GCE A LEVEL

eduqas
Part of WJEC

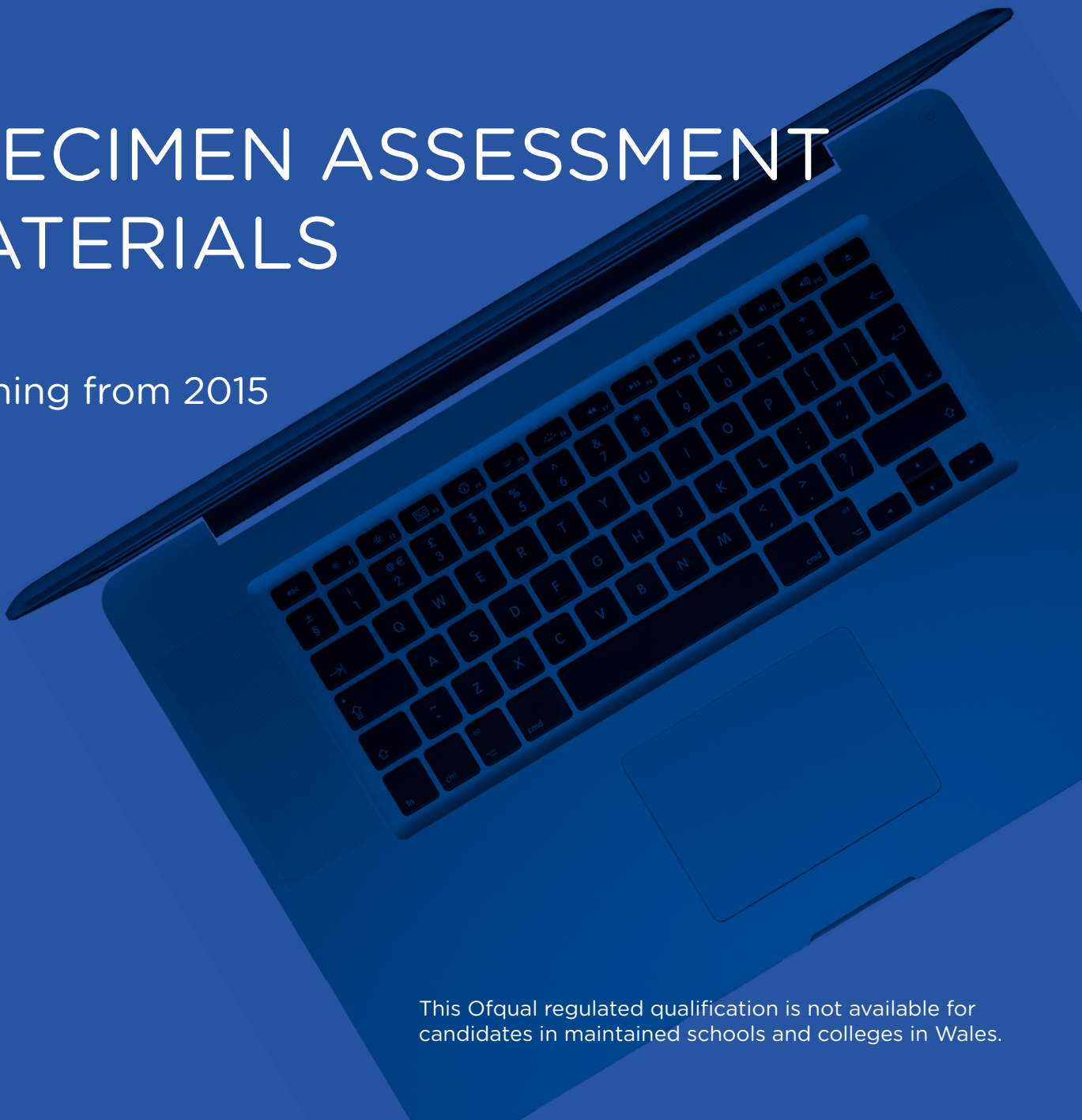
WJEC Eduqas GCE A LEVEL in

COMPUTER SCIENCE

ACCREDITED BY OFQUAL

SPECIMEN ASSESSMENT MATERIALS

Teaching from 2015



wjec cbac

This Ofqual regulated qualification is not available for candidates in maintained schools and colleges in Wales.

Contents

A LEVEL COMPUTER SCIENCE

COMPONENT 1

Programming and System Development

SPECIMEN PAPER

2 hours 45 minutes

ADDITIONAL MATERIALS

In addition to this examination paper, you will need a 16 page answer book.

INSTRUCTIONS TO CANDIDATES

Answer **all** questions.

Write your answers in the separate answer book provided.

INFORMATION FOR CANDIDATES

The number of marks is given in brackets at the end of each question or part-question; you are advised to divide your time accordingly.

The total number of marks is 100.

You are reminded of the need for good English and orderly, clear presentation in your answers.

No certificate will be awarded to a candidate detected in any unfair practice during the examination.

*Answer **all** questions*

1. A binary tree can be constructed using the following rules:

   Rule 1. The first item becomes the root node;
   Rule 2. Items earlier or at the same position in the alphabet follow the left pointer;
   Rule 3. Items later in the alphabet follow the right pointer.

