```
        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:	$10 / 2$	$= 5$

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'
```

```
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

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:
```

```
    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)
```

```
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)
```

(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.')
```

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.')
```

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:
```

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.

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.

3.5 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, …)

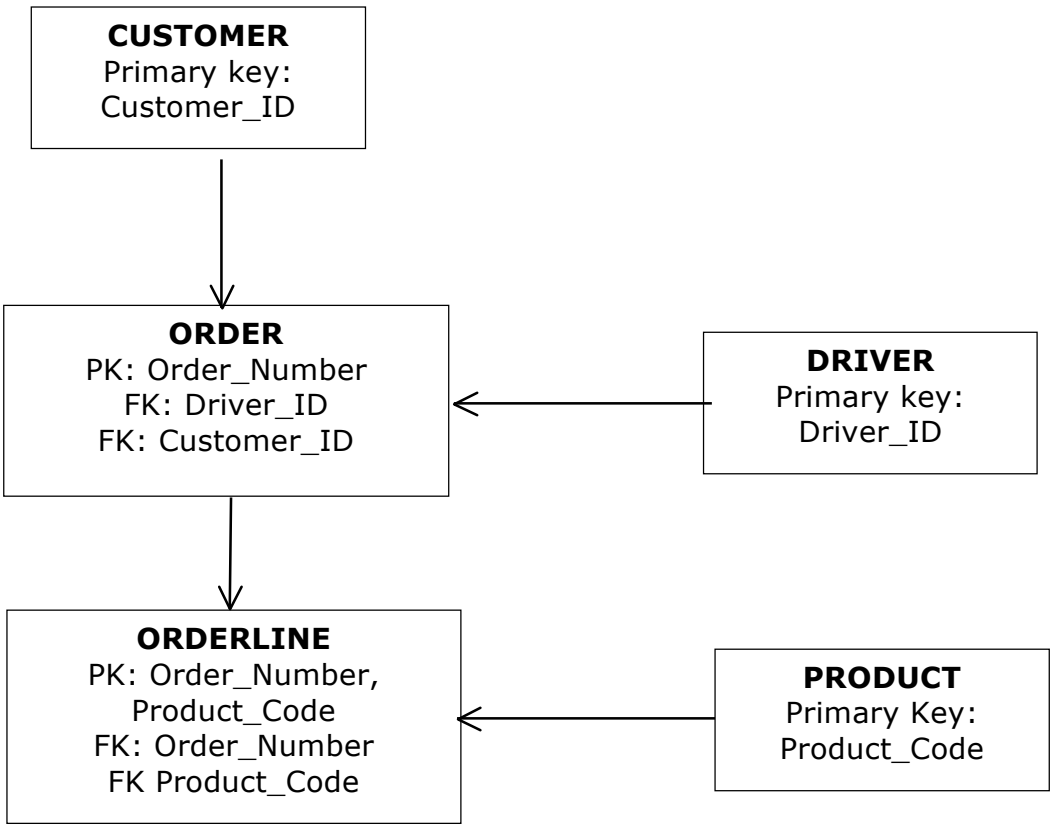
DRIVER(**Driver_ID**, Driver_Name, …)

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.

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.

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 identify 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.