Answers

Chapter 1: Problem Solving

1.1 Algorithms

Understanding algorithms

Activity 1

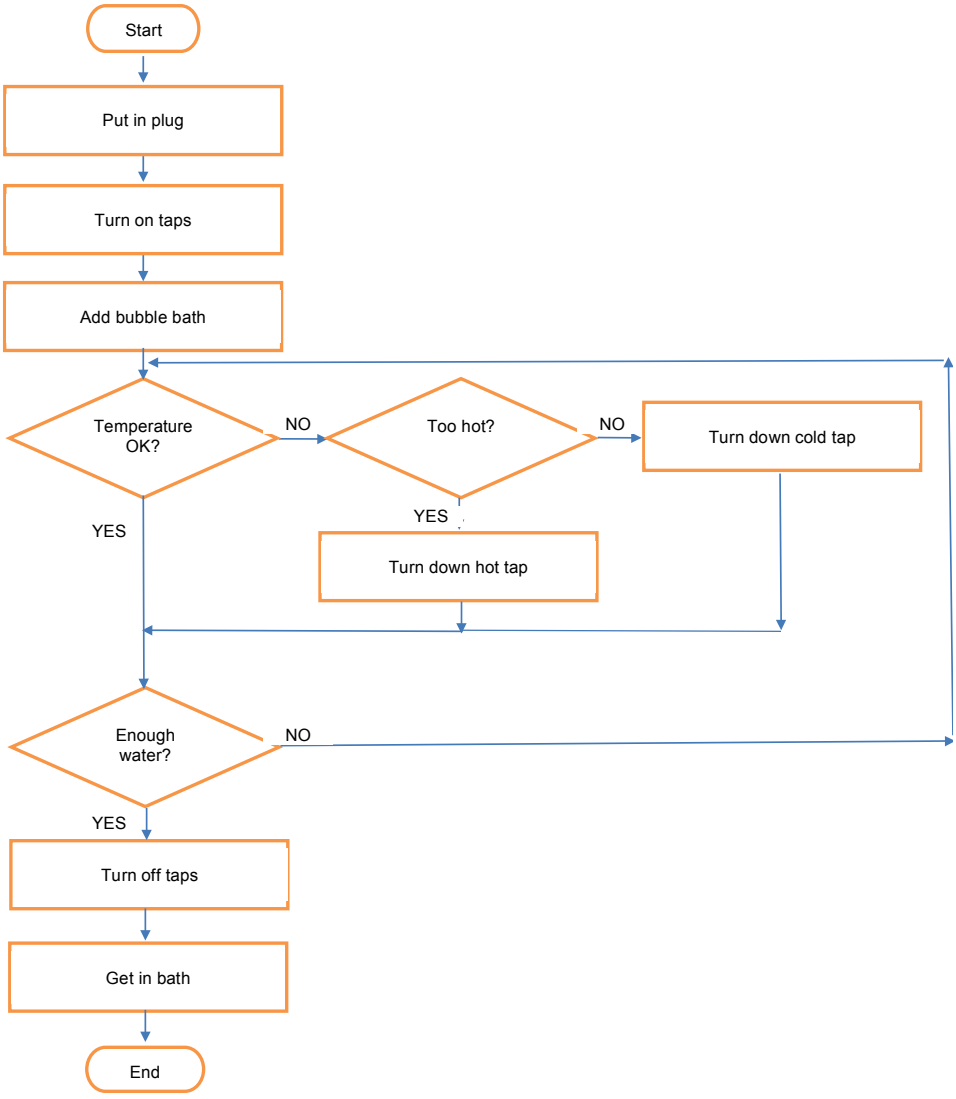
There is no single correct solution to this activity, since every student's walk to school will be unique. It is intended to highlight the need for instructions to be specific, precise and in a logical order. Encourage them to identify instances of repetition and selection, e.g. 'Keep walking until you get to the end of the street. Next cross the road, providing no cars are coming.'

Activity 2

Encourage students to use all the symbols shown in Figure 1.1 in their flowchart.

Activity 3

This flowchart incorporates selection and iteration. Students may simply identify a sequence of processes in their solution.



Activity 4

Students must be familiar with Edexcel pseudo-code, which will be used to frame some of the exam questions on Paper 2, but they don't have to use it themselves. They can use any pseudo-code, providing it is unambiguous and easy to follow.

Activity 5

```
SEND 'Enter first number.' TO DISPLAY
RECEIVE firstNumb FROM KEYBOARD
SEND 'Enter second number.' TO DISPLAY
RECEIVE secondNumb FROM KEYBOARD
SET thirdNumb TO firstNumb * secondNumb
SEND thirdNumb TO DISPLAY
```

Activity 6

Ask the user to enter their username. Repeat until an existing username is entered. Next ask the user to enter their password. Repeat until the correct password is entered.

Checkpoint S1

The specifics of this algorithm should reflect the procedure for borrowing books at the student's own local library or school library.

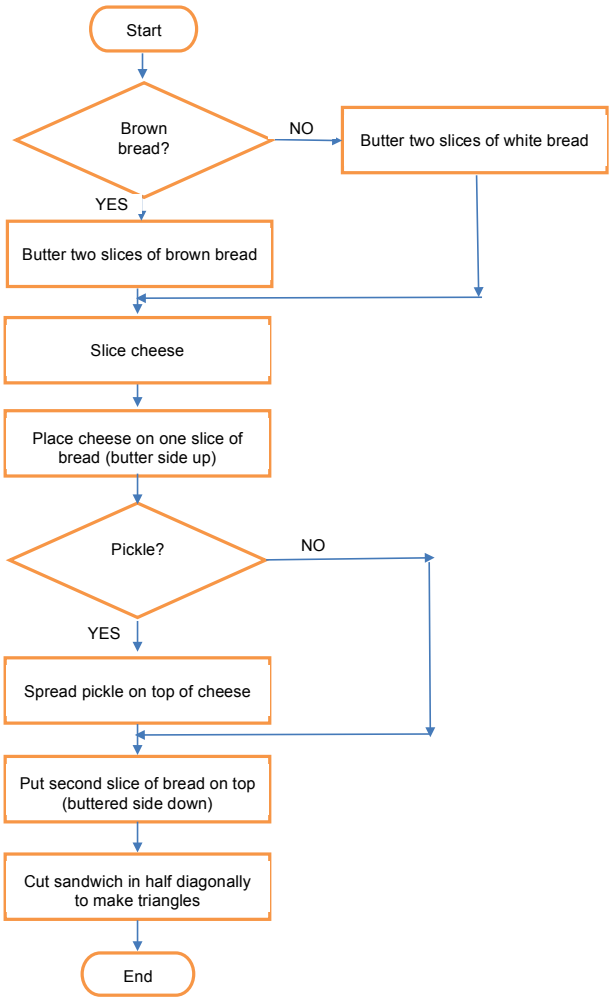
Checkpoint S2

The table on page 5 describes the function of each of the arithmetic operators.

Checkpoint S3 and S4

Variables and constants are explained on pages 5 & 6.

Checkpoint C1



Checkpoint C2

```
SEND 'Enter first number.' TO DISPLAY
RECEIVE firstNumb FROM KEYBOARD
SEND 'Enter second number.' TO DISPLAY
RECEIVE secondNumb FROM KEYBOARD
SEND 'Enter third number.' TO DISPLAY
RECEIVE thirdNumb FROM KEYBOARD
SET average TO (firstNumb + secondNumb + thirdNumb)/3
SEND average TO DISPLAY
```

Creating algorithms

Activity 7

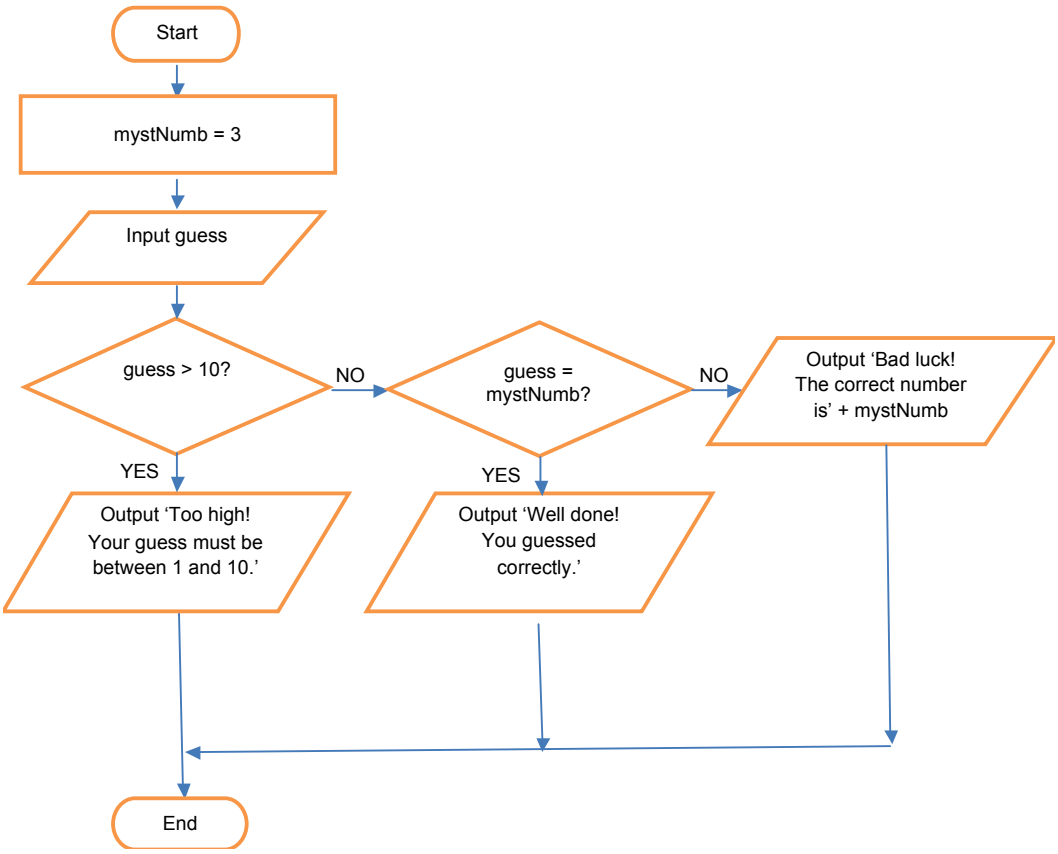
score	highScore	OUTPUT
5	10	You haven't beaten your high score.
20	10	You've exceeded your high score!
15	15	You haven't beaten your high score.

Activity 8

```
SEND 'Please enter your age.' TO DISPLAY
RECEIVE age FROM KEYBOARD

IF age <= 18 THEN
    SET numbLessons TO 20
ELSE
    SET numbLessons TO 20 + (age - 18) * 2
END IF

SEND 'You need' & numbLessons & 'driving lessons.' TO DISPLAY
#Here the '&' symbol is used to join data together. This will be explained on
page 18 of the Student Book.
```

Activity 9

Here's the algorithm expressed as a flowchart.



Here's the same algorithm expressed as pseudo-code.

```
SET mystNumb TO 3
SEND 'Please enter a number between 1 and 10.' TO DISPLAY
RECEIVE guess FROM KEYBOARD

IF guess > 10 THEN
    SEND 'Too high! Your guess must be between 1 and 10.' TO DISPLAY
ELSE
    IF guess = mystNumb THEN
        SEND 'Well done! You have guessed correctly.' TO DISPLAY
    ELSE
        SEND 'Bad luck! The correct number is' & mystNumb TO DISPLAY
    END IF
END IF
```

Activity 10

score	OUTPUT
91	A
56	D
78	B

Activity 11

Using a WHILE loop

```
SEND 'Please enter a number between 1 and 6.' TO DISPLAY
RECEIVE numb FROM KEYBOARD #User enters their first guess

WHILE numb <> diceRoll DO  #Loop repeats until user enters the correct number
    SEND 'Please enter a number between 1 and 6.' TO DISPLAY
    RECEIVE numb FROM KEYBOARD
END WHILE  #Loop terminator

SEND 'Well done, you've guessed correctly.' TO DISPLAY
```

Using a REPEAT loop

```
REPEAT #Start of loop
    SEND 'Please enter a number between 1 and 6.' TO DISPLAY
    RECEIVE numb FROM KEYBOARD
UNTIL numb = diceRoll      #Loop is repeated until user enters the correct
number

SEND 'Well done, you've guessed correctly.' TO DISPLAY
```

Activity 12

```
SEND 'Enter a start number.' TO DISPLAY
RECEIVE startNumb FROM KEYBOARD
SEND 'Enter an end number.' TO DISPLAY
```

```
RECEIVE endNumb FROM KEYBOARD
SET total TO 0

FOR index FROM startNumb TO endNumb DO
    SET total TO total + index
END FOR

SEND total TO DISPLAY
```

Activity 13

```
SET mysteryNumb TO RANDOM(19) + 1
#Generates a random number between 1 and 20
SET failedAttempts TO 0    #Keeps track of number of wrong guesses

REPEAT

    SEND 'Please enter a number between 1 and 20.' TO DISPLAY
    RECEIVE guess FROM KEYBOARD
    IF guess = mysteryNumb THEN

        SEND 'Congratulations. You've guessed correctly.' TO DISPLAY

    ELSE

        SET failedAttempts TO failedAttempts + 1

        IF guess < mysteryNumb THEN
            SEND 'Your guess was too low.' TO DISPLAY
        ELSE
            SEND 'Your guess was too high.' TO DISPLAY
        END IF


    END IF

UNTIL guess = mysteryNumb OR failedAttempts = 3

IF failedAttempts = 3 THEN
    SEND 'Bad luck! The mystery number was' & mysteryNumb TO DISPLAY
END IF
```

Activity 14

```
FOR index FROM 2 TO 12 DO
    FOR times FROM 1 TO 12 DO
        SEND times & 'x' & index & '=' & times * index TO DISPLAY
    END FOR
END FOR
```

Checkpoint S1

The activities in this chapter provide plenty of examples that students can use to illustrate their answer.

Checkpoint S2

The table on page 10 lists the six relational operators.

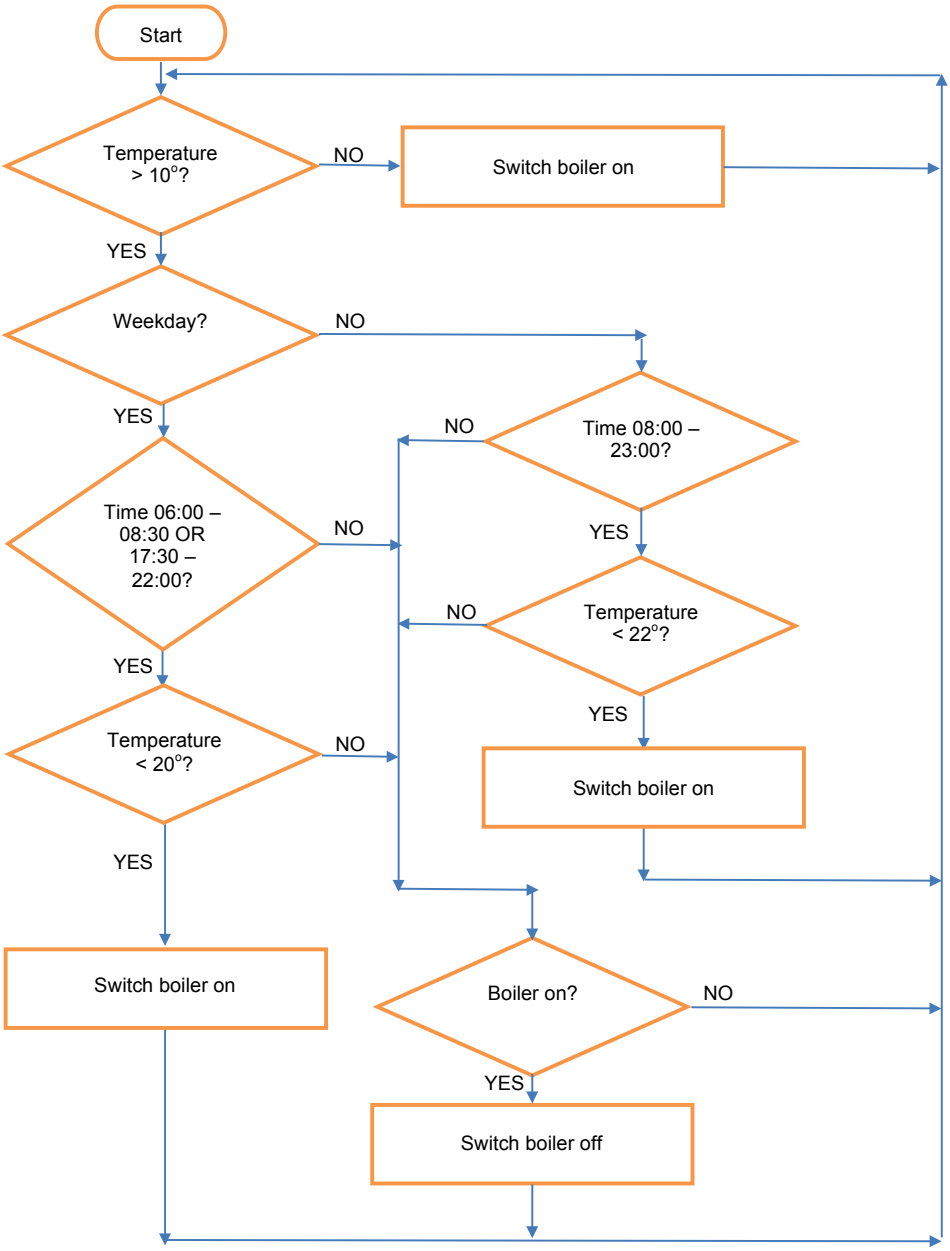
Checkpoint C1

```
SEND 'Enter your height (in metres).' TO DISPLAY
RECEIVE height FROM KEYBOARD
SEND 'Enter your weight (in kilograms).' TO DISPLAY
RECEIVE weight FROM KEYBOARD
SEND 'Your body mass index (BMI) is:' & weight/height^2 TO DISPLAY
```

Reminder: "height^2" means "height to the power of 2", i.e. "height^2". See "Arithmetic operators" on page 5.

