

Part 2

1 An example of a database application is MS Access.

Features:

- Forms to allow data to be collected for a model.
- Tables to store data relating to the model.
- Queries to allow users to extract useful data from the model.
- Data types to ensure that the model can process the data in the correct way.

2 An example of a spreadsheet application is MS Excel.

Features:

- Cells to record data items that make up the model.
- Labels to help the user understand the model.
- Formulae to link data items in the model.
- Conditional formatting to highlight important data in the model.

Part 3

1 Database:

- Few calculations involved
- Needs to be able to search effectively for books
- Needs to monitor overdue loans
- Needs to store members' details

2 Database:

- Few calculations involved
- Needs to be able to search effectively for students' contact details
- Needs to store assessment results
- Needs to add new data each year

3 Spreadsheet:

- Calculations may be needed to work out students' averages and grades
- Calculations are also needed to work out class averages

4 Spreadsheet:

- Calculations are needed to work out fastest times and average times in races

Computational thinking – decomposition (Student's Book pages 131–132)

1 Different levels of detail/different amounts of data stored for each student; this would make it difficult to process.

2 Break down 'Student name' into two headings: first name and surname.

Possible additional headings for 'Student address' could include: house number, street.

3 'Age' and 'Stage' could be difficult to search as not all are recorded using the same data format, and recording it as text makes it difficult to search, e.g. searching for 'Age' >12 would not work.

4 Both should be stored as numerical data.

Practise (Student's Book pages 133–137)

Part 1

1 'Student name' and 'Tutor name' could both be broken down to 'First name'/'Initial' and 'Surname'.

2 Create a field to record the session type instead of having lesson data and exam data. Session type could have only one of two answers, e.g. Exam or Lesson. Then add Session date/Time instead of the two fields Lesson time and Examination time.

3 As students must complete all grades in order, there is no need to record the current grade *and* examination grade. If a student is Grade 1 currently, then the next exam they will

take will be Grade 2. There is no need to record unnecessary data in a data model, as this could lead to increased processing time in large models.

4 New headings could include SessionID (this would be needed as a primary key), StudentInitial, StudentSurname, YearGroup, Instrument, CurrentGrade, SessionDate, SessionTime, Lesson/Exam, TutorSurname, Online/InPerson, FeePaid? Using the rule that students must take all exam grades in numerical order, the field ExaminationGrade could be removed; as long as students' current grade is known, their exam grade can be worked out.

5 The list of field headings should be close to those shown in the data dictionary below.

6 Display **8.5 Presentation** slide 3 to allow students to see a model version of the music school database. Use this to discuss potential changes to the database structure. Examine the student data dictionary and discuss any differences between students' answers and the model answer provided.

Field name	Data type	Field length
LessonID (primary key field)	Autonumber	n/a
StudentInitial	Short text	1
StudentSurname	Short text	25
YearGroup	Number	n/a
Instrument	Short text	20
CurrentGrade	Short text	1
SessionDate	Date/time	n/a
SessionTime	Data/time	n/a
Lesson/Exam	Short text	6
TutorInitial	Short text	1
TutorSurname	Short text	25
Online/InPerson	Short text	8
FeePayable	Currency	n/a
FeePaid	Y/N (or short text)	n/a (or 1)

Part 2

- A primary key helps distinguish one record from another. For example, in the database shown in this case, each student will appear more than once as they have many lessons at many different times. Using a key field, for example SessionID, helps distinguish one session (lesson or exam session) from another.
- Names were broken down into initials and surnames to ensure that the same data is recorded for each person in the database. If the field heading was 'name', there are lots of ways people might respond: some may only provide their first name, some may provide a full first name and surname, some may provide an initial and a surname. This might make it difficult to search for a person using their name as it may not be entered in the same format each time.
- The field headings were renamed because it was important to remove blank fields when the user was entering data into the database. Even when no data is added to a field, the database still sets aside a number of characters to record that data item. This means that storage space is being wasted. The field headings were renamed from ExaminationTime and LessonTime to SessionDate/SessionTime to help reduce the number of blank fields. This will also make the data validation process easier later in development (we could include a presence check for each field – no field should be left blank). We also added a primary key to make it easy to identify one

lesson from another – we can now use SessionID instead of having to use the student’s name. Examination grade is no longer required. If we are recording the student’s current grade, we can easily work out what their exam grade will be. Also, by breaking student name and tutor name down into student initial and student surname and tutor initial and tutor surname, we are providing a consistent means of recording data which again helps with validation and also makes it easier to search for records that relate to individual students or tutors.