   (a) Draw a representation of a dynamic binary tree with pointers using the following data:

   Newport, Canterbury, Oswestry, Warrington, Rugby, Bath, Derby [2]

   (b) Show how the above tree could be represented using a two dimensional array. [3]

   (c) Make the following amendments to the tree represented as a two dimensional array:

   (i) Insert *Newquay* into the tree; [3]
   (ii) Delete *Warrington* from the tree. [1]

   (d) Give one advantage and one disadvantage of using a binary tree to store data compared with a linked list. [2]

   (e) Giving examples, compare a balanced and an un-balanced binary tree and evaluate their effectiveness to solve problems by comparing the maximum number of comparisons to locate an item in each of these trees. [4]

2. In the programming industry, professional Codes of Conduct are used extensively and they set rules that should be followed by programmers.

   State **five** reasons why programmers are required to follow rules under a professional Code of Conduct. Each reason must relate to a specific rule. [5]

3. Below is a segment of code from a high level language.

```
Total = Total + NumInput;
print(Total);
Count = Count + 1;
print(Count);
```

Give an example of a reserved word table used by a compiler and construct a user identifier table that could be used by a compiler to translate the segment of code above into a stream of Hex tokens. [4]

Use the tables to translate the following line of code into a stream of Hex tokens

Total = Total + NumInput [1]

4. (a) Clearly showing each step, simplify the following Boolean expression using De Morgan's Laws and Boolean identities:

$$\overline{A \cdot B} + A \qquad [3]$$

(b) Clearly showing each step, simplify the following Boolean expression:

$$A + B \cdot (A + B) + A \cdot (\bar{A} + B) \qquad [5]$$

5. Below is an algorithm.

```
Algorithm CalculateArea

        declare subprocedure FindArea( R, A ) {procedure to calculate area of a circle}

        Pi = 3.142

        startproc
                set A = Pi * R * R
        endproc

        declare subprocedure MainProg

        Area is real
        Radius is real

        startproc
                output “Type in the radius”
                input Radius
                call FindArea (Radius, Area)
                output “The area is ”,Area
        endproc


start
        call MainProg
end
```

(a) Explain why it is good programming practice to use constants, meaningful names for variables and annotation in computer programs. [3]

(b) From the above algorithm, identify a value and a reference parameter. [1]

(c) Explain the difference between a value and a reference parameter giving an advantage of using a value parameter. [3]

6. Explain the facilities available in a typical IDE that would assist when debugging a computer program. [4]

7. The email addresses of staff at National Bank are made up of a first name, followed by a full stop, followed by a surname, followed by a single digit, followed by the @ sign, followed by nb.co.uk

   All first names and surnames consist of lower case letters only, and can be of any length.

   (a) Produce appropriate syntax diagrams to define an email address at National Bank. [5]

   (b) Produce an appropriate Backus-Naur Form (BNF) definition for an email address at National Bank. [4]

   (c) Explain why programmers find both syntax diagrams and BNF notation preferable methods for describing the syntax of programming languages compared to natural English. [3]

8. A software company has produced a stock control system for a small retailer. Using examples, describe different types of maintenance that should be carried out on the stock control system.

   [6]

9. Below is a section of a program that makes use of a recursive algorithm.

```
declare subprocedure RecursionAlgo(IN Num)
NewNum is integer
startproc
        if Num = 0 then
                output "Num is zero"
        else
                output "Num is ", Num
                set NewNum = Num - 1
                call RecursionAlgo(NewNum)
        endif
endproc
startmainprog
        input NumInput
        output "Starting recursion"
        call RecursionAlgo(NumInput)
        output "Recursion ended"
endmainprog
```

(a) Identify the main features of a recursive algorithm. [2]

(b) Describe, giving reasons, two problems that a programmer might encounter when using a recursive algorithm. [4]

10. Explain the terms class and inheritance in object-oriented programming. [3]

11. Write algorithms in pseudo-code that will:

(a) input 10 integers into an array;
search the array of 10 integers for the largest value;
output the largest value. [3]

(b) input 10 integers into an array;
sort the array of 10 integers into ascending order;
output the 10 integers in ascending order. [6]

12. Below is an algorithm that calculates the product of two square matrices A and B.

Both matrices have dimensions N by N where N is the number of rows and columns in each matrix.

```
for i = 1 To N
   for j = 1 To N
      for k = 1 To N
         set Answer( i, j ) = Answer( i, j ) + A( i, k ) * B( k, j )
      next k
   next j
next i
```

(a) Determine how many multiplication calculations will be carried out when N is equal to:

10
100

[1]

(b) Evaluate the efficiency of the algorithm and, using Big O notation, determine the growth rate for the time performance. Your answer should refer to the addition and multiplication calculations performed by the algorithm. [5]

(c) Using Big O notation, determine the growth rate of memory space used by this algorithm. [2]

13. A software company encourages their programming teams to write programs using modules.

Discuss implications of writing programs in modules.