Checkpoint C2

Encourage students to devise appropriate test data to check that the algorithm produces the right outcomes.



Working with algorithms

Activity 15

scores	win	loss	draw	totalFor	totalAgainst
1 – 0	1	0	0	1	0
3 – 2	2	0	0	4	2
0 – 2	2	1	0	4	4
1 – 1	2	1	1	5	5

Activity 16

index	number1	number2
	2	3
1	2	5
2	4	9
3	12	21
4	48	69
5	240	309

Exam-style question

1 The regular charge is £10, but children under £13 pay £5 and people aged 60 or above pay £9. A group consisting of five or more adults gets a discount of £10.

2 The variables used in the flowchart are: charge, total, number, customer, age.

3 Label A denotes a decision to be made and B denotes a process to be carried out.

4 (2 × £5) + (2 × £10) + £9 = £39 (They are not eligible for a group discount: there are five people, but the variable 'number' comes to 3, not 5. Why is that?)

Activity 17

The maximum car park charge is £20. (parkCharge is a random number between 1 and 20.)

The amount of change due is calculated by subtracting the parking charge from the payment tendered.

The algorithm uses a series of WHILE loops to work out how many notes/coins of each denomination should be given, starting with £10 notes. Each time a note/coin is given in change, its value is deducted from `changeDue`.

Here's the completed algorithm.

```
SET parkCharge TO RANDOM(20)
SET payment TO 0

WHILE payment < parkCharge OR payment > 20 DO
    SEND 'The charge is ' & parkCharge TO DISPLAY
    SEND 'Please enter payment up to £20 maximum.' TO DISPLAY
    RECEIVE payment FROM KEYBOARD
    SET changeDue TO payment - parkCharge
    #Calculates amount of change due

    #Loop continues paying out £10 notes until changeDue is less than 10
    WHILE changeDue >= 10 DO
        SEND '£10 note' TO DISPLAY
        SET changeDue TO changeDue - 10
    END WHILE
```

```
    #Loop continues paying out £5 notes until changeDue is less than 5
    WHILE changeDue >= 5 DO
        SEND '£5 note' TO DISPLAY
        SET changeDue TO changeDue - 5
    END WHILE
    #Loop continues paying out £2 coins until changeDue is less than 2
    WHILE changeDue >= 2 DO
        SEND '£2 coin' TO DISPLAY
        SET changeDue TO changeDue - 2
    END WHILE
    #Loop continues paying out £1 coins until changeDue is less than 1
    WHILE changeDue >= 1 DO
        SEND '£1 coin' TO DISPLAY
        SET changeDue TO changeDue - 1
    END WHILE
    #Loop continues paying out 50p coins until changeDue is less than 0.5
    WHILE changeDue >= 0.5 DO
        SEND '50p coin' TO DISPLAY
        SET changeDue TO changeDue - 0.5
    END WHILE
    #Loop continues paying out 20p coins until changeDue is less than 0.2
    WHILE changeDue >= 0.2 DO
        SEND '20p coin' TO DISPLAY
        SET changeDue TO changeDue - 0.2
    END WHILE
    #Loop continues paying out 10p coins until changeDue is less than 0.1
    WHILE changeDue >= 0.1 DO
        SEND '10p coin' TO DISPLAY
        SET changeDue TO changeDue - 0.1
    END WHILE
    #Loop continues paying out 5p coins until changeDue is less than 0.05
    WHILE changeDue >= 0.05 DO
        SEND '5p coin' TO DISPLAY
        SET changeDue TO changeDue - 0.05
    END WHILE
    #Loop continues paying out 2p coins until changeDue is less than 0.02
    WHILE changeDue >= 0.02 DO
        SEND '2p coin' TO DISPLAY
        SET changeDue TO changeDue - 0.02
    END WHILE
    #Loop continues paying out 1p coins until changeDue is less than 0.0
    WHILE changeDue >= 0.01 DO
        SEND '1p coin' TO DISPLAY
        SET changeDue TO changeDue - 0.01
    END WHILE

END WHILE
```

If the charge is £6.90 and the person pays with a £20 note, they are owed £13.10 change and will receive one £10 note, one £2 coin, one £1 coin and one 10p coin.

Activity 18

The developer used an OR instead of an AND. This means that any number greater or equal to 7 (e.g. 99) and any number less than or equal to 13 (e.g. –2) would be accepted.

Activity 19

Example 1

```
SET index TO 1
WHILE index < 10 DO
    SEND index TO DISPLAY
    SET index TO index + 1
    #Index must be incremented in the loop so that the terminating condition is
    eventually met
END WHILE
```

Example 2

```
SET index TO 1
WHILE index < 10 DO
    SEND index TO DISPLAY
    SET index TO index + 1
#The variable index must increase each time through the loop, not decrease
END WHILE
```

Example 3

```
SET index TO -5 #or any number less than 1
#The value of index must be less than 1 for the loop to execute
WHILE index < 1 DO
    SEND INDEX TO DISPLAY
    SET index TO index + 1
END WHILE
```

Checkpoint S1

Students should select an algorithm they have written themselves and/or one written by someone else and explain how it works.

Checkpoint S2

Logic errors are explained on pages 25 – 27. Encourage students to find examples of logic errors they have made themselves.

Checkpoint S3

Students should select an algorithm they have written themselves and/or one written by someone else. BIDMAS means Brackets, Indices, Division and Multiplication, Addition and Subtraction. The order of evaluation of the expression $4^3 \times 10 / 2 + (8 – 3)$ is therefore

Brackets:	$(8 – 3)$	$= 5$
Indices:	4^3	$= 64$
Division and Multiplication:	$10 / 2$	$= 5$

So the expression simplifies to $64 \times 5 + 5$

Checkpoint C1

The purpose of the algorithm is to work out the charge for sending a parcel.

There are three weight categories, each with its own charge.

The charge for sending parcels weighing 2 kilograms or less is £8.

The charge for parcels weighing between 2 and 10 kilograms is £8 plus £2.50 times the weight of the parcel minus 2 kilograms. So a 10 kilogram parcel would cost £28.

The charge for parcels weighing more than 10 kilograms is £28 plus £3.50 times the weight of the parcel minus 10 kilograms.

Checkpoint C2

```
SET anotherGo TO 'y'
#Loop allows user to have more than one go
WHILE anotherGo = 'y' OR anotherGo = 'Y' DO
    SEND ''Enter 'c' for cat or 'd' for dog.'' TO DISPLAY
    RECEIVE pet FROM KEYBOARD
    SEND 'How old is your pet?' TO DISPLAY
    RECEIVE petAge FROM KEYBOARD
    IF petAge = 1 THEN
    #First we look at the case of one-year old pets. One-year old cats and dogs
    have different human equivalent ages.
        #Nested IF
        IF pet = 'C' OR PET = 'c' THEN
            #Calculates human equivalent age of a one-year old CAT
            SET humanAge TO 15
        ELSE
            #Calculates human equivalent age of a one-year old DOG
            SET humanAge TO 12
        END IF

    ELSE
    #Next we look at the other ages, where cats and dogs have the same human
    equivalent ages.
        #Nested IF
        IF petAge = 2 THEN
            SET humanAge TO 24
        ELSE
            SET humanAge TO 24 + (petAge - 2) * 4
        END IF
    END IF

    SEND 'The human equivalent age of your pet is' & humanAge
    SEND 'Do you want to use the calculator again (y/n)?' TO DISPLAY
    RECEIVE anotherGo FROM KEYBOARD
END WHILE
```

This algorithm doesn't deal with entries other than 'c', 'C', 'd' and 'D'. You could challenge students to amend their version so that entry of any other letter gets an 'invalid entry' response.

Sorting and searching algorithms

Activity 20

```
SET total TO 0
SET highest TO 0
SET numbMarks TO LENGTH(arrayScores)

FOR index = 0 TO numbMarks - 1 DO
    SET total TO total + arrayScores[index]
    IF arrayScores[index] > highest THEN
        SET highest TO arrayScores[index]
    END IF
END FOR

SET averageMark TO total/numbMarks
SEND 'Her highest mark is' & highest TO DISPLAY
SEND 'Her average mark is' & averageMark TO DISPLAY
```

Activity 21

The variable `length` is used to store the length of the list. The variable `switch` is initially set to 0.

Starting at the beginning of the list, successive pairs of items are compared and swapped round if the first item is bigger than the second item. When a swap occurs the value of `switch` changes from 0 to 1. This process is repeated until the end of the list is reached.

If at the end of a pass through the list the value of `switch` hasn't changed from 0 to 1, this indicates that no swaps have taken place, meaning that the list is now sorted. The algorithm then terminates.

Activity 22

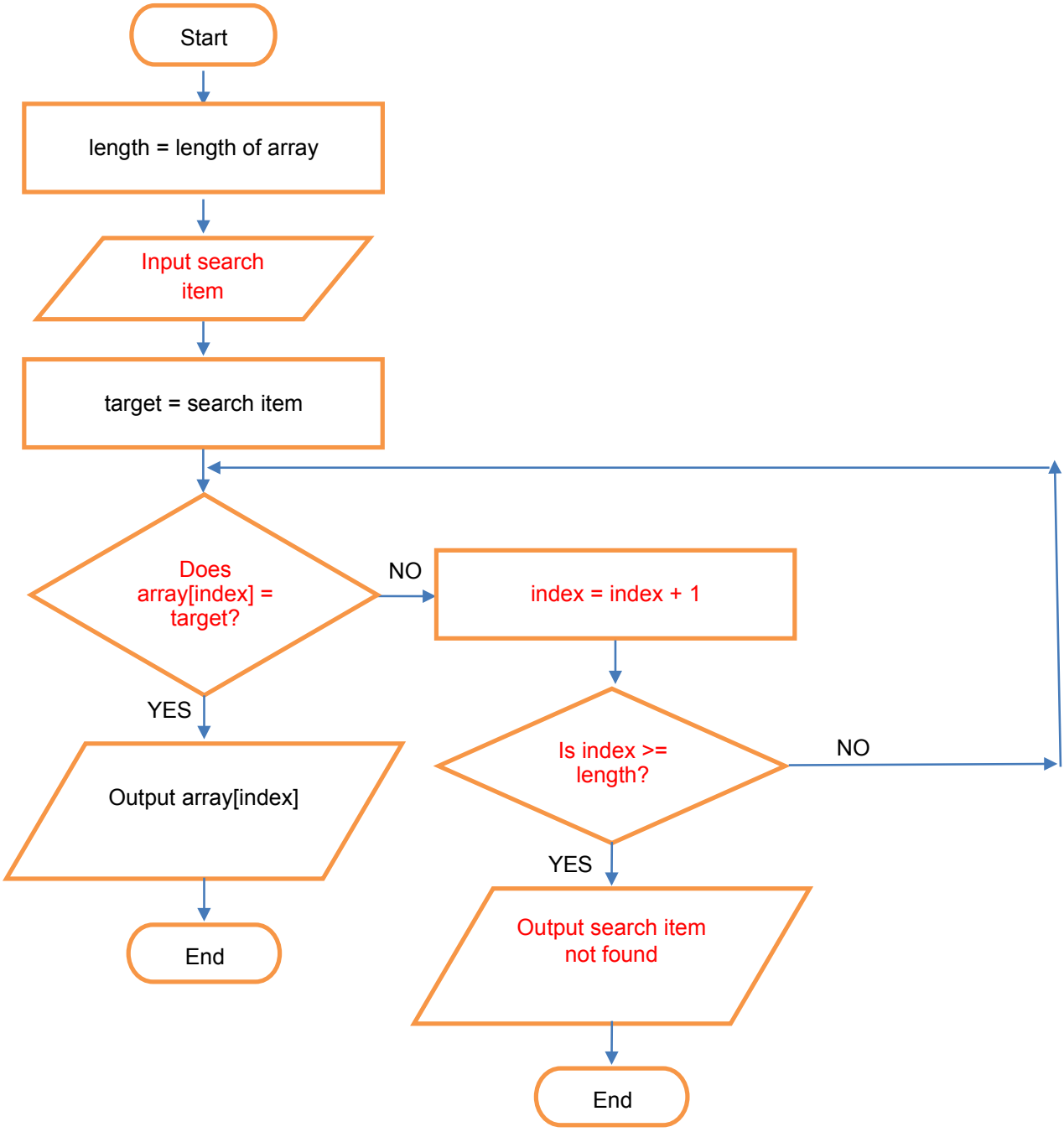
```
SET length TO LENGTH(exam1) - 1
SET swapped TO 1
WHILE swapped = 1 DO
    SET swapped TO 0

    FOR index FROM 1 TO length DO #Sorts items in descending order
        IF exam1[index - 1] < exam1[index] THEN
            SET temp TO exam1[index - 1]
            SET exam1[index - 1] TO exam1[index]
            SET exam1[index] TO temp
            SET swapped TO 1
        END IF
    END FOR

END WHILE
SEND 'The highest result was' & exam1[0] TO DISPLAY
SEND 'The lowest result was' & exam1[length] TO DISPLAY
```

Activity 23

```
[48, 20, 9, 17, 13, 21, 28, 60]
[48, 20, 9, 17] [13, 21, 28, 60]
[48, 20] [9, 17] [13, 21] [28, 60]
```

```
[48] [20] [9] [17] [13] [21] [28] [60]
[48, 20] [17, 9] [21, 13] [60, 28]
[48, 20, 17, 9] [60, 28, 21, 13]
[60, 48, 28, 21, 20, 17, 13, 9]
```

Activity 24



Activity 25

This algorithm assumes the existence of an array called `topHundred` that contains the names of the top one hundred most downloaded performers.

```
SEND 'Enter the name of the performer.' TO DISPLAY
RECEIVE performer FROM KEYBOARD
SET found TO False
SET length to LENGTH(topHundred) - 1
```

```
SET index TO 0

WHILE index <= length AND found = False DO
    IF topHundred[index] = performer THEN
        SET found to True
    END IF

    SET index TO index + 1
END WHILE

IF found = True THEN
    SEND performer & ' is in the top 100.' TO DISPLAY
ELSE
    SEND performer & ' is not in the top 100.' TO DISPLAY
END IF
```

Activity 26

```
3 9 13 15 21 24 27 30 36 39 42 54 69
3 9 13 15 21 24
3 9 13
13
```

Activity 27

```
SET aList TO [3, 13, 24, 27, 31, 39, 45, 60, 69]
end = aList[middle] - 1
start = aList[middle] + 1
```

Exam-style question

1 Marek, Jackson, Bachchan, Wilson, Abraham, French, Smith

Jackson, Bachchan, Marek, Abraham, French, Smith, Wilson

Bachchan, Jackson, Abraham, French, Marek, Smith, Wilson

Bachchan, Abraham, French, Jackson, Marek, Smith, Wilson

Abraham, Bachchan, French, Jackson, Marek, Smith, Wilson

2 Azikiwe, Bloom, Byrne, Davidson, Gateri, Hinton, Jackson, Linton, Smith, Wall

Median is Hinton; Jackson > Hinton

First half of list discarded. New list: Hinton, Jackson, Linton, Smith, Wall

Median is Linton; Jackson < Linton

Second half of list discarded. New list: Hinton, Jackson

Median is Jackson; search item found

Checkpoint S1

The difference between the bubble sort and merge sort algorithms is described on page 134.