- The structure the music department used previously would have been easy to understand as the field headings were clear, but as more and more data was entered, the blank fields in the database may have been confusing. Users may not have been sure if a field was left blank on purpose or by accident.
- It would have been easy to make an error when storing details for a lesson or an examination session as users may not have entered a student’s or tutor’s name in the same format each time a record was entered into the database.
- See above comments for discussion points on how the data structure in the new MusicSchool.accdb database is an improvement.

Practise (Student’s Book pages 140–145)

An error message should appear in all tasks if students have correctly added the validation checks outlined in the task.

Computational thinking – algorithms (Student’s Book page 148)

Search algorithms ensure that data in large data models are searched effectively for important data items. They tell the processor the steps to follow when searching for data included in the search criteria.

1

- 1** Make a note of the length of the list.
- 2** Set a pointer to mark the start of the list.
- 3** Compare the search value with the item held at the position in the list that matches the pointer.
- 4** If the two values match, then the item has been located and the search can end.
- 5** Otherwise, increase the pointer value by 1 and repeat the previous two steps.
- 6** Keep repeating these steps until the item is found or the number of the counter = the length of the list.

2 It is a suitable search method because the database is sorted into numerical order by the primary key.

3 In a large database, the search may need to check every single data item to be certain whether the item being searched for is in the list. If the data is at the end of the database, the search will take a long time if the database is large. This would be true whether or not the database had already been sorted.

Practise (Student’s Book page 150)

1 For example:

- Simulations: In the entertainment industry, e.g. rollercoasters, or to model weather-forecasting in meteorology.
- Data models, e.g. databases: Businesses keeping track of customers and their orders/libraries monitoring members and loans.

- Spreadsheets: Monitoring temperature changes/sea levels over time, or keeping track of income and expenditure for a business.
- AR: Allowing people to try new furniture in their home before they buy it.
- VR: In retail with a virtual catwalk to model goods, or body scanning to recommend clothes to suit customers' bodies.

2 Advantages and disadvantages are listed in the 'Learn' box.

For example, when referencing the use of spreadsheets to model income/expenditure for a business:

- Advantages: Formulae can be used to link important data items, e.g. to help the company work out total sales and total expenditure and then profit; the company can easily experiment with the model to see how increasing the price of one product might affect sales of the product and therefore overall profit; the company can ask 'what-if' questions.
- Disadvantage: It is difficult to take into consideration all factors, so the model being used to simulate future increases or decreases in sales and profits might not be accurate.
- Examples such as the use of AR as a tool for trying new products or hairstyles:
- Advantages: The customer can try the new product/hairstyle without spending money, so if they do not like it they are not affected personally; the software can be used to allow them to experiment with other products/hairstyles that they had not previously considered.
- Disadvantages: The image generated may not be accurate (e.g. in terms of scale or colour of hair/product); it may not accurately match the final outcome; the software can be expensive to create and continually needs to be updated – this can be expensive.

3

- Weather-modelling data requirements could include: date, level of rainfall, temperature at different points in the day, time of sunrise, time of sunset, windchill, cloud coverage.
- SatNav journey data requirements could include: starting location, destination, distance between the two points, walking speed, Google Maps data to plot the journey.

Practise (Student's Book page 152)

Part 1

1 a – D3: =(C3/\$C\$2)*100

– I3: =AVERAGE(C3,E3,G3)

– J3: =AVERAGE(D3,F3,H3)

Students do not need to see the formulae at this point; they will be given the opportunity to examine the spreadsheet later in the lesson.

b Relative cell referencing is where the cell reference in a formula is updated when the formula is replicated.

Absolute cell referencing is where the cell reference in a formula is not updated when the formula is replicated (e.g. using \$ in the cell references: \$A\$3).

c The formulae in columns I and J use relative cell referencing; the formulae in cells D3–D9 (also F3–F9 and H3–H9) use absolute cell referencing.

2 His score in Test 2 would increase; his % will increase to 91%.

3 I8, J8 (F10 and F11 would change only if his score increased to greater than 89 or less than 68, so they should not be included at this point).

4 a Circulate among students as they compare the formulae they wrote down at the beginning of the exercise with the actual formulae used in the spreadsheet. Take time to discuss the formulae with those who do not understand the structure of the formulae (especially the use of relative and absolute cell referencing).

b D3, I3, J3, D10 and D11.