You should draw on your knowledge, skills and understanding from a number of areas across your Computer Science course when answering this question. [12]

End of Paper

A LEVEL COMPUTER SCIENCE

COMPONENT 2

Computer Architecture, Data, Communication and Applications

SPECIMEN PAPER

2 hours 45 minutes

ADDITIONAL MATERIALS

In addition to this examination paper, you will need a 16 page answer book.

INSTRUCTIONS TO CANDIDATES

Answer **all** questions.

Write your answers in the separate answer book provided.

INFORMATION FOR CANDIDATES

The number of marks is given in brackets at the end of each question or part-question; you are advised to divide your time accordingly.

The total number of marks is 100.

You are reminded of the need for good English and orderly, clear presentation in your answers.

No certificate will be awarded to a candidate detected in any unfair practice during the examination.

Answer all questions

1. *(a)* In a certain computer, integers are stored using sign and magnitude representation and 16 binary digits. The left hand bit is set to zero for a positive number.

Find the integer represented by the binary number 1000000000001001 [1]

(b) In another computer, integers are stored using two's complement representation. Explain, using an example, how the two's complement of a binary number is derived. [2]

(c) In another computer, real numbers are stored in floating point form using 20 bits as shown below:

Mantissa	Exponent
(16 bits in two's complement form. The binary point in the mantissa is immediately after the left bit.)	(4 bits in two's complement form)

Convert the number 22.75 to this floating point form. [2]

(d) Evaluate the benefits and drawbacks of floating point form compared with integer form. [2]

2. *(a)* Explain the sequence of operations which will occur during the fetch phase of the fetch execute cycle, making clear the role of the PC (program counter), the MAR (memory address register) and the MDR (memory data register). [3]

(b) A simple computer has a number of 16 bit registers. Its assembly language set includes the following:

LOAD R,X	loads register R with the contents of address X
STOR R,X	stores contents of register R in address X
XORR R1,R2	executes a bit-wise XOR operation on the contents of register R1 and R2, storing the results in register R1

Explain what the following fragment of code does, by showing the content of registers and addresses at each step. Make clear the contents of register 1 when the fragment has finished execution.

Assume that the address d1 initially contains 1001 0110 1110 1100
Assume that the address d2 initially contains 1110 1011 1010 0011

```
LOAD 1,d1
LOAD 2,d2
XORR 1,2
STOR 1,d3
XORR 1,2
```

[4]

3. *(a)* Explain the term foreign key as used in a relational database system. [2]

(b) If a database is already in second normal form, describe the step necessary to ensure that it is in third normal form. [1]

(c) Two tables have already been created in a database using SQL commands.

They are:

STUDENTS

StuNum	StuName	DateBirth	TutNum
12675	Brown	02-Nov-96	307
13670	Abbakari	01-Jun-95	378
13777	Walker	23-Oct-95	307
14156	Pang	29-Feb-96	307
14238	Hartford	18-Jan-95	378

TUTORS

TutNum	TutName	RoomNum
307	Harris	106
345	Lester	113
378	Suleiman	113

(i) Write an SQL command to output the names and tutor numbers of all the students. [1]

(ii) Write an SQL command to output the names of students who have the tutor with number 378. [1]

(iii) Write an SQL command to change the tutor number of all those students whose tutor number is currently 378. Their new tutor number should be 345. [2]

(iv) Write an SQL command to output the names of all students who have the same tutor as student 13777. [2]

(v) Write an SQL command to output the names of all the students and their tutors' names. [2]

(vi) Write an SQL command to create a new table PHONES to contain the telephone number for each room (Assume there is only one telephone per room) [2]

(vii) Write an SQL command to enter the following data into the table you have created in part (vi): room 106 should have telephone number 625 and room 113 should have telephone number 670. [2]

(viii) Write an SQL command which will output the names of the students and their dates of birth in descending age order. [2]

4. *(a)* Access to computer files may be sequential or indexed sequential.

Compare these two types of access. [5]

(b) Computer files may also allow direct (random) access.

(i) A certain random access file system uses a hashing algorithm. The key field in this case is an integer in the range 001 to 999. The hashing algorithm operates by:

- squaring the key field, resulting in a six digit number with zeroes added to the left if necessary
- taking the middle two digits of these six digits

Calculate the output of this hashing algorithm for a key field of value 123 [1]

Calculate the output of this hashing algorithm for a key field of value 223 [1]

(ii) Another hashing algorithm operates on the same data by:

- squaring the key field, resulting in a six digit number with zeroes added to the left if necessary
- taking the right hand two digits of these six digits

Calculate the output of this hashing algorithm for a key field of value 123 [1]

Calculate the output of this hashing algorithm for a key field of value 223 [1]

(iii) Discuss these two hashing algorithms [2]

5. The following algorithm fragment is not suitable for parallel processing:

```
…….
set a = b * c
set d = a * e
set f = a + d
…….
```

Write an algorithm fragment which is suitable for parallel processing, to gain both marks, your fragment should demonstrate at least three calculations which can be carried out in parallel. [2]

6. *(a)* Define simplex, half duplex and full duplex data transmission. [3]

(b) Explain what is meant by a data collision on a network. Describe what should happen when this occurs. [3]