Checkpoint S2

The way in which a binary search algorithm finds the search item is explained on pages 36 – 37.

Checkpoint C1

The efficiency of a linear search as compared to a binary search is described on page 38, which includes mention of when it might be better to use the former rather than the latter.

Checkpoint C2

The advantage of using an array rather than separate variables is briefly explained on page 29.

1.2 Decomposition and abstraction

Activity 28

Inputs:

- When to start a new game.
- Number and names of players.
- When to spin the wheel.
- A player's selected answer.
- Whether players want to play again.

Outputs:

- A message to inform a player when it is their turn.
- A message to inform the player of the outcome of spinning the wheel (question category).
- A question plus four possible answers.
- A message to inform the player whether their answer is correct or incorrect.
- A message to inform the player that they can have another go.
- A message to inform the player how many points they have scored.
- A message at the end of each round to inform each player of their total score.
- A 'game over' message.
- A message at the end of the game informing players who has won.
- A message to ask whether the players want to play another game or want to finish.

Processing requirements:

- Set up question banks for each colour/subject.
- Flag each question as unused.
- Establish how many players there are (up to a maximum of four).
- Set each player's score to 0.
- Repeat until one player reaches a score of at least 30 or there are no more unanswered questions.
- Prompt next player to select a subject and simulate spinning the wheel. Display colour and subject selected.
- Randomly select a question from the remaining unanswered questions in the subject question bank.
- Display the selected question and four possible answers. Prompt player to select an answer.
- Receive player's answer. If correct, increment score by 2. If incorrect, prompt player to have a second go. If second attempt successful, increment score by 1.
- Display an appropriate message.
- Flag the question as used.
- When one of the players reaches a score of 30 or more or the computer runs out of questions stop the game.
- If there is a winner, display name and congratulatory message. If the computer has run out of questions display an appropriate message.
- Check if the players want to play another game or finish.

Subprograms:

- select_category
- display_Q&A
- check_response
- update_score
- mark_asked_questions
- establish_winner

Algorithm to simulate spinning the wheel to select a colour:

- Select a random number between 0 and 3.

- If the number is 0 then display a red square on the screen.
- If the number is 1 then display a blue square on the screen.
- If the number is 2 then display a green square on the screen.
- Otherwise display a yellow square on the screen.

Algorithm in pseudo-code:

```
SET number TO RANDOM(3)

IF number = 0 THEN
    colour = red
ELSE
    IF number = 1 THEN
        colour = blue
    ELSE
        IF number = 2 THEN
            colour = green
        ELSE
            colour = yellow
        END IF
    END IF
END IF

SEND 'The colour selected is:' & colour TO DISPLAY
```

Checkpoint S1

Decomposition is described on page 40.

Checkpoint S2

Abstraction is described on page 41.

Checkpoint C1

Students could use their solution to Activity 28 to illustrate the use of decomposition and abstraction. Alternatively, they may wish to come back to this checkpoint once they have completed Chapter 2.

Checkpoint C2

Computational thinking is defined in the Key terms box on page 40.

Chapter 2: Programming

2.1 Develop code

Note: All program code is written in Python 3.5

Activity 1

Students using a language other than Python could add an extra column to the pseudo-code guide to show constructions in that language. All students could also consider adding new material to the bottom of the table.

Activity 2

1 High level programming languages often provide more than the four basic data types.

2 a Integer

b Real

c Boolean

3 There is no single correct solution to this activity.

Activity 3

There is no single correct solution to this activity.

Activity 4

Variable	Purpose	Data type
totalVisitors	Keeps track of the number of people in the park at any one time. (It is incremented by one each time someone enters and decremented by 1 each time someone leaves.)	integer
visitorType	Set to ‘a’ if the visitor is an adult and ‘c’ if the visitor is a child.	character
Takings	Keeps a running total of the amount of money collected at the gate (£2.50 per adult; no charge for children).	real
parkFull	Set to ‘False’ initially but changes to ‘True’ when the number of visitors reaches 10,000.	Boolean

Activity 5

1 The algorithm takes in two integer numbers entered from the keyboard and displays:

- the result of dividing the first number by the second number;
- the number remaining after dividing the first number by the second number;
- the integer division.

2 a With 4 and 2 as inputs, the outputs would be: 2.0, 0, 2.

b With 10 and 3 as inputs, the outputs would be: 3.33333333333333, 1, 3.

c With 20 and 6 as inputs, the outputs would be: 3.33333333333333, 2, 3.

3 Here is the algorithm implemented in Python.

```
number1 = int(input('Enter first number: '))
number2 = int(input('Enter second number: '))
print('number1 / number2 = ', number1/number2)
print('number1 MOD number2 = ', number1 % number2)
print('number1 DIV number2 = ', number1 // number2)
```

Activity 6

1 No answer required.

2 Here is the algorithm implemented in Python.

```
age = int(input('Please enter your age: '))

if age <= 18:
    numbLessons = 20
else:
    numbLessons = 20 + (age - 18) * 2

print('You need', numbLessons, 'driving lessons.')
```

3 Here is the algorithm implemented in Python.

```
testScore = int(input('Enter test score: '))
if testScore >= 80:
    print('A')
elif testScore >= 70:
    print('B')
elif testScore >= 60:
    print('C')
elif testScore > 0:
    print('D')
else:
    print('FAIL')
```

Activity 7

1 a Algorithm A displays the numbers 1 to 10 cubed, i.e. 1 – 1000.

Here is the algorithm implemented in Python.

```
for index in range(1, 11):
    print (index * index * index)
```

b Algorithm B displays a countdown from 10 to 1.

Here is the algorithm implemented in Python.

```
counter = 10
while counter > 0:
    print(counter)
    counter -= 1
```

2 Here is the algorithm implemented in Python, using a WHILE loop. This program uses the imported random module.

```
import random
diceRoll = random.randint(1,6)
guessAgain = 'y'

while guessAgain == 'y':
    guess = int(input('Guess a number between 1 and 6:'))

    if guess == diceRoll:
        print ('Well done, you've guessed correctly.')
        guessAgain = 'n'
    else:
        print('Bad luck. try again.')
```

Python doesn't have a REPEAT…UNTIL loop. The nearest equivalent is a WHILE…TRUE loop.

```
import random
diceRoll = random.randint(1,6)

while True:
    guess = int(input('Guess a number between 1 and 6: '))

    if guess == diceRoll:
        print ('Well done, you've guessed correctly.')
        break
    else:
        print('Bad luck. try again.')
```

3 Here is the algorithm implemented in Python.

```
startNumb = int(input('Enter the start number: '))
endNumb = int(input('Enter the end number: '))
total = 0
for index in range(startNumb, (endNumb + 1)):
    total += index
print(total)
```

Activity 8

Converting this algorithm into program code is a bit tricky, since calculations involving real numbers (non-integer numbers like 4.25 and π) aren't accurate enough for this purpose. To get round this, the program calculates in integer numbers of pence rather than in pounds.

It uses the imported random module.

```
import random
parkCharge = random.randint(1, 2000)
#Pence used to ensure calculations are accurate
payment = 0

while payment < parkCharge or payment > 2000: #2000 pence = £20
    print('The charge is £', '{:02.2f}'.format(parkCharge/100))
    #"{:02.2f}" is a way of defining output that is specific to Python. It
    should not concern students unduly here.
    payment = int(input('Enter payment up to £20 maximum. £'))
    payment = payment * 100 #converts payment into pence

    if payment == parkCharge:
        print('Exact money tendered. Thank you.')
    else:
        changeDue = payment - parkCharge
        print('Change due is', changeDue)
        print('\nHere is your change.')
        #(\n means: Start a new line)
        while changeDue >= 1000: #£10 note
            print('£10 note')
            changeDue = changeDue - 1000
        while changeDue >= 500: #£5 notes
            print('£5 note')
            changeDue = changeDue - 500
        while changeDue >= 200: #£2 coins
            print('£2 coin')
            changeDue = changeDue - 200
        while changeDue >=100: #£1 coins
            print('£1 coin')
            changeDue = changeDue - 100
        while changeDue >= 50:  #50p coins
            print('50p coin')
            changeDue = changeDue - 50
        while changeDue >= 20:  #20p coins
            print('20p coin')
            changeDue = changeDue - 20
        while changeDue >= 10:  #10p coins
            print('10p coin')
            changeDue = changeDue - 10
        while changeDue >= 5: #5p coins
            print('5p coin')
            changeDue = changeDue - 5
        while changeDue >= 2:  #2p coins
```

```
        print('2p coin')
        changeDue = changeDue - 2
    while changeDue >= 1:  #1p coins
        print('1p coin')
        changeDue = changeDue - 1
```

Checkpoint S1

Variables are defined in the Key term box on page 5 of Chapter 1 and page 6 includes an explanation of the purpose of variables. Students need to understand that variables have a variety of uses, for example controlling the number of times a loop is executed, determining which branch of an IF statement is taken, keeping running totals and holding user input etc.

Checkpoint S2

Ideally, students should give examples of different data types from their own programs.

Checkpoint S3

The description will vary according to the high-level language the student is studying. The main thing for them to realise is that the language will have a number of different selection and loop constructs.

Checkpoint C1

Command sequences, selection and iteration were described in detail in Chapter 1. They are revisited in this chapter on pages 51 – 53. Variable initialisation is covered on page 50.

As well as being able to describe these constructs, students should also be able to recognise them in an algorithm.

2.2 Making programs easy to read

Code readability

Activity 9

Solution not required.

Activity 10

1

```
SET countDown TO 10
#The loop counts down from 10 to 0
WHILE countDown >= 0 DO
    IF countDown > 0 THEN
        SEND countDown TO DISPLAY
    ELSE
        SEND 'Blast Off' TO DISPLAY
        #Prints 'Blast Off' when 0 is reached
    END IF
    SET countDown to countDown - 1
    #Ensures that the loop will terminate
END WHILE
```

2 Here is the algorithm implemented in Python.

```
countDown = 10
while countDown >= 0:
    if countDown > 0:
        print(countDown)
    else:
        print('Blast off')
    countDown -= 1
```

3 Here is the calculator algorithm expressed as source code

```
REPEAT

    REPEAT
        SEND 'Select an option: a - addition, s - subtraction, d - division or m
        - multiplication' TO DISPLAY
        RECEIVE choice FROM KEYBOARD
    UNTIL choice = 'a' OR choice = 's' OR choice = 'd' OR choice = 'm'

    SEND 'Enter the first number' TO DISPLAY
    RECEIVE number1 FROM KEYBOARD
    SEND 'Enter the second number' TO DISPLAY
    RECEIVE number2 FROM KEYBOARD

    IF choice = 'a' THEN
        SEND number1 + number2 TO DISPLAY
    ELSE
        IF choice = 's' THEN
            SEND number1 - number2 TO DISPLAY
        ELSE
```

```
            IF choice = ‘d’ THEN
                SEND number1 / number2 TO DISPLAY
            ELSE
                SEND number1 * number2 TO DISPLAY
            END IF
        END IF
    END IF
    SEND ‘Another go?’ TO DISPLAY
    RECEIVE anotherGo FROM KEYBOARD

UNTIL anotherGO <> ‘y’ OR anotherGo <> ‘Y’
```

4 Here is the algorithm implemented in Python.

```
anotherGo = 'y'
while anotherGo == 'y' or anotherGo == 'Y':
while True: #Loop to filter out invalid choices
choice = input('Select an option; a - addition, s - subtraction, d - division or
m - multiplication. ')

if choice == 'a' or choice == 's' or choice == 'd' or choice == 'm':
    break
else:
    print('Invalid selection.')

firstNumber = int(input('Enter the first number. '))
secondNumber = int(input('Enter the second number. '))

if choice == 'a':
    print(firstNumber + secondNumber)
elif choice == 's':
    print(firstNumber - secondNumber)
elif choice == 'd':
    print(firstNumber / secondNumber)
else:
    print(firstNumber * secondNumber)

anotherGo = input('Another go? ')
```

Exam-style question

1

		Line number(s)
a	Selection	03 or 07
b	Iteration	02 – 12
c	Variable initialisation	01

2 Descriptive names and Indentation. Techniques for improving the readability of code are described on pages 54 – 55

3 Some reasons for writing code that is easy to read are outlined on page 54.

Checkpoint S1

Covered on page 54.

Checkpoint S2

Listed in Table 2.2 on page 55.

Checkpoint C1

Solution not required. Encourage students to look at each other's code and assess its readability.

2.3 Strings

Activity 11

```
SEND 'Enter your password.' TO DISPLAY
RECEIVE password FROM KEYBOARD
IF LENGTH(password) < 6 THEN
    SEND 'The password you have entered is not long enough.' TO DISPLAY
ELSE
    SEND 'Length of password OK.' TO DISPLAY
END IF
```

Here is the algorithm implemented in Python.

```
password = input('Enter your password. ')
if len(password)< 6:
    print('The password you have entered is not long enough.')
else:
    print('Length of password OK.')
```

Activity 12

Here is the program implemented in Python.

```
numberA = 0 #Running total for each vowel initialised to 0
numberE = 0
numberI = 0
numberO = 0
numberU = 0
sentence = input('Enter your sentence. ')
for letter in sentence: #Loops through sentence letter by letter.
if letter == 'a' or letter == 'A':
    numberA += 1
elif letter == 'e' or letter == 'E':
    numberE +=1
elif letter == 'i' or letter == 'I':
    numberI +=1
elif letter == 'o' or letter == 'O':
    numberO +=1
elif letter == 'u' or letter == 'U':
    numberU +=1
#Updates appropriate running total if letter is a vowel
numberVowels = numberA + numberE + numberI + numberO + numberU
print('In this sentence there are', numberVowels,'vowels. They are:')
print('A:\t', numberA)
print('E:\t', numberE)
print('I:\t', numberI)
print('O:\t', numberO)
print('U:\t', numberU)
```

Activity 13

1 Solution not required.

2 Here is the program implemented in Python.

```
firstName = input('Enter your first name: ')
surname = input('Enter your surname: ')
length = len(surname) #Deals with surnames less than 4 characters long

if length == 1:
    surname = surname & 'xxx'
elif length == 2:
    surname = surname & 'xx'
elif length == 3:
    surname = surname & 'x'

first4Letters = surname[0:4]
userName = firstName[0] & first4Letters
print('Your username is:', userName)
```

3 Python has a `string.split()` method which could be used here.

```
text = "23456,West Kirby,04/11/15,23.0,11.5,30,D1"
separateStrings = text.split(',')
print(separateStrings)
```

This produces a list of seven separate strings. At this stage students may not have had any practical experience of working with arrays (lists) so you may want to postpone this activity until after they have studied the section on data structures.

Activity 14

Solution not required.

Checkpoint S1

String indexing is explained on page 58.

Checkpoint S2

The program students developed in Activity 12 uses a loop to traverse a string.

Checkpoint S3

Concatenation and type conversion are explained on page 59.

Checkpoint C1

Here is the program implemented in Python.

```
sentence = input('Enter a sentence. ')
separateStrings = sentence.split(' ') #Splits up the sentence into a list of
strings
for item in separateStrings:
    #Prints each item in separateStrings on a separate line
    print(item)
```

2.4 Data structures

Activity 15

A linear search algorithm traverses a list sequentially until it finds the item it is looking for or the end of the list is reached.