Part 2

1 New % = 95%, new Average score = 76, new Average % = 93.

2 a It would increase their Average score to 80 and their Average % to 97%.

b It would increase the value of cell F10 to 100.

3 The spreadsheet would now contain the following values.

	A	B	C	D	E	F	G	H	I	J
1			Test 1		Test 2		Test 3		Average	
2	Student Initial	Student Surname	75	%	90	%	80	%	Score	%
3	S	Kahn	66	88	75	83	77	96	73	89
4	A	Amall	70	93	78	87	80	100	76	93
5	S	Karr	69	92	89	99	58	73	72	88
6	R	Rageb	67	89	78	87	79	99	75	92
7	A	Essan	74	99	68	76	75	94	72	89
8	D	Amer	70	93	78	87	68	85	72	88
9	M	Osman	67	89	88	98	76	95	77	94
10			Max %	99	Max %	99	Max %	100		
11			Min %	88	Min %	76	Min %	73		
12										

Go further (Student’s Book pages 155–157)

Databases

2 Example test plan:

Field name	Validation check/input mask	Testing
All fields	Set all fields to be a required field	Try entering a new record and not adding a StudentInitial, for example
StudentSurname	Input mask →L <????????????????????????????	Try to add a lowercase letter; it should automatically change to an uppercase letter
Instruments	Validation rule: 'Guitar' OR 'Drums' OR 'Piano' OR 'Voice' Validation text: Please enter only Guitar, Drums, Piano or Voice	Try to enter an alternative word or instrument, e.g. 'flute', to ensure that it will not accept the data item Then enter 'Guitar' to ensure it is accepted
CurrentGrade	Validation rule: Between 1 and 8 OR some students may have set this as a text file and will use: '1' OR '2' OR ... OR '8' Validation text: Please enter only 1, 2, 3, 4, 5, 6, 7 or 8	Try entering the number -9, 0 or 10 to ensure that it will not accept the data item Then enter 7 to ensure that it is accepted
Lesson/Exam	Validation rule: 'Lesson' OR 'Exam' Validation text: Please enter only Lesson or Exam	Try entering a Surname instead to ensure that it will not accept the data item Then enter Lesson to ensure that it is accepted
TutorInitial	Validation rule: Len([TutorInitial])=1 Validation text: Only one letter is needed for tutor initial	Try to add more than one letter to see whether the error message appears
FeePayable	Between 15 and 20 Validation text: Please enter only a value between \$15 and \$20	Try entering a value such as 10 to ensure that it will not accept the data item Then enter 15 to ensure that it is accepted

Final project (Student's Book pages 159–161)

Part 1

1 Example data dictionary:

Field name	Data type	Field length	Validation check/input mask
BookingID (primary key field)	Autonumber	n/a	
StudentInitial	Short Text	1	>L
StudentSurname	Short Text	25	>L<????????????????????????????
Stage	Number	n/a	Between 8 and 11
Day 1 (NumTickets)	Number	n/a	Between 0 and 4
Day 2 (NumTickets)	Number	n/a	Between 0 and 4
Day 3 (NumTickets)	Number	n/a	Between 0 and 4
ContactInitial	Short Text	1	>L
ContactSurname	Short Text	25	>L<????????????????????????????
TelephoneNumber	Short Text	11	"01"0000000000
TicketsCollected	Y/N	n/a	n/a
DateCollected	Date/Time	n/a	<=Date()

3 For example:

Festival Tickets Booking Form	
Student details	
Initial:	<input type="checkbox"/>
Surname:	<input style="width: 150px;" type="text"/>
Stage:	<input type="checkbox"/> (Please enter a value between 7 and 11)
Number of tickets	
Day 1:	<input type="checkbox"/> } <input type="checkbox"/> } (Maximum 4 tickets per day) <input type="checkbox"/> }
Day 2:	
Day 3:	
Contact details	
Initial:	<input type="checkbox"/>
Surname:	<input style="width: 150px;" type="text"/>
Telephone:	<input style="width: 150px;" type="text"/>
Office use only	
Tickets collected	Y <input type="checkbox"/> / N <input type="checkbox"/>
Date collected (dd/mm/yyyy)	<input style="width: 80px;" type="text"/>

Part 2

3 The discount to date is \$236.25; the maximum number of lessons the school could offer without exceeding their \$300 budget is 44.