7. *(a)* (i) Calculate the effect of carrying out an arithmetic shift left by two places on the eight-bit positive integer 00001111 and state the effect of this operation on the number. [2]

(ii) If an arithmetic shift left by two places was carried out on an eight-bit register containing the positive integer 01001111, a problem would arise. Describe the problem and how it can be rectified. [2]

(b) Explain, using an example, why hexadecimal notation is often used to represent binary numbers. [2]

8. *(a)* An estate agent has branches in many towns. It uses a distributed database. Describe why the estate agent might wish to use a distributed database and describe one difficulty that the estate agent will experience in using a distributed database. State what is actually distributed in a distributed database. [3]

(b) Explain the term Big Data. [1]

(c) A supermarket chain uses data mining techniques on data gathered from its customer loyalty card scheme. Explain what is meant by data mining in this instance and evaluate the benefits of this system to the supermarket. [4]

9. *(a)* Compare two different types of real-time operating system, giving an example of an application which would use this type of operating system, in each case. [6]

(b) Describe the functions carried out by a multiprogramming system. [4]

(c) A multiprogramming operating system is carrying out a low priority task. During the execution of the low priority task a high priority interrupt is generated immediately followed by a medium priority interrupt.

Explain how the operating system would deal with this situation. [4]

10. *England and World Carriers* employs a number of drivers for its vehicles. Each driver is based at just one depot (for instance Avonmouth or Sheffield) and each depot has a full address, telephone number and a supervisor. Each driver has an ID number, name and home address, and is based at just one depot. Each vehicle has a registration number, make and model, and is also based at just one depot. Each supervisor (who may be the supervisor for more than one depot) has an ID number and a home address.

A database is required by *England and World Carriers* to store the data required.

(a) Design an entity-relationship diagram to illustrate the above situation. [3]

(b) Design a database for the above situation in third normal form. [6]

11. When designing a major computer-based control system in a large safety critical system, the designers need to consider both operational and safety issues. There is a particular concern over possible malicious attacks on the computer system.

Discuss the general issues related to safety critical control systems of this type and discuss approaches that can be adopted to ensure system security.

You should draw on your knowledge, skills and understanding from a number of areas across your Computer Science course when answering this question. [13]

End of Paper

Component 1 – Programming and System Development

Mark Scheme

Guidance for examiners

Positive marking

It should be remembered that learners are writing under examination conditions and credit should be given for what the learner writes, rather than adopting the approach of penalising him/her for any omissions. It should be possible for a very good response to achieve full marks and a very poor one to achieve zero marks. Marks should not be deducted for a less than perfect answer if it satisfies the criteria of the mark scheme.

For questions that are objective or points-based the mark scheme should be applied precisely. Marks should be awarded as indicated and no further subdivision made.

For band marked questions mark schemes are in two parts.

Part 1 is advice on the indicative content that suggests the range of computer science concepts, theory, issues and arguments which may be included in the learner's answers. These can be used to assess the quality of the learner's response.

Part 2 is an assessment grid advising bands and associated marks that should be given to responses which demonstrate the qualities needed in AO1, AO2 and AO3. Where a response is not credit worthy or not attempted it is indicated on the grid as mark band zero.

Banded mark schemes

Banded mark schemes are divided so that each band has a relevant descriptor. The descriptor for the band provides a description of the performance level for that band. Each band contains marks.

Examiners should first read and annotate a learner's answer to pick out the evidence that is being assessed in that question. Once the annotation is complete, the mark scheme can be applied.

This is done as a two stage process.

Stage 1 – Deciding on the band

When deciding on a band, the answer should be viewed holistically. Beginning at the lowest band, examiners should look at the learner's answer and check whether it matches the descriptor for that band. Examiners should look at the descriptor for that band and see if it matches the qualities shown in the learner's answer. If the descriptor at the lowest band is satisfied, examiners should move up to the next band and repeat this process for each band until the descriptor matches the answer.

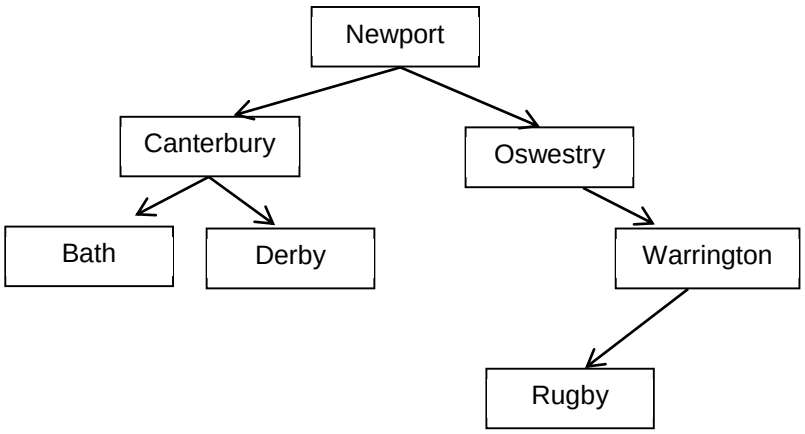
If an answer covers different aspects of different bands within the mark scheme, a 'best fit' approach should be adopted to decide on the band and then the learner's response should be used to decide on the mark within the band. For instance if a response is mainly in band 2 but with a limited amount of band 3 content, the answer would be placed in band 2, but the mark awarded would be close to the top of band 2 as a result of the band 3 content. Examiners should not seek to mark candidates down as a result of small omissions in minor areas of an answer.