Here is the algorithm implemented in Python.

```
#Linear search
firstNames = ['Alex', 'Bryn', 'Eloise', 'Lois', 'James', 'Sally']
searchName = input('What name are you looking for? ')
found = False
index = 0

while found == False and index <= (len(firstNames) - 1):
    if searchName == firstNames[index]:
        found = True
    else:
        index += 1

if found == True:
    print(searchName, 'is in position', index, 'in the list.')
else:
    print(searchName, 'is not in the list.')
```

Activity 16

Here is the algorithm implemented in Python.

```
#Bubble sort - items sorted in descending order
exam1 = [25, 46, 71, 18, 31, 78, 92, 49, 63, 62, 11, 90, 83, 71, 41]
swapped = 1

while swapped == 1:
    swapped = 0

    for index in range(1, len(exam1)):
        if exam1[index - 1] < exam1[index]:
        temp = exam1[index - 1]
        exam1[index - 1] = exam1[index]
        exam1[index] = temp
        swapped = 1

print('The highest result was', exam1[0])
print('The lowest result was', exam1[len(exam1) - 1])
```

Activity 17

Solution not required.

Activity 18

1 Solution not required.

2 See solution to 3.

3 Here is the program implemented in Python.

```
marks = [[80, 59, 34, 89], [31, 11, 47, 64], [29, 56, 13, 91], [55, 61, 48, 0],
[75, 78, 81, 91]]
#Initialises the list
highestMark = marks[0][0]
lowestMark = marks[0][0]

#Initialises highest and lowest mark to first mark in the list
total = 0 #Adds up the marks
count = 0 #Counts the numbers of marks

for row in marks:
    for column in row:
    total += column
    count +=1
    if column > highestMark:
        highestMark = column
    elif column < lowestMark:
        lowestMark = column

print('The highest mark is', highestMark)
print('The lowest mark is', lowestMark)
print('The average mark is', total/count)
```

4 Here is the program implemented in Python.

```
gameScores = [['Alex', 1, 19], ['Seema', 1, 29], ['Seema', 2, 44], ['Lois', 1,
10], ['Alex', 2, 17], ['Alex', 3, 36], ['Dion', 1, 23], ['Emma', 1, 27],
['Emma', 2, 48]]
highestL1Score = 0
highestL1Player = ''
highestL2Score = 0
highestL2Player = ''
highestL3Score = 0
highestL3Player = ''

for row in gameScores:
    player = row[0]
    level = row[1]
    score = row[2]

    if level == 1 and score > highestL1Score:
        highestL1Score = score
        highestL1Player = player
    elif level == 2 and score > highestL2Score:
        highestL2Score = score
        highestL2Player = player
    elif level == 3 and score > highestL3Score:
        highestL3Score = score
        highestL3Player = player
```

```
print('The highest score in Level 1 was', highestL1Score, 'achieved by',
highestL1Player)
print('The highest score in Level 2 was', highestL2Score, 'achieved by',
highestL2Player)
print('The highest score in Level 3 was', highestL3Score, 'achieved by',
highestL3Player)
```

Activity 19

1 a string

b string

c integer

d string

2 Here is the program implemented in Python.

```
albumCollection = [['Where Rivers Meet', 'Z Rahman', 2008, 'World'], ['Best of
Cat Stevens', 'C Stevens', 1984, 'Pop'], ['Come Away With Me', 'N Jones', 2012,
'Pop'], ['Shine', 'Bond', 2002, 'Instrumental'], ['Blessing', 'J Rutter', 2012,
'Classical']]
anotherGo = 'y'
while anotherGo == 'y':
    while True:
        choice = input("Press 'e' to enter details of a new album, or 's' to
        search for an album. ")
        if choice == 'e' or choice == 's':
            break

    if choice == 'e':
        newAlbum = []
        album = input('Enter the name of the album: ')
        artist =  input('Enter the name of the artist: ')
        year =  int(input('Enter the year of release: '))
        genre =  input('Enter the genre: ')
        newAlbum = [album, artist, year, genre]
        albumCollection.append(newAlbum)
    else:
        searchAlbum = input('Enter the title of the album: ')
        found = False
        index = 0

        while found == False and index <= (len(albumCollection) - 1):
            if albumCollection[index][0] == searchAlbum:
                print('Artist:', albumCollection[index][1], '\nYear of release:',
                albumCollection[index][2])
                found = True
            else:
                index = index + 1

        if found == False:
        print('This album is not in your collection.')

    anotherGo = input("Press 'y' if you want another go. ")
```

Exam-style question

1 record

2 The data to be stored about each product includes productID, which has the data type *string*, and monthly sales figures, which have the data type *real*. The record data structure is most suitable for this purpose because it allows related values with different data types to be stored together using a single identifier.

3

ProductID	JanSales	FebSales	MarSales
X14	1200	1650	1876
X98	150	132	512
K12	325	320	355
P91	950	1010	1710

Checkpoint S1

0

Checkpoint S2

There's a detailed explanation of how a linear search algorithm works in Chapter 1, pages 34 – 35. And students implemented the algorithm themselves in Activity 15.

Checkpoint S3

Indexing in two-dimensional arrays is described in the Key term box on page 64.

Checkpoint S4

Using a nested loop to traverse a two-dimensional array is illustrated on page 64.

Checkpoint C1

The address book is best represented by an array of arrays. Here is the program implemented in Python.

```
addressBook = [['Andrew Smith', 'as@buzz.com'], ['Maddison Jones',
'madJon@gmool.co.uk'], ['Bert Tintwhistle', 'BTWhistle@vgmail.com'], ['Richard
Burton', 'RichBurt@ct.com'], ['Ann Mills', 'ann@mills.co.uk']]
anotherGo = 'y'

while anotherGo == 'y':
    searchName = input('Enter a name: ')
    found = False
    index = 0

    while found == False and index <= (len(addressBook) - 1):
        if addressBook[index][0] == searchName:
            print('The email address of', searchName, 'is',
            addressBook[index][1])
            found = True
        else:
            index = index + 1

    if found == False:
        print('This contact is not in your address book.')
    anotherGo = input("\nPress 'y' if you want another go. ")
```

Checkpoint C2

There is no single right answer to this challenge. Here is one attempt implemented in Python.

```
import random
anotherGo = 'y'

while anotherGo == 'y' or anotherGo == 'Y':
    locRow = random.randint(0, 3)
    locCol = random.randint(0, 3)
    found = False
    print('\nThe treasure map consists of a grid of four rows and four columns.
    Indexing begins at 0.\n')

    while not found:
        guessRow = int( input('Enter a row: '))
        guessCol = int(input('Enter a column: '))
        print(guessRow, guessCol)
        if guessRow == locRow and guessCol == locCol:
            found = True
            print("Well done. You've found the treasure.")
        elif guessRow == locRow:
            print('Right row, wrong column. Try again.')
        elif guessCol == locCol:
            print('Right column, wrong row. Try again.')
        else:
            print('Nowhere near. Try again.')

    anotherGo = input('\nDo you want another go? ')
```

2.5 Input/output

Activity 20

Here is the range check algorithm implemented in Python.

```
validNum = False
while validNum == False:
    number = int( input('Please enter a number between 1 and 10: '))
    if number >=1 and number <= 10:
        validNum = True
        print('You have entered:', number)
```

Activity 21

Here is the user menu implemented in Python.

```
print("\n_________MENU_________\n")
print("A: Display average temperature\n")
print("B: Display temperature range\n")
print("C: Display wind speed\n")
print("Enter 'Q' to quit the program\n")
validChoice = True
while validChoice:
    optionsChoice = input('\nPlease enter an option (A, B, C or Q): ')
    if optionsChoice == "a" or optionsChoice == "A":
        print ('You have selected Option A - Display average temperature.' )
    elif optionsChoice == "b" or optionsChoice == "B":
        print ('You have selected Option B - Display temperature range.' )
    elif optionsChoice == "c" or optionsChoice == "C":
        print ('You have selected Option C - Display wind speed.' )
    elif optionsChoice == "q" or optionsChoice == "Q":
        print ('You have opted to quit the program.' )
        validChoice = False
    else:
        print('Invalid selection.')
```

Activity 22

Here is the length check algorithm implemented in Python.

```
postCode = input('Enter a postcode: ')
length = len(postCode)
if length >= 6 and length <= 8:
    print('Valid')
else:
    print('Invalid')
```

Activity 23

Here is the presence check algorithm implemented in Python.

```
userName = ''
```

```
while userName == '':
    userName = input('Enter a username: ')
    print('Hello', userName)
```

Activity 24

Here is the look-up check algorithm implemented in Python. It uses the keyword ‘in’ to check if the form entered matches one of those stored in arrayForms. This avoids looping and so enables the code to be much shorter and simpler than the pseudo-code example given in the question.

```
arrayForms = ['7AXB', '7PDB', '7ARL', '7JEH']
form = input('Enter a form: ')
if form in arrayForms:
    print('Valid form.')
else:
    print('The form you entered does not exist.')
```

Activity 25

Here is the program produced for activity 18 with a menu added.

```
marks = [[80, 59, 34, 89], [31, 11, 47, 64], [29, 56, 13, 91], [55, 61, 48, 0],
[75, 78, 81, 91]]
#Initialises the list
highestMark = marks[0][0]
lowestMark = marks[0][0]

#Initialises highest and lowest mark to first mark in the list
total = 0 #Adds up the marks
count = 0 #Counts the numbers of marks

for row in marks:
    for column in row:
        total += column
        count +=1
        if column > highestMark:
            highestMark = column
        elif column < lowestMark:
            lowestMark = column

#Menu
print("\n_________MENU_________\n")
print("A: Display highest mark\n")
print("B: Display lowest mark\n")
print("C: Display average mark\n")
print("Enter 'Q' to quit the program\n")
validChoice = True

while validChoice:
    optionsChoice = input('\nPlease enter an option (A, B, C or Q): ')
    if optionsChoice == "a" or optionsChoice == "A":
        print ('\nThe highest mark is', highestMark)
    elif optionsChoice == "b" or optionsChoice == "B":
        print ('\nThe lowest mark is', lowestMark)
    elif optionsChoice == "c" or optionsChoice == "C":
```

```
        print ('\nThe average mark is', total/count)
    elif optionsChoice == "q" or optionsChoice == "Q":
        print ('\nYou have opted to quit the program.')
        validChoice = False
    else:
        print('Invalid selection.')
```

Activity 26

1 & 2 Here is the program plus extension implemented in Python. The data file 'marks.txt' is supplied with this pdf file.

```
#Stage 1
rawData = open("marks.txt", "r")
inputData = rawData.readlines()
rawData.close()

#Stage 2 - splits each line of rawData into a list of strings and appends each
list to the 2 dimensional array examResults
examResults= []
index = 0
for line in inputData:
    examResults.append(inputData[index].split(","))
    index += 1
length = len(examResults)

#Stage 3
totalE1 = 0
totalE2 = 0
totalE3  =0
totalE4 = 0
for student in examResults: #Totals up all the marks achieved in each exam
    totalE1 += int(student[1])
    totalE2 += int(student[2])
    totalE3 += int(student[3])
    totalE4 += int(student[4])

#Menu
print("\n_________MENU_________\n")
print("A: Average mark achieved in the first exam\n")
print("B: Average mark achieved in the second exam\n")
print("C: Average mark achieved in the third exam\n")
print("D: Average mark achieved in the fourth exam\n")
print("Enter 'Q' to quit the program\n")

validChoice = True

while validChoice:
    optionsChoice = input('\nPlease enter an option (A, B, C , D or Q): ')
    if optionsChoice == "a" or optionsChoice == "A":
        print ('\nThe average mark achieved in the first exam was',
        totalE1/length)
```

```
    elif optionsChoice == "b" or optionsChoice == "B":
        print ('\nThe average mark achieved in the second exam was',
        totalE2/length)
    elif optionsChoice == "c" or optionsChoice == "C":
        print ('\nThe average mark achieved in the third exam was',
        totalE3/length)
    elif optionsChoice == "d" or optionsChoice == "D":
        print ('\nThe average mark achieved in the fourth exam was',
        totalE4/length)
    elif optionsChoice == "q" or optionsChoice == "Q":
        print ('\nYou have opted to quit the program.')
        validChoice = False
    else:
        print('Invalid selection.')
```

Activity 27

Here is the amended program that rejects any invalid records and writes them to an error log. The data file 'marks2.txt', with two invalid records, is supplied with this pdf file.

```
rawData = open("marks2.txt", "r")
inputData = rawData.readlines()
rawData.close()
#Splits each line of rawData into a list of strings and appends each list to the
2 dimensional array examResults
examResults = []
index = 0


for line in inputData:
    examResults.append(inputData[index].split(","))
    index += 1


errorLog = open("errorLog.txt", "w")


for student in examResults:
    #Strips out any erroneous records
    if int(student[1]) < 1 or int(student[1]) >100 or int(student[2]) < 1 or
    int(student[2]) >100 or int(student[3]) < 1 or int(student[3]) >100 or
    int(student[4]) < 1 or int(student[4]) >100:
        invalidRec = ",".join(student) #Converts invalid record into a single
        string with elements separated by a comma
        errorLog.write(invalidRec)
        examResults.remove(student) #Removes record from examResults


errorLog.close()
length = len(examResults)
#Sums up marks achieved in each exam
totalE1 = 0
totalE2 = 0
totalE3  =0
totalE4 = 0
```

```
for student in examResults: #Totals up all the marks achieved in each exam
    totalE1 += int(student[1])
    totalE2 += int(student[2])
    totalE3 += int(student[3])
    totalE4 += int(student[4])

#Menu
print("\n_________MENU_________\n")
print("A: Average mark achieved in the first exam\n")
print("B: Average mark achieved in the second exam\n")
print("C: Average mark achieved in the third exam\n")
print("D: Average mark achieved in the fourth exam\n")
print("Enter 'Q' to quit the program\n")
validChoice = True

while validChoice:
    optionsChoice = input('\nPlease enter an option (A, B, C , D or Q): ')
    if optionsChoice == "a" or optionsChoice == "A":
        print ('\nThe average mark achieved in the first exam was',
        totalE1/length)
    elif optionsChoice == "b" or optionsChoice == "B":
        print ('\nThe average mark achieved in the second exam was',
        totalE2/length)
    elif optionsChoice == "c" or optionsChoice == "C":
        print ('\nThe average mark achieved in the third exam was',
        totalE3/length)
    elif optionsChoice == "d" or optionsChoice == "D":
        print ('\nThe average mark achieved in the fourth exam was',
        totalE4/length)
    elif optionsChoice == "q" or optionsChoice == "Q":
        print ('\nYou have opted to quit the program.')
        validChoice = False
    else:
        print('Invalid selection.')
```

Activity 28

If a file is written to, any pre-existing data at that point in the file will be over-written. To avoid this happening, new records should be appended to the *end* of an existing file.

Exam-style question

There are a number of acceptable answers, including

(i) Make sure the products are in alphabetical order to enable faster retrieval.

(ii) Put all the data for a single product-type (e.g. Drinks) in a single record.

(iii) Split the product (e.g. Apple Juice) and the quantity (e.g. 1 litre) into separate items.

Checkpoint S1

The ‘garbage in, garbage out’ principle is explained on page 68.

Checkpoint S2

A one-dimensional array would be suitable for this purpose.

Checkpoint S3

Gordon Weidmann, 30 May 1954, purple

Checkpoint S4

The advantage of writing data to a text file is explained on page 72.