Part 3

1 For example:

- A data model is a computer-based representation of a real-life scenario as it exists at the current time. Formulae and algorithms can be used to create relationships between data items in the model.
- A computer simulation uses a data model to imitate a real-life scenario. Inputs can be amended to allow users to investigate how changes in input can alter output from the simulation. Some simulations use specialised input and output devices to support user interaction and may incorporate high-quality graphics to create a realistic environment for the user.

2 For example:

- Using AR in science to allow students to investigate inside the body or inside a beehive; VR could allow them to visit and interact with far-off planets in space.
- Using AR in language learning to allow students to experiment with new words and phrases by interacting with apps in their classroom or other environment; using AR, overlays for words and phrases for items in their current environment could be displayed on screen for students to repeat or click on to hear.

8.6 Iterating through a solution: Software design and development

Practise (Student's Book page 168)

1 Each student acting as the developer should be able to describe how they have developed the design. They should have sketches of each iteration to show how they have developed the design after each round of feedback from the client.

2 For example: The iterative process differs from the waterfall process by its approach to the project – the iterative process develops in cycles and can be followed continuously until the final product meets the project requirements.

Practise (Student's Book pages 170–171)

Part 1

1–3 The infographic should clearly show that the process is iterative; using an arrow that circles back to where it starts can help to illustrate an iteration.

Part 2

1 Features of the app could include storing the details of a new contact, searching for someone, updating a record, a way to collect a new contact, e.g. a form to complete, and instructions for the user.

2 Each pair will have their own interpretation of how to design the app page. Remind students to focus on the page requirements of displaying contact details. Areas that could be considered are the positioning of the contact details on the page, what information will be displayed and how will it look.

3–4 The benefits of decomposing a larger problem into smaller problems are that you are able to refine areas of focus by working on each smaller problem before putting them together to solve the larger problem.

Practise (Student's Book page 173)

1–4 Students may have different answers to this section depending on the direction they take for the questions, e.g. students focusing on recycling may have the following questions.

- Easy question: 'What are the three Rs of waste management: reduce, reuse, recycle or refuse, reduce, recycle?' Answer: reduce, reuse, recycle.
- Hard question: 'How long does it take for a plastic bottle to decompose: 200 or 450 years?' Answer: 450 years.

5–6 Feedback may include:

- The question does not let me know what format the answer should be in; having a set of options could help the user to enter the answer in the correct format.
- The question is not clear; reword it so that the user will understand it better.
- I really like the questions but would suggest swapping the order so they lead nicely into each other.
- This question is too hard or too easy for this set.

7 Advantages may include:

- supporting each other
- identifying and fixing issues
- saving time
- adding improvements.

8 For example: Focusing on smaller parts can allow you to refine one section before moving on to the next section. It is better to plan and check each question before writing the code as errors and improvements are easier to spot.

Practise (Student's Book pages 179–180)

1 This creates a variable called 'score' with a starting value of 0. It asks the user two questions. Each question has two answers. If the user answers a question correctly, then they are told that it is correct and 1 point is added to their score; otherwise, they are told that the answer is incorrect and then the current score is displayed.

2 a

- User 1:
 - Correct
 - 1
 - Incorrect
 - 1
- User 2:
 - Incorrect
 - 0
 - Incorrect
 - 0
- User 3:
 - Correct
 - 1
 - Correct
 - 2

b

- User 1: 1
- User 2: 0
- User 3: 2

3

- score: integer
- question1: string
- question2: string

Practise (Student's Book page 183)

1–4 For example:

Variable	What the variable is used for	Data type	How will it change?
question1	Stores the user's answer to question 1	string	question1 = INPUT
question2	Stores the user's answer to question 2	integer	question2 = INPUT

5–7 Each student's list will have similarities, e.g. the user's answer to the question. Differences may include the variable names, e.g. 'question1' or 'Q1'. There may be additional variables that some students add, including a score that increases or decreases depending on the user's answers, or additional variables to store the correct answers. The comparison allows students to compare their findings and then develop their own tables, which should include the following.

Variable	What the variable is used for	Data type	How will it change?
question1	Stores the user's answer to question 1	string	question1 = INPUT
question2	Stores the user's answer to question 2	string	question2 = INPUT
score	Stores the score as a number starting at 0 and increasing when an answer is correct	integer	IF answer is correct THEN score increases by 1
question1Answer	Stores the correct answer	string	IF question1 == question1Answer THEN score increases by 1
question2Answer	Stores the correct answer	integer	IF question2 == question2Answer THEN score increases by 1

8 Students have developed the list iteratively as they created the first list individually and then, through discussion and comparison with others, they gained feedback on areas to improve or add, which they added into a new list.