Stage 2 – Deciding on the mark

Once the band has been decided, examiners can then assign a mark. During standardising (marking conference), detailed advice from the Principal Examiner on the qualities of each mark band will be given. Examiners will then receive examples of answers in each mark band that have been awarded a mark by the Principal Examiner. Examiners should mark the examples and compare their marks with those of the Principal Examiner.

When marking, examiners can use these examples to decide whether a learner's response is of a superior, inferior or comparable standard to the example. Examiners are reminded of the need to revisit the answer as they apply the mark scheme in order to confirm that the band and the mark allocated is appropriate to the response provided.

Indicative content is also provided for banded mark schemes. Indicative content is not exhaustive, and any other valid points must be credited. In order to reach the highest bands of the mark scheme a learner need not cover all of the points mentioned in the indicative content but must meet the requirements of the highest mark band. Where a response is not creditworthy, that is contains nothing of any significance to the mark scheme, or where no response has been provided, no marks should be awarded.

<table>
<tr><th>Qu</th><th>Answer</th><th>Mark</th><th>AO1</th><th>AO2</th><th>AO3</th><th>TOT</th></tr>
<tr><td>1
(a)</td><td> 

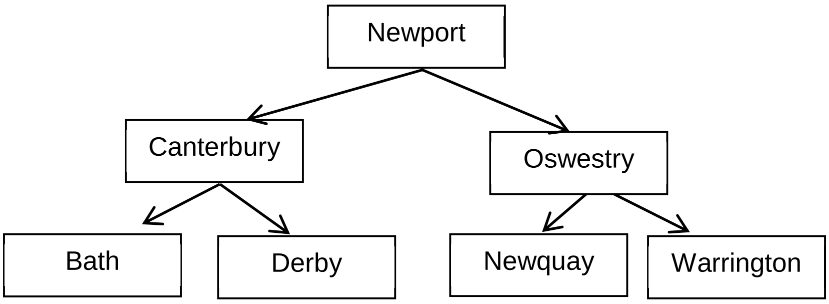
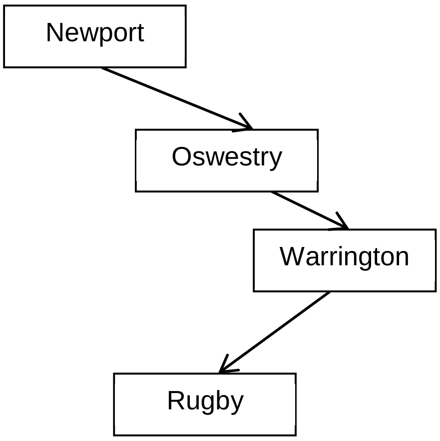
Marking
1 mark for correct root
1 mark for ALL left pointers correct and ALL right pointers correct</td><td>2</td><td></td><td>2.1b</td><td></td><td>2</td></tr>
<tr><td>1
(b)</td><td>
<table>
<tr><th></th><th>Left Pointer</th><th>Data</th><th>Right Pointer</th></tr>
<tr><td>Start 0</td><td>1</td><td>Newport</td><td>2</td></tr>
<tr><td>1</td><td>5</td><td>Canterbury</td><td>6</td></tr>
<tr><td>2</td><td>-1</td><td>Oswestry</td><td>3</td></tr>
<tr><td>3</td><td>4</td><td>Warrington</td><td>-1</td></tr>
<tr><td>4</td><td>-1</td><td>Rugby</td><td>-1</td></tr>
<tr><td>5</td><td>-1</td><td>Bath</td><td>-1</td></tr>
<tr><td>6</td><td>-1</td><td>Derby</td><td>-1</td></tr>
<tr><td>7</td><td></td><td></td><td></td></tr>
<tr><td>8</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td></td></tr>
</table>

Marking
1 mark for correct root with start pointer
1 mark for ALL left pointers correct
1 mark for ALL right pointers correct</td><td>3</td><td></td><td>2.1a</td><td></td><td>3</td></tr>
</table>