Checkpoint C1

Here's the program implemented in Python. It's worth revisiting it, once students have studied the next section of this chapter, and getting them to redesign it using subprograms. The data file 'employees.txt' is supplied with this pdf file.

```
departments = ['Sales', 'Production', 'Design', 'Finance', 'Cust Service']
#Menu
print("\n_________MENU_________\n")
print("E: Enter an employee's details\n")
print("S: Search for an employee's details\n")
print("Enter 'Q' to quit the program\n")
validChoice = True

while validChoice:
    optionsChoice = input('\nPlease enter an option (E, S or Q): ')

    if optionsChoice == "e" or optionsChoice == "E":
        employees = open("employees.txt", "a")
        #Uses append tag ("a") to avoid overwriting existing records
        while True:
            employeeNumb = input("\nEnter employee's number: ")
            if len(employeeNumb) == 3:
                #Checks that employeeNumb is three digits long
                break

        employeeName = input("\nEnter employee's name: ")

        while True:
            employeeDept = input("\nEnter employee's department: ")
            if employeeDept in departments:
                break

        newRecord =  employeeNumb & "," & employeeName & "," & employeeDept &
        "\n"
        employees.write(newRecord)
        employees.close()

    elif optionsChoice == "s" or optionsChoice == "S":
        rawData = open("employees.txt", "r")
        inputData = rawData.readlines()
        rawData.close()
        #Splits each line of rawData into a list of strings and appends each list
        to the 2 dimensional array employeeRecords
        employeeRecords = []
        index = 0

        for line in inputData:
            employeeRecords.append(inputData[index].split(","))
```

```
            index += 1

        #Linear search used to find employee in file
        searchID = input("\nEnter the employee's number: ")
        found = False
        index = 0

        while found == False and index <= len(employeeRecords):
            if employeeRecords[index][0] == searchID:
                print('\nThis is', employeeRecords[index][1], 'who works in',
                employeeRecords[index][2])
                found = True
            else:
                index += 1

        if found == False:
            print('This employee is not in the file.')

    elif optionsChoice == "q" or optionsChoice == "Q":
            print ('\nYou have opted to quit the program.' )
            validChoice = False
    else:
print('Invalid selection.')
```

2.6 Subprograms

Activity 29

1 Solution not required.

2 Here is the die function implemented in Python.

```
import random
def die():
    simDie = random.randint(1,6)
    return simDie

#Main program
print(die())
```

Here is the averageScore procedure implemented in Python.

```
def averageScore(score1, score2, score3):
    total = score1 + score2 + score3
    average = total / 3
    print(average)
#Main program
averageScore(20, 24, 19)
```

Activity 30

1 Here is the cubeSurfaceArea procedure implemented in Python.

```
def cubeSurfaceCalc(width, height):
    cubeArea = width * height
    cubeSurfaceArea = cubeArea * 6
    print(cubeSurfaceArea)
#Main program
cubeSurfaceCalc(10, 5)
```

2 Here is the function implemented in Python, using two of Python's built in list functions – max and min.

```
def highLow(list):
    highest = max(list) #Function uses two built-in list methods
    lowest = min(list)
    return highest, lowest
#Main program
numbers = [12, 45, 1002, 3, 734, 0, -13, 950]
print(highLow(numbers))
```

Activity 31

Local variables: `total, saving, newTotal`

Global variables: `subtotal, salePrice`

Constants: `discount`

Exam-style question

1

	Line number(s)
a. variable initialisation	01, 06, 07
b. type declaration	02
c. selection	09
d. iteration	08 – 12
e. data structure	01
f. subprogram	04 – 14

2 The call to the subprogram maxCalc is on line 16.

Exam-style question

1

```
PROCEDURE processItem(itemName, itemPrice, itemWeight)
BEGIN PROCEDURE
    SET previousTotal TO totalWeight
    SEND itemName, itemPrice TO DISPLAY
    REPEAT
        SEND 'Place item in bagging area.' TO SPEAKER
        delay()
    UNTIL totalWeight = previousTotal + itemWeight
    SEND 'Scan next item.' TO SPEAKER
END PROCEDURE
```

2 The function has two parameters: `width` and `height`.

3 50

4 50

5 5

Checkpoint S1

The benefits of using subprograms are explained on page 77.

Checkpoint S2

Variable scope is explained on page 78. Students could use the exam-style question on page 81 to illustrate the scope of a variable.

Checkpoint S3

The Top Tip on page 79 provides the answer to this question.

Checkpoint S4

There are loads to choose from. The two specified in Edexcel pseudo-code are LENGTH() and RANDOM(). Others students may have come across in Python are max(), int(), print(), range(), type(), str().

Checkpoint C1

Here is the program implemented in Python. It makes use of the imported statistics module, which includes mean, median and mode methods. Error handling is incorporated to prevent the user from entering any non-integers and to stop the program from crashing if the mode can't be calculated for the given set of numbers.

```
import statistics
def enterSet():
    numbers = []
    anotherNumber = 'y'
```

```
    while anotherNumber == 'y' or anotherNumber == 'Y':
        while True:
            #Prevent the user from entering non-integer values
            try:
                nextNumber = int(input('\nNumber? '))
                numbers.append(nextNumber)
                break
            except ValueError:
                print('Your entry must be a valid integer. Please try again.')

        anotherNumber = input('\nAny more numbers to be entered? ')

    numbers.sort()
    print('\nYou have entered this set of numbers', numbers)
    return(numbers)

def menu():
    print("\n_________MENU_________\n")
    print("A: Mean\n")
    print("B: Median\n")
    print("C: Mode\n")
    print("Enter 'Q' to quit the program\n")
    validChoice = True

    while validChoice:
        optionsChoice = input('\nPlease enter an option (A, B, C or Q): ')
        if optionsChoice == "a" or optionsChoice == "A":
            print ('\nThe mean is', statistics.mean(numberSet))
        elif optionsChoice == "b" or optionsChoice == "B":
            print ('\nThe median is', statistics.median(numberSet))
        elif optionsChoice == "c" or optionsChoice == "C":

            while True:
            #Prevents program from crashing if mode can't be calculated for given
            set of numbers
                try:
                    print ('\nThe mode is', statistics.mode(numberSet))
                    break
                except ValueError:
                    print('Not possible to calculate mode for this set of
                    numbers.')
                    break

        elif optionsChoice == "q" or optionsChoice == "Q":
            print ('\nYou have opted to quit the program.' )
            validChoice = False
        else:
            print('Invalid selection.')
```

```
#Main program
numberSet = enterSet()
menu()
```

2.7 Testing and evaluation

Activity 32

length	count	index	gender[index]
10			
	0		
		0	
			M
		1	
			M
		2	
			F
	1		
		3	
			M
		4	
			F
	2		
		5	
			F
	3		
		6	
			M
		7	
			F
	4		
		8	
			M
		9	
			F
	5		
		10	

Activity 33

1 If height has the value 0 a runtime error will occur. However, there could be a logic error here too since area is calculated by multiplying, not dividing, width by height. Alternatively, the confusion could be the result of choosing an inappropriate variable name ('area') – 'ratio' would be more appropriate here.

2 Here is the algorithm implemented in Python.

```
width = int(input('Enter the width: '))
while True:
    height = int(input('Enter the height: '))
    if height > 0:
        break

ratio = width/height
print('The ratio is:', ratio)
```

Activity 34,

The array has only four elements, so when `index` has the value 4 the end of the array will be exceeded

Activity 35

The RANDOM function in Edexcel's pseudo-code has only one parameter 'n' and generates numbers between 0 and 'n'.

1

```
SET number TO RANDOM(5) + 1
SET guessed TO False
WHILE guessed = False DO
    REPEAT
        SEND 'Enter a number between 1 and 6' TO DISPLAY
    UNTIL guess > 0 AND guess <7 THEN
    IF guess = number THEN
        SEND 'Well Done.' TO DISPLAY
        SET guessed TO True
    ELSE
        SEND 'Try again.' TO DISPLAY
    END IF
END WHILE
```

2 The plan needs to test that:

- the program generates numbers in the correct range (1 – 6)
- user input is restricted to numbers between 1 and 6
- appropriate messages are output for correct and incorrect guesses
- the program terminates when a correct guess is entered and continues until that point.

3

```
import random
number = random.randint(1, 6)
guessed = False
while guessed == False:
    while True:
        guess = int(input('Enter a number between 1 and 6: '))
        if guess > 0 and guess < 7:
            break

    if guess == number:
        print('Well Done')
        guessed = True
    else:
        print('Try again')
```

4 Solution not required.

Exam-style question

1 The algorithm counts the number of marks of 50 or above in the array scores.

2

length	count	index	scores[index]
5			
	0		
		0	
			45
		1	
			67
	1		
		2	
			34
		3	
			98
	2		
		4	
			52
	3		

Checkpoint S1

The three types of error are summarised in the table on page 85.

Checkpoint S2

Trace tables were introduced on page 21 – 22 of Chapter 1 and are revisited on pages 83 – 84.

Checkpoint S3

Some features of IDE's are described on page 86, but students should provide examples from the IDE they use.

Checkpoint S4

The purpose of a test plan and how to design and implement a test plan are covered on pages 87 – 90.

Checkpoint S5

Normal, boundary and erroneous data are explained on page 88.

Checkpoint S6

Whenever an improvement is made to a program, there's a possibility that fresh errors are unwittingly introduced at the same time – hence the need to retest.

Checkpoint C1 – C3

The 'Preparing for your Exam' section includes an opportunity (pages 235 – 239) for students to tackle this challenge.

Chapter 3: Data

3.1 Binary

Activity 1

a 00111001 = 57

b 11000110 = 198

c 10101010 = 170

Activity 2

a 69 = 01000101

b 193 = 11000001

c 239 = 11101111

Activity 3

a 10011010 + 11010111 = 101110001

b 00001101 + 10101010 = 10110111

c 11010111 + 10001010 = 101100001

Activity 4

1 a −113 = 10001111

b −56 = 11001000

c −90 = 10100110

2 90 + –33 = 01011010 + 11011111 = 00111001

Activity 5

Calculate the results of the following shifts on unsigned integers.

a 00111010 * 2^3 = 111010000

b 10011101 / 2^4 = 00001001

Activity 6

Show the following division:

1 & 2

a 10010000 / 2^2 = 11100100 (–112 / 4 = –28)

b 11110110 / 2^1 = 11111011 (–10 / 2 = –5)

c 11000000 / 2^3 = 11111000 (–64 / 8 = –8)

Activity 7

1 a 01101110 = 6E

b 10011100 = 9C

c 00101010 = 2A

2 a A6 = 10100110

b 9C = 10011100

c 2D = 00101101

Exam-style questions

1 a 101000000

b An overflow has occurred.

2 a 00010010

b Can lead to a loss of precision OR Divides by a power of 2.

c The diagram on page 104 illustrates the steps involved in converting from binary to hex. 11101110 is EE in hex.

Checkpoint S1

11011001 + 10010010 = 101101011 (217 + 146 = 363)

Checkpoint S2

00011001 + 11110011 = 00001100 (25 + (−13) = 12)

Checkpoint S3

1001001000 (73 × 8 = 584)

Checkpoint S4

11001101 = 205 in denary and CD in hex.

Checkpoint C1

The use of binary to represent data and program instructions is explained on page 92.

Checkpoint C2

The difference between a logical shift and an arithmetic shift is explained on pages 101 – 103.

3.2 Data representation

Activity 8

The ASCII code for ‘ASCII code’ is: 01000001 01010011 01000011 01001001 01001001 00100000 01100011 01101111 01100100 01100101 00101110

Activity 9

```
FUNCTION chr(number)
BEGIN FUNCTION
    SET arrayNumb2Ascii TO [[32, ' '], [33, '!'], [34, '"'], [35, '#'], [36,
    '$'], [37, '%'], [38, '&'], ……..[125,'}']]
    #A 2-dimensional array containing all of the denary ASCII codes and
    characters
    FOR index FROM 0 TO LENGTH(arrayNumb2Ascii) - 1 DO
        IF number = arrayNumb2Ascii[index, 0] THEN
            character = arrayNumb2Ascii[index, 1]
        END IF
    END FOR
    RETURN character
END FUNCTION
```

Activity 10

This algorithm is revisited later in the chapter in the section Encryption. You may want to postpone doing this activity until students have learned more about the Caesar cipher algorithm. Here’s the program written in Python.

```
message = input('Enter the message to encrypt: ')
shift = int(input('\nEnter the size of the shift: '))
secretMessage = ''

for character in message:
    number = ord(character)

    if character.lower() in'abcdefghijklmnopqrstuvwxyz':
        #Checks that the character is a letter rather than a space or punctuation
        mark
        number += shift

        #deals with letters at start and end of alphabet
        if character.isupper(): #It's upper case
            if number > ord('Z'):
                number -= 26
            elif number < ord('A'):
                number += 26
        else: #Must be lower case
            if number > ord('z'):
                number -= 26
            elif number <ord('a'):
                number += 26
        secretMessage = secretMessage & chr(number)
```

```
        else:
            #Non-letters are added unchanged
            secretMessage = secretMessage & character

    print (secretMessage)
```

Activity 11

1 8 × 640 × 480 = 2,457,600 bits = 307,200 bytes = 0.29 MB

2 24 × 640 × 480 = 7,372,800 bits = 921,600 bytes = 0.88 MB

Activity 12

44,100 × 24 × 3 × 60 × 2 = 381,024,000 bits = 47,628,000 bytes = 45.42 MB

Exam-style questions

1 Pixel is defined in the key term box on page 109.

2 a There are 36 pixels and one bit is needed to represent each pixel. Encoding pixel information is explained on pages 110 – 111.

b Assuming that 0 represents black and 1 represents white, the bit pattern is:

100011

101111

100111

101111

101111

100011

c 4 bits would be needed to represent 16 (2^4) colours.

3 a The process of digital recording is explained on pages 113 – 114.

b Increasing the sampling rate will improve the fidelity of the recording (definition on page 114) and increase the file size of the recording.

c 44100 × 16 × 2 × 60 = 84,672,000 bits = 10,584,000 bytes = 10.09 MB.

Checkpoint S1

Text to be converted: 'Data is represented as bits.'

01000100 01100001 01110100 01100001 00100000 01101001 01110011 00100000 01110010 01100101 01110000 01110010 01100101 01110011 01100101 01101110 01110100 01100101 01100100 00100000 01100001 01110011 00100000 01100010 01101001 01110100 01110011 00101110

Checkpoint S2

24 × 640 × 480 = 737,280 bits = 0.88 MB.

Checkpoint C1

How resolution affects image quality is illustrated on page 110.

Checkpoint C2

The factors that affect the file size of audio recordings are listed on page 115 and explained on the previous two pages.

3.3 Data storage and compression

Activity 13

1 Using a binary prefix:

a 1.5 TB = 1,536 GB

b 1.5 TB = 1,572,864 MB

c 1.5 TB = 1,610,612,736 KB

d 1.5 TB = 1.5 TB (given in the question. TB means terabytes)

Using a decimal prefix:

a 1.5 TB = 1,500 GB

b 1.5 TB = 1,500,000 MB

c 1.5 TB = 1,500,000,000 KB

d 1.5 TB = 1.5 TB

2 363,143,213 bits = 0.04 GB

Activity 14

Code	RLE version	Size of coded version
wbbbbbww	1w5b2w	6
wbbbbbww	1w5b2w	6
wwwbwwww	3w1b4w	6
wwwbwwww	3w1b4w	6
wwwbwwww	3w1b4w	6
wwwbwwww	3w1b4w	6
wwwbwwww	3w1b4w	6
wwwbwwww	3w1b4w	6
64 bytes		48 bytes

Activity 15

1

Variables	Data items
text	Input string
newText	The next character in the input string to be evaluated
code	The encoded string
length	Length of input string
run	The number of repetitions of a character
index	Index position of character in input string

2 A4, B2, C1, D5, E3

1 Item C. This checks to see if the end of the input text has been reached. (C1)

2 Item B. If there is only one character it and its run length (of 1) are added to the output file. (B2)

3 Item E. This adds the character and its run length to the output string. (E3)

4 Item A. This checks to see that some text has actually been entered. (A4)

5 Item D. This checks to see if the next character in the input string is the same as or different from the one that is being checked. (D5)

Exam-style question

1 bit, nibble, byte, MB, GB, TB

2 3w3b2w6b3w3b3w1b

Checkpoint S1

bit, nibble, byte, KB, MB, GB, TB

Checkpoint S2

2.3 TB = 2,528,876,743,884.8 bytes

Checkpoint S3

3a2b9a3c3d

File size of original = 20 bytes

File size of encoded version= 10 bytes

Checkpoint C1

A lossless compression algorithm looks for repeat values in an image in order to create a new more compact representation with a smaller file size. The amount of compression that can be achieved using a lossless compression algorithm depends on how many areas of uniform colour an image has. Images with very short runs of different colours – even if the difference is only marginal – don't compress well.

A lossy algorithm would replace do away with marginal differences in colour in an image and replace them with the same colour value, relying on the fact that the human eye isn't capable of detecting the difference.

Checkpoint C2

Why a compressed audio file often sounds the same as the uncompressed version is explained on page 123.