Practise (Student's Book page 185)

1 Students' program code will differ based on the question they use.

Students may combine the question into the input statement in the Python code.

2 Students may have used one or more Boolean operators to allow the user to enter their answer in a variety of formats, e.g. 'Y' OR 'y' OR 'YES' OR 'Yes'. Alternatively, they may have asked the user to enter a number to indicate their chosen answer, e.g. 1 for 200 years or 2 for 450 years.

3–4 The variables and data types will vary depending on the question asked; the example above uses:

– question1 – string

– score – integer.

5 The code has been developed by planning and designing an algorithm. This has been turned into program code one line at a time. Some students may have tried to run their program after they added each new section of code to make sure that there were no obvious errors.

Practise (Student's Book page 186)

1 Students should realise that new information will need to be added to the test plan each time a new question is added to the program, to make sure that all routes through the program work correctly.

2–3 The test plan that students create will link to the question they are asking in their quiz, and so these will all be different. Students need to consider how a user will answer each question and use this to generate the test data for each section of the program.

Practise (Student's Book page 188)

1–4 Areas for improvement will vary from student to student, depending on how successfully they wrote their program code. Some students may have programs that work correctly already, but they should consider whether there are any improvements that could be made to the display of the text on the screen.

5 The requirements that have been met will vary from student to student, depending on what they included in their original list. However, they should all have a program that asks

one question, allows the player to enter their answer, gives feedback based on the answer given, and awards points if the answer is correct. The question should be suitable for an eco quiz.

6 Testing may take place during the development of the program code and will always take place once the section of code for each iteration has been completed. Reviewing and evaluating take place at the end of each iteration or cycle.

Practise (Student's Book page 189)

1–2 The list of requirements would include:

- a clear question about recycling or the environment
- an output stating whether the answer is correct or incorrect
- the score increasing if the second question is answered correctly
- the second question playing after the user has answered the first question
- the score being displayed to the user.

4 Possible test plan example linked to:

Test	Test data	Expected outcome	Pass/fail
1	Early Morning	Correct Score increases by 1 Score displayed	Fail
2	Early morning	Correct Score increases by 1 Score displayed	Pass
3	early morning	Correct Score increases by 1 Score displayed	Pass
4	Early Afternoon	Incorrect Score displayed	Pass
5	Early afternoon	Incorrect Score displayed	Pass
6	early afternoon	Incorrect Score displayed	Pass

5–6 Feedback will be linked to students' program code. For an example linked to the test plan shown above in answer 4, the feedback would be that the answer was still added in the first test and the output was incorrect as there were uppercase letters on both words.

8 Test plan example:

Test	Test data	Expected outcome	Pass/fail
1	Early Morning	Correct Score increases by 1 Score displayed	Pass
2	Early morning	Correct Score increases by 1 Score displayed	Pass
3	early morning	Correct Score increases by 1 Score displayed	Pass
4	Early Afternoon	Incorrect Score displayed	Pass
5	Early afternoon	Incorrect Score displayed	Pass
6	early afternoon	Incorrect Score displayed	Pass

9 For example:

- a clear question that asks about recycling or the environment
- an output stating whether the answer is correct or incorrect
- the score increasing if the second question is answered correctly
- the second question playing after the user has answered the first question
- the score being displayed to the user.

Go further (Student's Book pages 189–190)

1 The data type can be either an integer or a string as no comparisons or calculations are being carried out with the numbers. The difference in the Python code is the use of quotation marks around the number for a string and without for an integer.

2 Testing will involve running the program and observing whether the output has a delay between each of the number outputs. Students could use a stopwatch to further test the time delay. Here is an example test plan.

Test	Test data	Expected outcome	Pass/fail
1	Start	OUTPUT "Countdown 3 to 1" Delay 1second OUTPUT 3 Delay 1second OUTPUT 2 Delay 1second OUTPUT 1 Delay 1second OUTPUT "GO"	Pass

The program could be developed by adding selection, and then asking the user whether they are ready before the countdown starts, e.g. **if** user is ready, **then** start the countdown.

Challenge yourself (Student's Book pages 190–191)

1 Students will need to create the Python code to generate a random number between 1 and 100.

2 For example, variables used:

- 'storedNumber' = integer
- 'numberGuessed' = integer.