<table>
<tr><th>Qu</th><th>Answer</th><th>Mark</th><th>AO1</th><th>AO2</th><th>AO3</th><th>TOT</th></tr>
<tr><td>1
(c)(i)</td><td>
<table>
<tr><th></th><th>Left Pointer</th><th>Data</th><th>Right Pointer</th></tr>
<tr><td>Start 0</td><td>1</td><td>Newport</td><td>2</td></tr>
<tr><td>1</td><td>5</td><td>Canterbury</td><td>6</td></tr>
<tr><td>2</td><td>7</td><td>Oswestry</td><td>3</td></tr>
<tr><td>3</td><td>4</td><td>Warrington</td><td>-1</td></tr>
<tr><td>4</td><td>-1</td><td>Rugby</td><td>-1</td></tr>
<tr><td>5</td><td>-1</td><td>Bath</td><td>-1</td></tr>
<tr><td>6</td><td>-1</td><td>Derby</td><td>-1</td></tr>
<tr><td>7</td><td>-1</td><td>Newquay</td><td>-1</td></tr>
<tr><td>8</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td></td></tr>
</table>
Marking
1 mark for Newquay in correct location 7
1 mark for left pointer of Oswestry set to 7
1 mark for left and right pointers of Newquay set to -1</td><td>3</td><td></td><td>2.1b</td><td></td><td>3</td></tr>
<tr><td>1
(c)
(ii)</td><td>
<table>
<tr><th></th><th>Left Pointer</th><th>Data</th><th>Right Pointer</th></tr>
<tr><td>Start 0</td><td>1</td><td>Newport</td><td>2</td></tr>
<tr><td>1</td><td>5</td><td>Canterbury</td><td>6</td></tr>
<tr><td>2</td><td>-1</td><td>Oswestry</td><td><s>3</s> 4</td></tr>
<tr><td>3</td><td>4</td><td>Warrington</td><td>-1</td></tr>
<tr><td>4</td><td>-1</td><td>Rugby</td><td>-1</td></tr>
<tr><td>5</td><td>-1</td><td>Bath</td><td>-1</td></tr>
<tr><td>6</td><td>-1</td><td>Derby</td><td>-1</td></tr>
<tr><td>7</td><td>-1</td><td>Newquay</td><td>-1</td></tr>
<tr><td>8</td><td></td><td></td><td></td></tr>
<tr><td>9</td><td></td><td></td><td></td></tr>
</table>
Marking
1 mark for right pointer of Oswestry set to point at Rugby 4</td><td>1</td><td></td><td>2.1b</td><td></td><td>1</td></tr>
<tr><td>1
(d)</td><td>Advantage is that access to data is generally faster or data can be retrieved in different useful orders such as ‘in order’ or ‘post order’
Disadvantage is that there is the overhead of two pointers and the access algorithm is more complicated</td><td>1

1</td><td>1.1b

1.1b</td><td></td><td></td><td>2</td></tr>
</table>

Qu	Answer	Mark	AO1	AO2	AO3	TOT
1 *(e)*						4
	The above tree is balanced and the maximum number of comparisons to locate an item would be the same as the number of levels, 3 in this case					
						
	The above tree is unbalanced and the maximum number of comparisons to locate an item would be the same as the number of items, 4 in this case					
	Marking					
	1 mark for example of each tree – balanced and unbalanced	2		2.1a		
	1 mark for each description of number of comparisons	2		2.1b		

Qu	Answer	Mark	AO1	AO2	AO3	TOT
2	One mark for each valid reason for following a specific rule, up to a maximum of five marks. No marks for simply stating rules, as question requires reasons for following rules. If rule is implicit within reasoning, award mark for response (e.g. 'data must be kept safe **to prevent data from being stolen that could cause an individual embarrassment'**) – Security rule implied. Indicative content (bold indicates example valid reasoning within a summary, other valid reasoning accepted, if justified) • Programmers should have due regard for public health, privacy, security and wellbeing of others and the environment. **This will ensure that no harm physical, emotional or financial comes to an individual from not taking these factors into account (could also be reasoned with the use of an appropriate example; for instance in the case of privacy, data could be stolen that could cause an individual embarrassment)** • Programmers should have due regard for the legitimate rights of any person or organisation that might be affected by their activities. **This will ensure that the rights of others are respected and no harm comes to the public (could also be reasoned with the use of an appropriate example; for instance the right of an individual that data held is only used for an agreed purpose and not abused)** • Programmers should conduct their professional activities without discrimination on the any grounds. **This will ensure that no individual is denied their rights (could also be reasoned with the use of an appropriate example; for instance that software is carefully designed to consider other groups' needs, for example accessibility for disabled people)** • Programmers should promote equal access to the benefits of IT and seek to promote the inclusion of all sectors in society wherever opportunities arise. **This will ensure that there is no technology gap between sectors in society (could also be reasoned with the use of an appropriate example; for instance that a deprived community has opportunity access to the same website as an affluent group)** • Programmers should not claim any level of competence that they do not possess. **This safeguards an employer placing a programmer on a task that could not be completed or would be completed with significant errors which would waste time or money. (could also be reasoned with the use of an appropriate example; for instance a programmer claiming that they could use a given language but could not and then could not write the necessary program)** • Programmers should develop their professional knowledge, skills and competence on a continuing basis, maintaining awareness of technological developments, procedures, and standards that are relevant to their field. **This ensures that the product produced by a programmer is up-to-date and will function in contemporary systems (could also be reasoned with the use of an appropriate example; for instance ensuring that a programmer writes software that will function on a new operating system)** • Programmers should ensure that they have the knowledge and	5	1.1b			5