3.4 Encryption

Activity 16

'This is a message' → 'Wklv lv d phvvdjh'

Students could use the program they wrote in Activity 10 to encrypt the message.

Activity 17

The program students developed in Activity 10 will also *decrypt* messages if the key is reversed (i.e. +3 becomes –3). But they may want to refine this program to offer users a choice of encrypt or decrypt.

Exam-style questions

1 'Computer' → 'lusvazkx'

2 'Pzfbkzb' → 'Science'

Checkpoint S1

'Data should be encrypted' → 'Mjcj bqxdum kn nwlahycnm'

Checkpoint S2

How a Caesar cipher works is explained on pages 126 – 128.

Checkpoint C1

	+5	**–2**			**+3**
g	l	j		**g**	j
o	t	r		**o**	r
l	q	o		**l**	o
d	i	g		**d**	g

3.5 Databases

Activity 18

CAR(make, model, year, engine size, colour)

BOOK(author, title, publisher, year, genre)

DVD(title, year, length, director, production company)

Activity 19

1 a & b Data redundancy and resulting data inconsistency are explained on pages 132 –133.

2 Here is the restructured database with data redundancy removed.

BOOKS(Book_ID, Title, Publisher, Date_Published, Author_ID)

AUTHORS(Author_ID, Author_Surname, Author_First_name, Author_DOB, Author_Nationality)

Author_ID is the primary key of the AUTHOR table. A compound primary key consisting of Author_Surname, Author_First_name and Author_DOB could be used as an alternative. The primary key Author_ID is posted as a foreign key in the BOOKS table to create a relationship between the two tables.

Activity 20

Here is a design for the database.

Tables:

CUSTOMER(**Customer_ID**, Customer_Surname, Customer_Firstname, Post_Code, …)

ORDER(**Order_Number**, Date, Order_Total, **Customer_ID, Driver_ID**, Order_Status, …)

ORDERLINE(**Order_Number**, Product_Code)

PRODUCT(**Product_Code**, Product_Name, Item_Price, …)

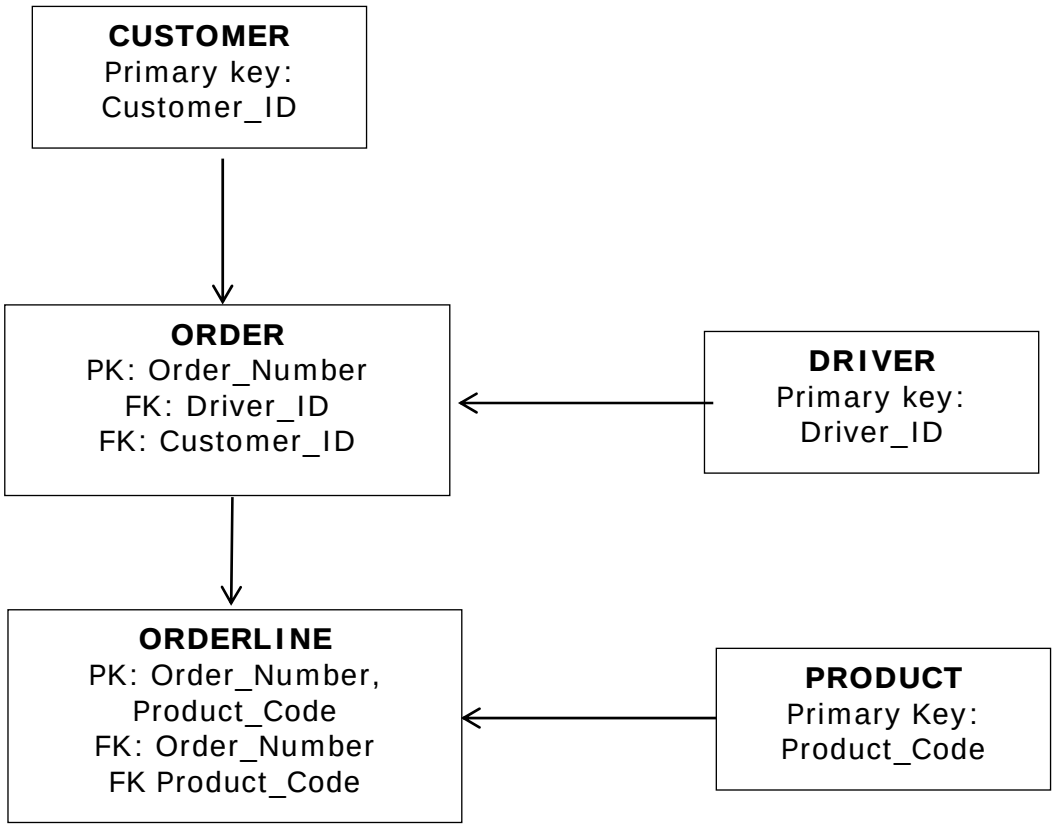
DRIVER(**Driver_ID**, Driver_Name, …)

There is a one-to-many relationship between the CUSTOMER table and the ORDER table. Each customer is likely to place more than one order, but every order is for a particular customer. Customer_ID is the primary key of the CUSTOMER table and is planted as a foreign key in the ORDER table.

There is a one-to-many relationship between the ORDER table and the ORDERLINE table. Each order is likely to consist of a several order lines, but every order line is associated with a particular order. Order_Number is the primary key of the ORDER table and is planted as a foreign key in the ORDERLINE table.

There is a one-to-many relationship between the DRIVER table and the ORDER Table. One driver delivers each order, but each driver makes lots of deliveries. Driver_ID is the primary key of the DRIVER table and is planted as a foreign key in the ORDER table.

Each order line is for a particular product, but each product can appear in many order lines. The primary key of the PRODUCT table is Product_Code. It is planted as a foreign key in the ORDERLINE table.



Exam-style questions

1 a The primary key of the MEMBERS table is MemberNumber.

b A primary key must uniquely identifier each record in the table. There's a chance that there could be more than one member with the same surname, first name and even date of birth. Using a unique number for each member avoids any potential confusion.

2 SportCode and TeamCode are foreign keys used to link the MEMBERS table with the SPORTS and TEAMS tables. By linking the tables in this way, information about sports and teams doesn't need to be included in the MEMBERS table, avoiding unnecessary duplication and potential data inconsistency.

Checkpoint S1

It must be unique so that it can uniquely identify each record in the table.

Checkpoint S2

The key field from one table may be included as a foreign key in another in order to create a relationship between the two tables and avoid the need to duplicate data.

Checkpoint S3

Page 137 explains what unstructured data is and how it differs from structured data.

Checkpoint C1

How relational databases minimise data redundancy and inconsistency is explained on page 132 – 135.

Checkpoint C2

Big data is touched upon in Chapter 6, but students will probably need to research the topic in order to tackle this question.

Chapter 4: Computers

4.1 Machines and computational modelling

Exam-style question

Outputs could include: touch screen, speaker, earphones, vibration motor, light/torch.

Inputs could include: touch screen, camera, microphone, motion sensor.

Activity 1

1 a Desktop computer

Inputs could include: keyboard, mouse, scanner, microphone, camera, interactive whiteboard.

Outputs could include: screen, printer, speakers, headphones, interactive whiteboard, projector.

b Washing machine

Inputs could include: sensors (temperature, water level, weight of load, door), touch screen controls, switches.

Outputs could include: motor, pump, door lock, heater, detergent mixer.

2 Computing devices that a central heating installer might use could include:

tablet, exhaust temperature probe, portable printer, fault-finding equipment, mobile phone, sat nav.

3 Categories could include:

manual input devices, automated input devices, visual output, audio output, actuators, switches, sensors, input/output.

Activity 2

1 Challenge C1 on page 82 is a useful starting point for this activity.

2 **Input**: password.

Outputs: message ('Strong' or 'Weak')

Processing: selects which message to display based on the length of the password and the symbols it contains.

Activity 3

This is a great activity for revisiting topics covered in the previous chapters, namely ASCII representation of text (pages 106 –109), logical operators (page 17) and linear searches (pages 34 – 35 and 63).

Here is the program implemented in Python.

```
password = input('Enter your password. ')
#checks length
if len(password) >= 8:
    length = True
else:
    length = False

#checks for an uppercase character
upperCase = False
index = 0

#uses a while loop, so search finishes when the first upper case letter found - more efficient than a for loop
while (index < len(password)) and (upperCase == False):
    if ord(password[index]) >= 65 and ord(password[index]) <= 90:
        upperCase = True
```

```
        else:
            index += 1

    #checks for a symbol
    permittedSymbols = ['!', '#', '&', '*', '+', '?']
    symbol = False
    index = 0
    while (index < len(password)) and (symbol == False):
        if password[index] in permittedSymbols:
            symbol = True
        else:
            index += 1

    #prints message
    if length and upperCase and symbol:
        print('Strong')
    elif (length and upperCase) or (length and symbol) or (upperCase and symbol):
        print('Medium')
    else:
        print('Weak')
```

Checkpoint S1

The input-process-output model is defined on page 140.

Checkpoint S2

Inputs: login details and other text entered via the keyboard, image captured by a camera/webcam

Outputs: image displayed on screen

Processing: The computer would need to authenticate the user's login details, locate the appropriate web page, possibly convert the image to an appropriate file type and optimise it for display on the web, strip out any/all the metadata, upload the image.

Checkpoint C1

Here is the program implemented in Python, with comments used to identify the inputs, output and processing.

```
#inputs
number1 = int(input('Enter first number: '))
number2 = int(input('Enter second number: '))
number3 = int(input('Enter third number: '))

#processing
total = number1 + number2 + number3

#output
print(total)
```

4.2 Hardware

Activity 4

No solution required.

Activities 5 & 6

No solution required. The Key terms box and the boxed text on page 148 summarise the fetch-decode-execute cycle.

Activity 7

A Google search for ARM versus Intel will provide students with plenty of information to complete this activity.

Activity 8

The important thing for students to learn from this activity is that the term ‘spooling’ means putting jobs (in this case print jobs) in a special area called a *buffer*, in memory or on a disc, so that slower peripheral devices, such as printers, can work at their own pace without slowing down the processor.

Students may already have come across the memory buffer register (MBR) when researching the fetch-execute cycle in activities 5 and 6. This is another example of a buffer being used as a temporary store for data.

Activity 9

1 The purpose of cache memory is explained on page 146.

2 The difference between RAM and ROM is explained on pages 144 – 145.

3 This is an extension of activities 5 and 6 on page 144.

4 The most obvious difference is size – components have got considerably smaller and the Pi has fewer of them than the ZX81. USBs weren’t around back in the 1980’s either. Not visible on the photo, the ZX81 had a proper keyboard and used a tape recorder for storage. It didn't come with a screen: customers were expected to use their televisions for this. In today’s money, the ZX81 cost considerably more than a Raspberry Pi and had only a fraction of the processing power, but like the Pi it did inspire a lot of people to try their hand at programming.

Activity 10

1 There isn’t a definitive solution to this activity, which is designed to generate some lively discussion, whilst reinforcing students’ understanding of the role of the clock, cache and RAM.

2 No solution required.

Activity 11

1 The purpose of secondary storage is explained on page 149.

2 Magnetic and solid-state storage are described on pages 149 – 150.

3 There isn’t a definitive ‘best choice’ for any of Monty’s stated activities. Remind students that they need to explain why their chosen secondary storage device is appropriate. For example, a USB memory stick is extremely portable so might be a good choice for storing garden designs to take out to show customers, providing they have a computer. Alternatively Monty could buy a laptop with solid-state storage, which is more robust than a magnetic hard drive and therefore less likely to be damaged if Monty drops the laptop whilst out and about.

4 & 5 No solution required.

Activity 12

No solution required.

Activity 13

1 a The multitude of sensors in an engine management system (EMS) collect data such as throttle position, vehicle speed, engine temperature, engine oil pressure, fuel pressure, oxygen.

The EMS uses actuators to control engine components such as the ignition, fuel injector, fuel pump, dashboard display, gear adjustment.

b The EMS controls the running of an engine by monitoring the engine speed, load and temperature, providing the ignition spark at the right time and controlling the amount of fuel made available to the engine

2 Embedded systems students are likely to encounter in their daily lives include:

TV, remote control, fridge, microwave, washing machine, oven, electronic toys, car, sports band, traffic lights, central heating, burglar alarm, under-floor heating, solar panels, lift.

Exam-style questions

1 Volatile memory loses its content when the power is switched off.

2 The CPU can access cache memory much more quickly than RAM. Frequently used data and instructions are therefore loaded in chunks from RAM into the cache. This means that the CPU isn't slowed down by having to wait for data transfers from RAM.

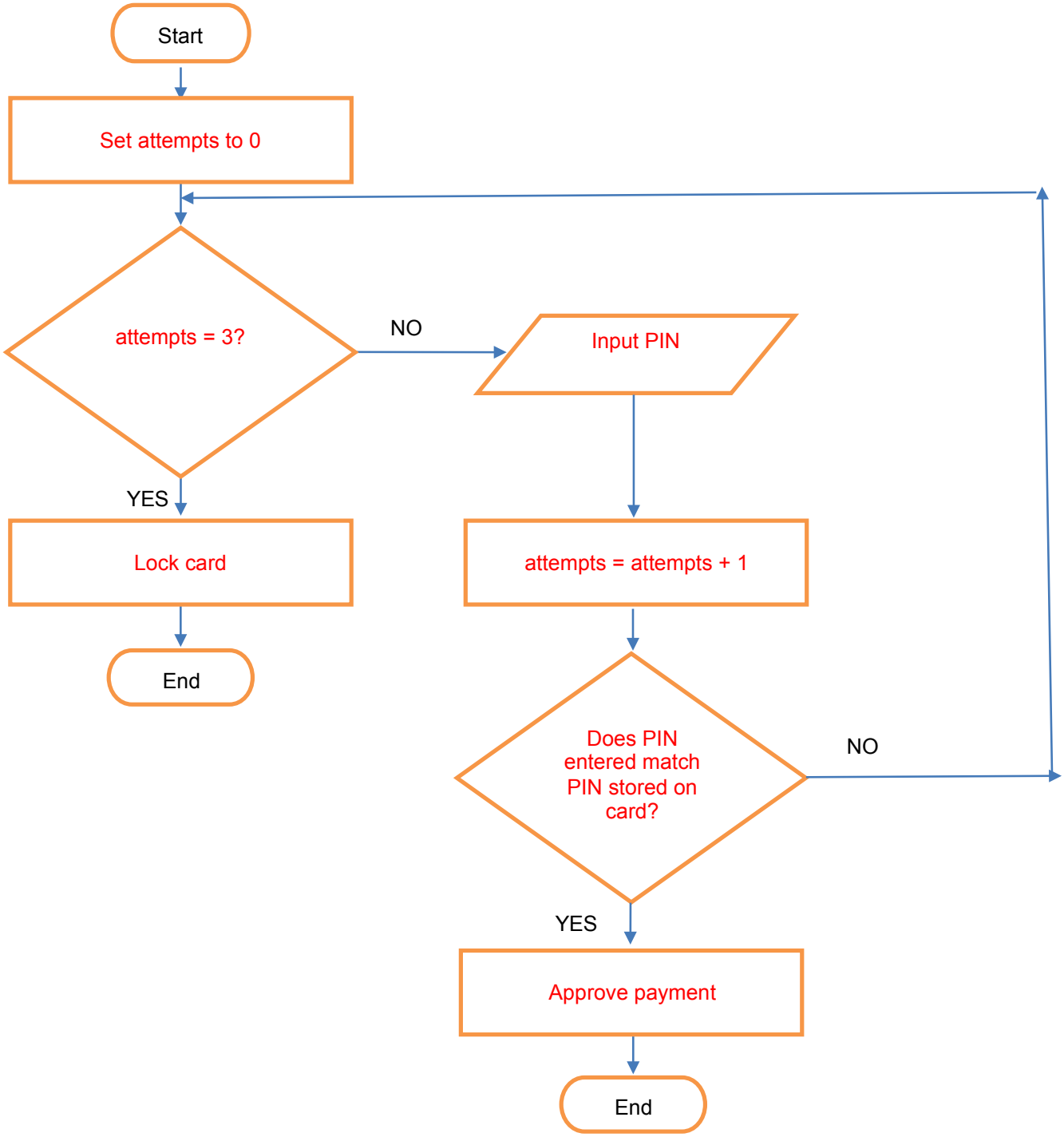
3 The control unit places the memory address of the next instruction onto the address bus (1) and sends a memory read signal (2). The content of this memory address is placed on the data bus (3). The content of the data bus is copied into a special register in the CPU called the Memory Buffer Register (4).