3 For example:

- **if numberGuessed == storedNumber:**

```
print("You guessed")
print(storedNumber)
```

4 For example:

- **elif numberGuessed > storedNumber and numberGuessed < 100:**

```
print("Lower")
print(storedNumber)
```

5 For example:

- **elif numberGuessed < storedNumber and numberGuessed > 0:**

```
print("Higher")
print(storedNumber)
```

6 A test plan is harder to generate with a random number, as the expected outcome is not known because you do not know the randomly generated number. Documenting the output and the value being stored can show whether the correct output is being displayed, e.g.:

Test	Test data	Expected outcome	Pass/fail
1	Input	Higher or Lower and number that was being stored	Pass
2	Input = 50	Higher 85	Pass
3	Input = 3	Higher 96	Pass
4	Input = 200	Output "Out of guess range"	Pass
5	Input = 99	Lower 10	Pass

Final project (Student's Book pages 191–193)

Plan

1–2 For example:

- questions linked to whether or not an item can be recycled
- quiz containing three questions
- user must answer all three questions
- after the three questions are answered, the user is told whether the item can be recycled the questions need to flow into each other and use a delay to allow the question and answers to be displayed.

Design

3 Important characteristics of pseudocode include:

- it is short, clear and precise

- OUTPUT and INPUT specified and in uppercase
- **if, then, else** and **endif** used for selection
- indentation used for code to run if the condition is TRUE or FALSE
- quotation marks around text to be output
- START and STOP used.

4 If none of the conditions are met, then the output will be ‘Your item cannot be recycled’.

5–6 Questions will differ between students, depending on how they determine whether an item can be recycled; one set of questions has been used in the following example.

Question	Answer if suitable for recycling	Conditional statement to be used in program	Reason for using any Boolean operator
Is your item made from any of the following materials? 1 Glass 2 Plastic 3 Aluminium 4 Paper 5 Cardboard	Your item can be recycled	If item1 == "1" or item1 == "2" or item1 == "3" or item1 == "4" or item1 == "5"	OR used to accept alternative options
Does the item have a recycling logo on it?	Your item can be recycled	If item2 == "Yes" or item2 == "Y" or item2 == "yes"	OR used to accept alternative options
Is your item a battery?	Your item can be recycled	If item3 == "Yes" or item3 == "Y" or item3 == "yes"	OR used to accept alternative options

7 Variables will be individual but need to be suitably named, e.g. ‘q1Answer’.

Variable	Data type
items	Integer
q1Answer	String
q2Answer	String
q3Answer	String

Test/Review and evaluate

10–13 Students should create a test plan with a suitable data range. The data range will consider the different inputs that the user can enter, e.g. ‘Y’ or ‘y’ should be tested for each question as well as other answers, e.g.:

Test	Test data	Expected outcome	Pass/fail
1	Y	The item may be able to be recycled.	Pass
2	y	The item may be able to be recycled.	Pass
3	N	We need to know more.	Pass
4	n	We need to know more.	Pass

Iteration 2

14–16 For example, here is a test plan to test question 2.

Test	Test data	Expected outcome	Pass/fail
1	Y	Clean the item fully.	Pass
2	y	Clean the item fully.	Pass
3	N	We need to know more.	Pass
4	n	We need to know more.	Pass

Iteration 3

17–19 For example, here is a test plan to test question 3.

Test	Test data	Expected outcome	Pass/fail
1	Y	Yes, it can be recycled.	Pass
2	y	Yes, it can be recycled.	Pass
3	N	The item cannot be recycled.	Pass
4	n	The item cannot be recycled.	Pass

Iteration 4

20–24 For example, question 1:

Test	Test data	Expected outcome	Pass/fail
1	Question 1 = Y	The item may be able to be recycled. Delay 1second	Pass
2	Question 1 = y	The item may be able to be recycled. Delay 1second	Pass
3	Question 1 = N	We need to know more. Delay 1second	Pass
4	Question 1 = n	We need to know more. Delay 1second	Pass
5	Question 2 = Y	Clean the item fully. Delay 1second	Pass
6	Question 2 = y	Clean the item fully. Delay 1second	Pass
7	Question 2 = N	We need to know more. Delay 1second	Pass
8	Question 2 = n	We need to know more. Delay 1second	Pass
9	Question 3 = Y	Yes, it can be recycled.	Pass
10	Question 3 = y	Yes, it can be recycled.	Pass
11	Question 3 = N	The item cannot be recycled.	Pass
12	Question 3 = n	The item cannot be recycled.	Pass