Qu	Answer	Mark	AO1	AO2	AO3	TOT
	understanding of legislation and that they comply with such legislation, in carrying out their professional responsibilities. **This ensures that the programmer does not unwittingly break the law when undertaking their day to day job which could cause embarrassment or losses (could also be reasoned with the use of an appropriate example; for instance developing insecure software that breaches data protection laws)** • Programmers should respect and value alternative viewpoints and, seek, accept and offer honest criticisms of work. **This ensures that all relevant approached and options are considered, and the best one chosen (could also be reasoned with the use of an appropriate example; for instance when developing a user interface all opinions should be considered and the best design used)** • Programmers should avoid injuring others, their property, reputation, or employment by false or malicious or negligent action or inaction. **This ensures that staff are aware that they should consider others before taking action and do not take risks that could injure others (could also be reasoned with the use of an appropriate example; for instance programmers should avoid altering a program that may lose work for others)** • Programmers should reject and not make any offer of bribery or unethical inducement. **This ensures that staff are not open to corruption from others and take actions that could harm a company or client (could also be reasoned with the use of an appropriate example; for instance programmers should not disclose sensitive data if offered an incentive to do so)** • Programmers should carry out their professional responsibilities with due care and diligence in accordance with the employer or client's requirements whilst exercising professional judgement at all times. **This would ensure that programs are developed in line with a client's requirements and that time/money is not wasted in developing other, unrequired areas (could also be reasoned with the use of an appropriate example for instance a programmer should let an employer know if a certain methodology is not working and advise on methods of changing methodology)** • Programmers should seek to avoid any situation that may give rise to a conflict of interest between them and their employer or client. **This would ensure that a programmer does not have conflicting tasks that may result in one not being completed properly (could also be reasoned with the use of an appropriate example; for instance that a programmer should not embark on a personal programming project that competes with that commissioned by a client.)** • Programmers should accept professional responsibility for their work and for the work of colleagues who are defined in a given context as working under their supervision. **This gives ownership of work, and with this, less chance of neglecting the work as the programmer is directly responsible. (could also be reasoned with the use of an appropriate example; for instance if a programmer has a set role in a task, they are likely to feel that they own that task and are more likely to do that task to the best of their ability)** • Programmers should not disclose or authorise to be disclosed, or use for personal gain or to benefit a third party, confidential information except with the permission of their employer or client, or as required by legislation. **This would undermine a client, and possibly result in**					

Qu	Answer	Mark	AO1	AO2	AO3	TOT
	loss if a competitor were to develop a product based on information disclosed. (could also be reasoned with the use of an appropriate example; for instance this prevents a programmer from selling information on a product to a company developing a similar product) • Programmers should not misrepresent or withhold information on the performance of products, systems or services (unless lawfully permitted to do so by a duty of confidentiality) or take advantage of the lack of relevant knowledge or inexperience of others. **This prevents making financial or other gain from overstating the work required for a given task. (could also be reasoned with the use of an appropriate example; for instance could mean that a programmer could charge more money by stating that a simple task took longer to complete than it actually did)** • Programmers should accept their personal duty to uphold the reputation of the profession and not take any action which could bring the profession into disrepute. **This ensures that the profession is not seen negatively by the wider public and not undermined by a lack of trust. (could also be reasoned with the use of an appropriate example – many potential examples)** • Programmers should encourage and support fellow members in their professional development. **This ensures that fellow members are able to support their team in development and that individuals are not undermined or lose out as a result of a lack of knowledge. (could also be reasoned with the use of an appropriate example – many potential examples)** • Programmers seek to improve professional standards through participation in their development, use and enforcement. **This ensures that programmers have ownership of the standards and these standards are more likely to be relevant to programmers as a result. (could also be reasoned with the use of an appropriate example, for instance if there were a new programming certification, the fact that programmers were part of its development would give the certification more status)** • Programmers notify the employer if convicted of a criminal offence. **This ensures that an employer can judge if a programmer can continue in their role as there may be risks if the crime is relevant to their work. (could also be reasoned with the use of an appropriate example – for instance if convicted of fraud, a programmer would not be permitted to program financial systems)**					

<table>
<tr><th>Qu</th><th>Answer</th><th>Mark</th><th>AO1</th><th>AO2</th><th>AO3</th><th>TOT</th></tr>
<tr><td>3</td><td>
Example lookup tables used by the compiler during translation of this language are:

One mark for all reserved words and symbols
One mark for unique hex tokens

<table>
<tr><th>Reserved word</th><th>Token (Hex)</th></tr>
<tr><td>print</td><td>2A</td></tr>
<tr><td>=</td><td>2E</td></tr>
<tr><td>+</td><td>2F</td></tr>
<tr><td>;</td><td>30</td></tr>
<tr><td>(</td><td>31</td></tr>
<tr><td>)</td><td>32</td></tr>
</table>

One mark for all user identifiers
One mark for unique hex tokens

<table>
<tr><th>Used identifier</th><th>Type</th><th>Token (Hex)</th></tr>
<tr><td>NumInput</td><td>Integer</td><td>7A</td></tr>
<tr><td>Total</td><td>Integer</td><td>7B</td></tr>
<tr><td>Count</td><td>Integer</td><td>7C</td></tr>
</table>

Example of output stream:
7B 2E 7B 2F 7A (accept answer with spaces between tokens or without spaces)
</td><td>1
1