Activity 14

1 This would make a great Raspberry Pi project. Alternatively, students could investigate the traffic light program available on pythoncode.co.uk. Here's a simple text-based program written in Python. It uses the time library module.

```
import time
from time import sleep

start = 'n'
while start != 'y':
    start = input("Press 'y' to start.")

for sequence in range(8):
#Light sequence repeated twice
    light = ['red', 'red and amber', 'green', 'amber'][sequence %4]
    print (light)
    time.sleep(1.5)
```

2



Checkpoint S1

The role of the memory, buses and registers in the fetch-decode-execute cycle is explained on pages 147–148.

Checkpoint S2

When comparing different kinds of secondary storage, students should consider factors such as speed, portability durability and capacity. Points they might make are:

Data is written to and read from a magnetic hard disc more quickly than to/from an optical disc, so backing up and restoring takes less time.

Hard discs are permanently located within a drive so are less portable than optical discs, such as DVDs, which can be removed from the drive when not in use. However, nowadays external hard drives are very light and compact so are reasonably portable. That said, an external hard drive still has moving parts, which might get damaged if the drive is roughly handled or dropped.

In general optical media tends to be more durable than magnetic media.

DVDs offer unlimited storage in the sense that you can use as many as are needed. However, nowadays HDDs can store several terabytes of data.

Nowadays magnetic hard drives have been replaced by solid state drives in many laptops. Solid state drives are faster, lighter, quieter and more resilient than magnetic hard drives. They are, however, more expensive and have a smaller capacity than HDDs.

Checkpoint S3

The 'internet of things' is briefly described on page 153 and revisited in Chapter 5, page 181. Students may want to carry out some additional research if they're interested in this topic.

Checkpoint C1

The operation of a computer with a von Neumann architecture is explained on pages 144 – 147.

Checkpoint C2

All four are types of storage and their characteristics and function are described on pages 146 focus on cache, RAM and ROM.

Checkpoint C3

The purpose of secondary storage is explained on pages 149 – 150. Optical storage is ideal for data that doesn't need to / mustn't be changed, e.g. films, music, financial records, and archives.

Checkpoint C4

This challenge requires students to conduct independent research into embedded systems used in driverless cars. They may prefer to focus on existing assisted driving technologies, such as collision avoidance, cruise control or self-parking.

4.3 Logic

Activity 15

1 The game is over either when the player's score exceeds a million or if the player runs out of 'health' and isn't operating in 'god mode'.

2 ALU

3 Here is the 'starter' program written in Python.

```
yearGroup = input('Enter your year group: ')
grade = input('Enter your grade (9 - 1): ')
target = input('Enter your target grade (9 - 1): ')
if yearGroup == '11' and grade < target:
    print('\nYou should attend the revision class.')
else:
    print("\nThere's no need for you to attend the revision class.")
```

4 Here is the truth table for the statement year = 11 AND (grade < target OR grade > 7)

grade < target	target > 7	grade < target OR target > 7	Year = 11	Year = 11 AND (grade < target OR target > 7)	revision_class
0	0	0	0	0	0
0	0	0	1	0	0
0	1	1	0	0	0
0	1	1	1	1	1
1	0	1	0	0	1
1	0	1	1	1	1
1	1	1	0	0	0
1	1	1	1	1	1

Activity 16

1

```
SET advice TO 'nothing needed'
IF colour = 'yellow' THEN

    IF leaf tips only THEN
        SET advice TO 'magnesium'
    ELSE
        SET advice TO 'nitrogen'
    END IF

ELSE

    IF colour = 'brown' AND size = 'small' THEN
        SET advice TO 'phosphorous'
    ELSE
```

```
        IF colour = 'brown' AND size = 'normal' THEN
            SET advice TO 'potassium'
        ELSE

            IF leaves = 'cracked' OR leaves = 'misshapen' THEN
                SET advice TO 'magnesium'
            END IF

        END IF

    END IF
END IF
```

2 Abstraction was explained in Chapter 1, page 41. Monty's advice is an example of abstraction because it ignores a lot of the details and focuses on just the essential information needed to solve the problem.

Activity 17

1 The code doesn't work because the condition of the first IF statement is met if it is a Saturday, irrespective of whether it's term time or holiday.

Students can see this for themselves by running this Python version of the pseudo-code example given in the Worked Example.

```
alarm = '7:30'
term_time = input('Term time? ')
day = input('Day? ')

if term_time == 'n' or (day == 'Saturday' or day == 'Sunday'):
    alarm = '9:00'
else:
    if term_time == 'y' and day == 'Saturday':
        alarm = '8:00'

print(alarm)
```

2 Here's an amended version of the program that produces the correct outcome.

```
alarm = '7:30'
term_time = input('Term time? ')
day = input('Day? ')

if term_time == 'y' and day == 'Saturday':
    alarm = '8:00'
else:
    if term_time == 'n' or (day == 'Saturday' or day == 'Sunday'):
        alarm = '9:00'

print(alarm)
```

Activity 18

1 a Nothing displayed

b 'Superstar' displayed

c Nothing displayed

2 Answer c would be different.

Exam-style questions

1

```
IF (S = 1 OR B = 1) AND M = 0 THEN
    SEND open TO door_opening_mechanism
END IF
```

2

```
SET standardPrice TO 5
IF age < 4 THEN
    SET price TO 0
ELSE
    IF age < 16 OR age > 65 THEN
        SET price TO standardPrice/2
    ELSE
        SET price TO standardPrice
    END IF
END IF
```

Checkpoint S1

The AND table.

0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

Checkpoint S2

```
IF sensor_reading = dark OR headlamp_switch = on THEN
    SEND switch_on_signal TO headlamps
END IF
```

Checkpoint S3

No solution required.

Checkpoint C1

A	B	A AND B	C	(A AND B) OR C	P
0	0	0	0	0	0
0	0	0	1	1	1
0	1	0	0	0	0
0	1	0	1	1	1
1	0	0	0	0	0
1	0	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

Checkpoint C2

```
P = NOT(A = B)
```

4.4 Software

Exam-style questions

1 The operating system uses scheduling to give each program exclusive use of the CPU for a short time before switching to the next program. This creates the illusion that several applications are running simultaneously.

2 Functions of an operating system include managing memory – sharing RAM between applications and transferring data and programs in and out of memory; managing files and maintaining a file directory structure; providing a user interface; managing peripheral devices such as disc drives and printers; authenticating users and controlling access to programs and data.

Activity 19

1 No solution required.

2 ‘First come first served’ and 'Shortest job first’ are two well-known scheduling algorithms that students could research.

3 No solution required.

Activity 20

No solution required.

Activity 21

A function for simulating throwing a die is provided in Chapter 2 on page 77. There are various ways in which this function could be adapted to increase the probability of throwing a six. One way is to generate a wider range of numbers, e.g. RANDOM(7). If the number generated is less than 6, return the number generated, otherwise return 6.

Checkpoint S1

This activity builds on and reinforces the exam-style questions on page 162.

Checkpoint S2

Different types of user interfaces for different purposes are described on page 162.

Checkpoint S3

Anti-virus software is not essential to the operation of the operating system, isn’t an application in its own right but does do a useful job – hence its categorisation as utility software.

Checkpoint S4

The advantages and disadvantages of using computer models to predict what might happen in the future are explained on page 165. There’s more in Chapter 6 about using computer technology to explore the environmental impact of human activities.

Checkpoint C1

Processing and scheduling are explained on pages 161 – 162.

Checkpoint C2

See S2 above.

Checkpoint C3

This is quite a tough challenge. Students might instead restrict themselves to exploring one or more of the many foxes-rabbits simulation programs available on the web.

4.5 Programming languages

Activity 22

Tasks for which assembly language is particularly well suited include ones that require hardware specific code, such as device drivers; embedded devices, where the size of the code is important; real-time systems where speed is critical etc.

Activity 23

No solution required.

Exam-style questions

1 High level languages use natural language commands such as print and repeat that are easy for humans to understand. However, computers only understand machine code, so Manjit's program will need to be *translated* into that.

2 The table on page 169 summarises the differences.

Checkpoint S1

Machine code is defined in the Key terms box on age 167.

Checkpoint S2

The need to translate programs written in high-level languages is explained on page 168.

Checkpoint S3

The table on page 169 compares the two methods of translation. The main challenge that Momina is likely to face has to do with error detection and correction.

Checkpoint C1

The differences between high–level and low-level language are explained on pages 167 – 169.

Checkpoint C2

The table on page 169 summarises the factors to consider when choosing between a compiler and an interpreter.

Chapter 5: Communication and the internet

5.1 Networks

Activity 1

1 There's a definition of a network on page 172.

2 No solution required.

3 Services provided by a home network include:

- access to shared devices such as printers, NAS drives, central heating controllers etc.
- access to the internet
- file sharing.
-

Exam-style question

A LAN is confined to just one site whereas a WAN covers a much larger geographical area. This is essential for the bank since its branches are likely to be scattered across the country.

Activity 2

1 The difference between a client-server and a peer-to-peer network is explained on pages 174.

2 One major use of peer-to-peer (P2P) networks is file sharing – notably media files. In a P2P network, each 'peer' is an end-user's computer connected to another 'peer' via the internet – without going through an intermediary server. To participate, users must download and install P2P software that searches other connected computers on the P2P network to locate specified content.

Most P2P software has file-sharing features that are turned on by default, making any media files on a user's computer available to others to download. As a result, it's perfectly feasible for a user to unwittingly share files stored on their computer.

Policing P2P networks is difficult. No records of file downloads are kept and even if one source of an illegal file is pinpointed and shut down, there are hundreds/thousands of other computers that have the same file installed. Any one of these can share that file with other peers.

Exam-style question

A version of the star network diagram (Figure 5.4) on page 178 showing three computers (1), a printer (1), a hub/switch (1) and a server (1) is required.

Activity 3

1 No solution required.

2 No solution required.

3 The wireless mesh network topology is a good choice in this situation for a number of reasons.

- Robustness:

  Network nodes 'talk' directly to each other. A big advantage of this decentralized topology is that it is 'self-healing'. If one node can no longer operate, the others can still communicate with each other, directly or through one or more intermediate nodes.

- Cost and simplicity:

  The technology used is relatively cheap and can be simply maintained and extended by users with limited technical expertise.

- Ease of deployment:

  They're relatively easy and quick to deploy.

- Lack of basic communications infrastructure in many developing regions:

  A wireless mesh network can also support telephony and other services.

Activity 4

1 No solution required.

2 Students should look back at the worked example on page 182 to work out how long it would take to download a 300 MB file.

Activity 5

The purpose of HTTP and the difference between HTTP and HTTPS are explained on pages 186 –187. In addition all the major high street banks provide customers with information about how their online transactions are kept secure.

Activity 6

1 The term ‘protocol’ is defined in the Key terms box on page 181.

2 All the information about protocols that students need to know in order to produce a table is described on pages 184 – 185.

3 Students should produce a version of the diagram shown in Figure 5.6 on page 186 and there’s an explanation of the purpose of each layer on page 185.

Checkpoint S1

The difference between a LAN and a WAN is explained on page 173.

Checkpoint S2

The table on page 181 summarises the advantages and disadvantages of wired and wireless networks.

Checkpoint S3

The three main email protocols, SMTP, POP3 and IMAP, are described on page 184.

Checkpoint S4

A protocol stack is a collection of protocols that work together. The TCP/IP stack has four layers. It is described on pages 185 – 186.

Checkpoint S5

There’s a diagram of a bus topology on page 175 (Figure 5.2), a ring topology on page 177 (Figure 5.3), a star topology on page 178 (Figure 5.4) and a mesh topology on page 179 (Figure 5.5).

Checkpoint C1

The supermarket’s stores are likely to be scattered round the country so connecting them via a WAN would allow them to communicate and share data with head office and with each other.

Checkpoint C2

There’s no central server to ‘eavesdrop’ on communications in a P2P network. Files are sent directly to the recipient.

Checkpoint C3

The advantages of using a star network are listed on page 178.

Checkpoint C4

POP3 creates local copies of emails and deletes the originals from the server, with the result that viewing is restricted to the computer on which they’ve been downloaded.

In contrast, IMAP allows users to log into many different email clients or webmail interfaces and view the same emails, because the emails are kept on remote email servers. Furthermore, a user’s inbox, sent, and customised folders look alike, and have the same content, whether they’re checking mail on their smart phone, tablet, or PC.

Having instant access to email wherever you happened to be may well be a factor that has contributed to the popularity of smart phones – though there are of course many others.

5.2 Network security

Activity 7

1 & 2 This activity gives students a good opportunity to revisit working with text files and subprograms. Here is one solution written in Python.

```
def read_in_data():
    rawData = open("users.txt", "r")
    inputData = rawData.readlines()
    rawData.close()

    users = []
    index = 0

    #Splits each line of rawData into a list of strings
    for line in inputData:
        users.append(inputData[index].split(","))
        index += 1

    #Appends each list to the 2 dimensional array users
    for names in users: #Strips out newline characters
        names[1] = names[1].rstrip()

    return users
###end function read_in_data###

def check_user_name():
#Returns EITHER the position in the list of a valid user name OR a zero value
for 'index' that indicates that the name isn't in the list.
    attempt = 1
    nameCorrect = False

    while attempt < 4 and nameCorrect == False:
        print('\nUsername attempt:', attempt)
        nameEntered = input('Enter your username: ')
        valid = False
        index = 0

        while valid == False and index < length:

            if users[index][0] == nameEntered:
                valid = True
            else:
                index += 1

        if valid == False:
            print('Invalid user name.')
            attempt += 1
```

```
        else:
            nameCorrect = True

    return index
 ###end function check_user_name###

def enter_password(position):
#Gives user three goes to enter the correct password
    attempt = 1
    password = ''

    while attempt < 4 and password != users[position][1]:
        print('\nPassword attempt:', attempt)
        password = input('Enter your password: ')

        if password == users[position][1]:
            print ('Correct password entered. Proceed.')
        else:
            print('Password incorrect.')
            attempt += 1

        if attempt == 4:
            print('\nYour account is locked. Contact the system administrator.')

###end function enter_password###

####main program####
users = read_in_data()
length = len(users)
positionInList = check_user_name()
if positionInList < length:
    enter_password(positionInList)
else:
    print('/nContact the system administrator')
```

3 The security of the file holding user names and passwords is crucial. How and where is it stored? Who has access to it? Who has the right to read the file? Who can modify it?

One simple way of making this program more secure would be to conceal the password that the user enters. Limiting the number of attempts a user has to enter a valid user name and matching password is already a step in the right direction, but letting them have three goes may be not such a good idea.

Activity 8

1 Authentication is explained on page 191. A web search will provide students with plenty of further information about secure passwords. They may want to revisit Activity 3 on page 142, in which they produced a password checker program.

2 The servers should be located in an area where physical access can be controlled, preferably somewhere that is safe from flooding. Obviously the technicians will need access to the servers – but arguably no-one else.

Activity 9

Factors students should take into account are described on pages 193 – 194.

Activity 10

1 No solution required.

2 No solution required.

Activity 11

1 No solution required.

2 Admitting to having been the target of a successful cyberattack could have a negative impact on a business's reputation, with serious financial consequences.

Activity 12

By not applying a patch, the user increases their chance of becoming a victim of a malware attack that exploits a known flaw in the web browser software in order to do its work.

Exam-style question

Phishing is explained on page 196. Students may want to amplify their answers by conducting a Google search for email + phishing + bank.

Activity 13

1 No solution required.

2 No solution required.

Checkpoint S1

Authentication is explained on page 191.

Checkpoint S2

Access control is explained on pages 191 – 193.

Checkpoint S3

Social engineering is defined on page 196, along with a description of common social engineering techniques.

Checkpoint C1

Students should refer to pages 196 – 197 where forms of cyberattack are described.

Checkpoint C2

Reasons might include:

- More cyberattacks are now being reported.
- More small businesses are now operating online and they are often weak target, particularly vulnerable to cyber crime.
- Organised hacking crime groups have replaced individual lone hackers as the main source of cyberattacks. Thus cybercriminals are becoming more expert and using ever more sophisticated methods.
- Because many organisations are unwilling to admit to having been hacked, they don’t collaborate sufficiently to fight cybercrime effectively.

5.3 The internet and the World Wide Web

Activity 14

Services that use the WWW include: messaging, file sharing, video conferencing, e-commerce, media streaming, video gaming, voice calls, software as a service, cloud storage, health monitoring, the internet of things.

Exam-style question

IMAP

Exam-style question

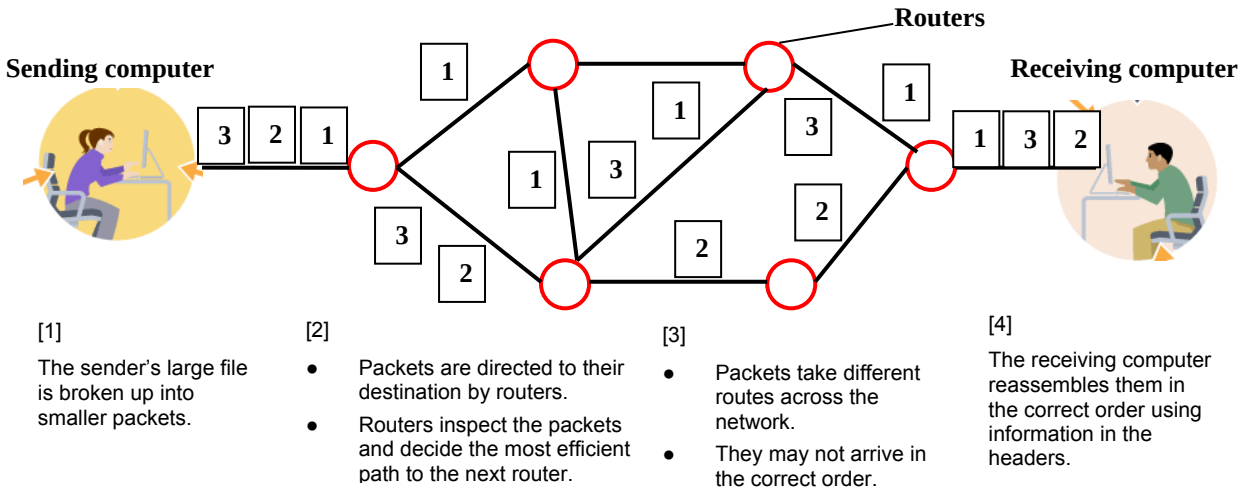
The internet is a global system that enables multiple computers to connect to each other – a network of networks. Many different kinds of data are transferred over this network and many applications make use of this system. The World Wide Web (WWW) is one of them.

Every publicly accessible device on the WWW is on the internet, but not every device on the internet participates in the WWW.

The WWW is a system of hyperlinked pages written in HTML. These are rendered on client computers by web browsers. Browsers use a system of URLS to locate where the data is stored and HTTP or HTTPS to transport it. The 'Web' in the name refers to the hyperlinks between pages connecting them together like a spider's web.

Activity 15

1 The diagram should illustrate the process described at the top of page 203.



[1]

The sender's large file is broken up into smaller packets.

[2]

- Packets are directed to their destination by routers.
- Routers inspect the packets and decide the most efficient path to the next router.

[3]

- Packets take different routes across the network.
- They may not arrive in the correct order.

[4]

The receiving computer reassembles them in the correct order using information in the headers.

2 The internet is a mesh network. Mesh networks were discussed earlier in this chapter on pages 179 – 180, including what makes them so fault tolerant.

Checkpoint S1

There's a definition of the internet on page 202.

Checkpoint S2

The WWW is described on pages 203 – 204 and the exam-style question at the top of page 205 asks about the difference between the internet and the WWW (see answer above).

Checkpoint S3

What happens when a user enters a web address into a browser is explained on page 204.

Checkpoint C1

Students should include in their explanation HTTP, HTTPS, TCP/IP and HTML. All of these protocols are covered in this chapter.

Checkpoint C2

This is a great topic for class discussion. Factors students may want to consider include:

- The founding principle of the web that anyone, anywhere can use it without paying a fee or having to ask for permission
- Its ease of use
- The growth/availability of web-enabled devices
- Developments in networking, telecommunications and mobile phone technology
- The dot-com boom of the 90's
- The growth in popularity of social networking.

Chapter 6: The bigger picture

6.1 Computing and the environment

Activity 1

Laptops are designed to be more energy efficient than desktops because they rely on battery-power. Factors that contribute to their energy efficiency include:

- A low-voltage/ lower performance processor that generates less heat and therefore requires less cooling
- Components such as the graphics processing unit (GPU) and the network interface controller (NIC) are integrated into the chip that contains the processor rather than each being on a separate chip.
- A small, energy-efficient power supply unit (PSU)
- A solid-state hard disc drive for storage
- Various power-management features designed to conserve the battery
- Rather than having an internal DVD drive, an external drive is plugged in and uses power only as and when required.

Activity 2

1 In 2015 data centres consumed 416.2 terawatt hours of electricity – representing 3% of the world's total energy usage.

Data centres use energy to power the huge number of servers they use and the giant fans needed to cool them down.

2 Measures that can be used to make data centres more environmentally friendly include:

- Reducing dependency on energy generated from fossil fuels by making more use of renewable energy.
- Locating them in regions where it's cold most of the year, so that the amount of energy needed to cool down them down is significantly reduced. Facebook has a huge data centre in the far north of Sweden close to the Arctic Circle.
- Adopting energy-efficiency measures, such as using a hot-aisle/cold-aisle configuration to increase cooling system efficiency; using blanking panels to minimize recirculation of hot air; and sealing the floor to prevent cooling losses.
- Investing in research to develop a new, less energy-hungry alternative to silicon-based data storage. It has been suggested that graphene has enormous potential to do this.
- Ration internet usage – possibly by imposing a tax on uploading data – and/or educate users so that they behave in a more environmentally responsible fashion. Simply switching from colour to black and white photos when uploading to social media sites rather would have a significant impact.

Activity 3

Computers 4 Africa (www.computers4africa.org.uk) and Computer Aid International (www.computeraid.org) are two initiatives that make use of pre-owned computing technology.

Activity 4

Summary of the environmental impact of computing technology and actions that can be taken to reduce it.

Environmental impact	Actions
Computing technology consumes massive amounts of energy, much of it generated from non-renewable fossil fuels, which is contributing to global warming.	Develop more energy-efficient components/practices and switch from non-renewable to renewable sources of energy.
Disposal of cast-off computing technology in landfill sites, which exposes people living nearby to harmful materials and could result in toxic substances leaking into the ground and polluting the air.	Reuse old computer technology rather than consign it to landfill sites. Manage disposal more effectively so that dangerous materials are disposed of safely and reusable materials are recovered.
Use of hazardous materials in the manufacturing process, which put workers at risk.	Legislate to force producers of computing technology to replace hazardous materials with safer alternatives
Use of valuable, non-renewable materials in the manufacturing process some of which are already in short supply.	Legislate to force producers to use alternative materials and to ensure that valuable materials are recovered from redundant computing technology and reused.

Exam-style question

This is an extended answer question. The indicative content that students should include in their response is listed in the summary on page 212. However, emphasise to students that a bulleted list is not an appropriate response to this type of question. In the Paper 1 exam candidates will be expected to produce a well-structured essay-style response to this sort of question and use appropriate evidence to support their conclusions.

Checkpoint S1

The table on page 208 lists some of the hazardous substances used in the manufacture of computing technology.

Checkpoint S2

The Restriction of Hazardous Substances (RoHS) Directive (adopted by the UK in 2013).

Checkpoint S3

The danger of dumping e-waste in landfill sites is explained on page 210.

Checkpoint S4

See solution to Activity 2 above.

Checkpoint S5

Ways in which computing technology is helping to preserve the environment are described on pages 211 – 212.

Checkpoint C1

The health risks associated with raw material extraction and the manufacture of computing technology components are explained on pages 207 – 208 and those associated with disposal on page 210.

Checkpoint C2

The use of computing technology to make buildings more energy efficient is outlined on page 211. Other ways that computing technology can help to reduce energy consumption include:

- Use of smart meters, thermostats, and sensors – part of the ‘internet of things’ – to monitor energy consumption raising ‘energy awareness’ and reducing ‘energy wastage’
- Vehicle technologies such as adaptive cruise control that significantly reduce a vehicle’s fuel consumption and emissions
- Use of GPS tracking, wireless communication and adaptive traffic control systems to ease traffic congestion and improve journey efficiency, which – in turn – reduces fuel consumption
- Use of automation and process control systems to make manufacturing processes more energy efficient
- Modifying the design of hardware to make computer technology more energy efficient
- Writing energy-efficient sorting and searching algorithms

6.2 Privacy

Activity 5

1 No solution required.

2 The Safe Harbour Agreement (2000) made between the EU and the US government allowed US companies such as Facebook and Google to self-certify that they would protect any EU citizens' data that they transferred and stored in US data centres.

In 2015 the EU Court of Justice decided that the agreement was invalid because it did not adequately protect consumers. Nowadays any company wanting to store personal information about EU citizens in US data centres must guarantee an adequate level of protection in line with EU rules.

Activity 6

More data means more information, which can be used to create more targeted, focused and efficient services. Examples of how society is benefiting from big data analysis include:

- In retail, companies are analysing large volumes of customer data to provide a smarter shopping experience. This could be through location-based promotions or targeted advertising, or by making the supply chain more efficient.
- In healthcare, big data analytics is being used to predict outbreaks of diseases such as dengue and malaria.
- Disaster relief organisations are using big data analysis to extract insights and key trends. This provides faster relief and helps staff on the ground to solve problems before they escalate into a crisis.
- The Kepler telescope captures data on 200,000 stars every 30 seconds. Analysis of this mountain of data has led to the discovery of the first earth-like planets outside our solar system.

Activity 7

Obvious situations in which an invasion of privacy may be justified are those associated with keeping society safe from terrorist attacks, crime prevention, improving road safety, and protection of key installations such as nuclear power stations, military establishments etc. As well as citing two of these, students should explain why their severity outweighs the loss of privacy involved.

Activity 8

Benefits of location-based services include:

- More efficient route planning and navigation systems that use real-time information to avoid hold-ups and congestion
- People on the move can get relevant information about shops, hotels, restaurants etc. in the vicinity and see if any of their friends are close by or have left tips about the area.
- Photographs can be geo-tagged with the precise longitude and latitude of where they were taken.
- Parents can monitor their offspring's whereabouts helping to keep them out of danger.
- Conservationists can monitor animal/bird migration patterns.
- The response time of emergency services is improved by the availability of accurate and reliable location information helping to save lives.

Risks of location-based services include:

- They make it easy for predators to locate and stalk their victims.
- They're a useful source of personal information for criminals intent on identity theft.
- They can be used to establish when a home owner is most likely to be away from home, i.e. the optimum time to carry out a burglary.
- They can be used to infer other sensitive information about an individual, such as their religious affiliation or political standpoint.

Activity 9

In August 2015 hackers used a distributed denial of service (DDOS) attack to swamp Carphone Warehouse's online systems with junk traffic. This was a smokescreen: they broke in and stole the personal details of 2.4 million customers, including email addresses, bank details and – in some cases – encrypted credit card details.

There was a huge public backlash. Customers, not surprisingly, blamed the company for not having adequate security measures in place.

Affected customers were put at increased risk of identity theft and were advised to inform their bank and credit

card company, monitor account activity closely and check their credit rating to make sure nobody applied for credit in their name.

The company suffered both damage to its reputation and considerable financial loss, as customers opted to take their business elsewhere.

Checkpoint S1

How personal data ends up stored on third party databases is outlined on page 213.

Checkpoint S2

The problems associated with personal data falling into the wrong hands are described on pages 214 and 216.

Checkpoint S3

The table on page 216 provides an overview of privacy-enhancing tools.

Checkpoint S4

The activities covered by the Computer Misuse Act (1990) are listed on pages 216 – 217.

Checkpoint C1

The benefits of revealing personal information are outlined on page 214.

Checkpoint C2

See solution to Activity 7 and the paragraph on Big Data on page 215.

6.3 Digital inclusion

Activity 10

Some useful starting point are:

- 'How mobile phones are changing the developing world', in a series of blog posts published by UNICEF Innovation: https://blogs.unicef.org/innovation/how-mobile-phones-are-changing-the-developing-world/
- 'Emerging Nations Embrace Internet, Mobile Technology', written by the Pew Research Center: http://www.pewglobal.org/2014/02/13/emerging-nations-embrace-internet-mobile-technology/

Activity 11

Actions a government can take to reduce digital exclusion include:

- Improving high speed broadband connectivity so that everyone has access to the internet irrespective of where they live
- Making connection and access to the internet affordable for all
- Providing training courses aimed at helping people to develop digital skills
- Making computing a national curriculum subject which all pupils must study.
- Providing computer/internet access points in public locations, such as libraries, community centres, cafes and pubs.
- Making government services, such as car registration and licensing, TV licence renewal, and tax returns, available online and encouraging people to use them.

Checkpoint S1

Technology empowered: Having affordable access to computing technology and the necessary skills to take advantage of it.

Technology excluded: Not having access to computing technology and/or the skills to use it.

Checkpoint S2

The table on page 218 highlights some of the drawbacks of being technology-excluded.

Checkpoint S3

Factors contributing to the digital divide are listed on page 219.

Checkpoint S4

The solution to Activity 10 illustrates one way of achieving connectivity in regions that have a poor telecoms infrastructure. And the 'Did you know?' box above Activity 10 flags up plans by Facebook to use drones and satellites to overcome the problem.

Checkpoint C1

See the solution to Activity 11 above.

6.4 Professionalism

Exam-style question

The BCS code of conduct for computer scientists stipulates that they should not withhold information on the performance of systems. Therefore, the programmer should inform their manager immediately. Furthermore, the Code also states that they must avoid injuring others. If the testing software is producing faulty information about exhaust emissions it could also endanger human health, which is another reason for the programmer to take action to flag up the problem.

Activity 12

No solution required.

Checkpoint S1

Relevant aspects of the BCS Code of Conduct are listed on page 221.

Checkpoint C1

Professionalism in the context of computer science is discussed on page 221.

6.5 The legal impact

Exam-style question

A patent gives the patent holder the exclusive right for 20 years to make, use and sell an invention. This encourages inventiveness by ensuring that the owner of the patent (usually the employer of the inventors) gets recognition and benefits financially from the invention. However, in recent years big companies such as Apple and Samsung have been embroiled in long and expensive legal battles over alleged patent infringements. To defend a patent is very costly. There is an argument that the money spent on legal fees would be better invested in research and development. Patent law encourages companies to keep new inventions secret and block others from using them for 20 years. If inventions were shared from the outset, the pace of technological progress and innovation would be accelerated.

Activity 13

1 No solution required.

2 Copyright and patents are explained on page 223. Students may want to include an overview of licensing (explained on page 224) in their podcast.

Activity 14

Proprietary software	Open-source software
User licences apply strict conditions on the way in which the software is used and distributed.	Under the licence users can pass on the software to other users for no charge.
The source code is protected. Users aren't allowed to modify it.	Users can study the source code to see how the software works and modify it however they like.
The software is developed professionally and extensively tested prior to release. Any bugs that come to light thereafter are quickly fixed.	The software may not appear as professional or have such a user-friendly interface. It's often released before it has been thoroughly tested so bugs may well have slipped through the net.
Support is provided to keep customers happy so that they will keep using the software. Support and updates may be expensive.	There might be little or no technical support available, but there's likely to be a community of dedicated enthusiasts who are willing to provide help and support free of charge.
Software is developed for the majority of users and may not meet individual needs.	The software can be modified to meet a specific need.
The software must be paid for.	The software is free to use.
The software has been rigorously tested and is usually less likely to have any technical weaknesses.	Criminals may be able to exploit vulnerabilities in the code.

Checkpoint S1

Software licensing is explained on page 224.

Checkpoint S2

The difference between open-source and proprietary software is outlined on pages 224 – 225.

Checkpoint C1

The protection provided by a patent is explained on page 223.

Checkpoint C2

The Top tip on page 223 lists relevant legislation.