1
1

1</td><td></td><td>2.1b
2.1b

2.1b
2.1b

2.1a</td><td></td><td>5</td></tr>
<tr><td>4
(a)</td><td>
One possible solution is:

A . B + A

$\bar{A} + \overline{B} + A$ [$\overline{A . B} = \bar{A} + B$ De Morgan]

$\overline{B} + 1$ [$\bar{A} + A \rightarrow 1$]

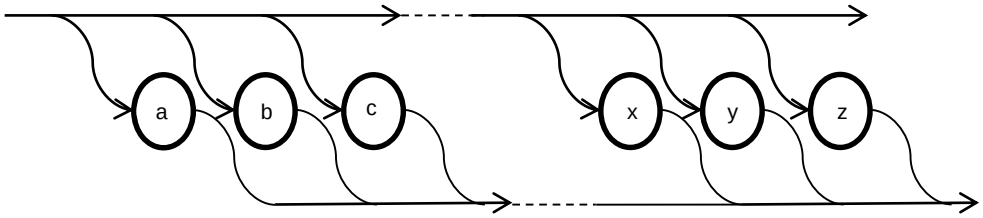
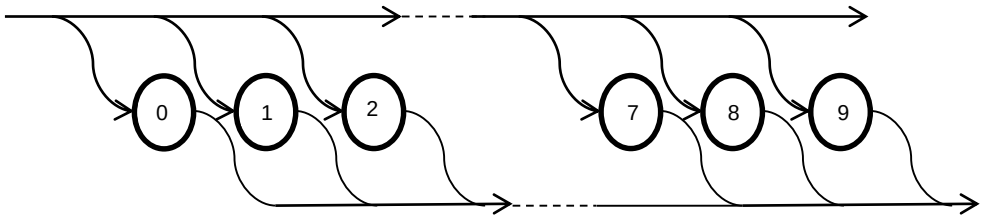
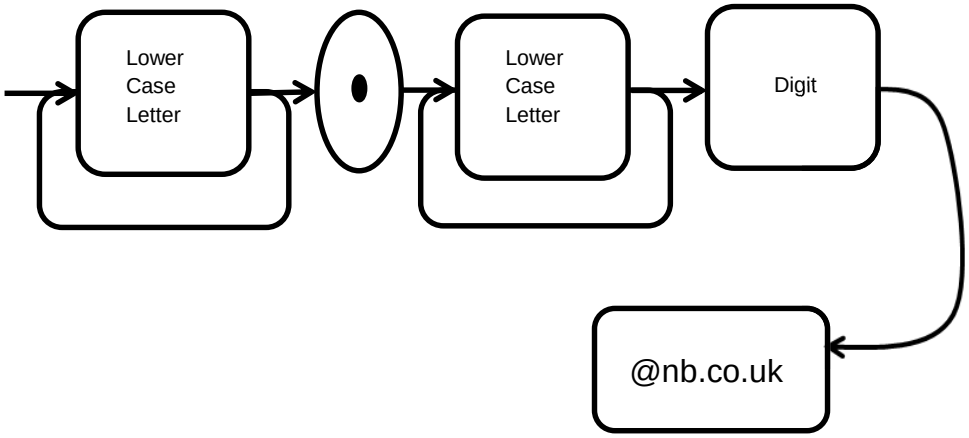
1 [$\overline{B} + 1 \rightarrow 1$]

$\overline{A . B} + A = 1$

Marking
Correctly applying De Morgan’s law – 1 mark
Correctly applying identities to arrive at correct answer – 2 marks
Correctly applying identities but arriving at incorrect answer - 1 mark
</td><td>3</td><td></td><td>2.1a</td><td></td><td>3</td></tr>
</table>

Qu	Answer	Mark	AO1	AO2	AO3	TOT
4 *(b)*	Correct answer can be established using different steps / laws / rules / identities / dual relations One possible solution is: A + B.(A + B) + A.(Ā + B) A + B.A + B.B + A.(Ā + B) [distributive law] A + B.A + B + A.(Ā + B) [B.B → B] A + B + A.(Ā + B) [B.A + B → B] A + B + A.Ā + A.B [distributive law] A + B + A + A.B [A.Ā → A] A + B + A [A.B → A] A + B + A [A.B → A] A + A + B [commutative law] A + B [A + A → A] Alternatively A + B.(A + B) + A.(Ā + B) A + B + A.(Ā + B) [B.(A + B) → B] A + B + A.B [A.(Ā + B) → A.B] A + A.B + B [commutative law] A + B [A + A.B → A] Marking Correctly applying rules to arrive at correct answer – 5 marks Correctly applying rules but arriving at incorrect answer then one mark per correct rule applied – max 4 marks	5		2.1a		5
5 *(a)*	It is good programming practice to use constants, such as VAT = 20%, because they can be set once and then used many times throughout the program and if they change then they only have to be changed once. It is good programming practice to use meaningful names for variables because it is then clear what the variable is holding and aids program readability. It is good programming practice to use annotation because other programmers will find the code easier to follow.	1 1 1	1.1b 1.1b 1.1b			3
5 *(b)*	Value = R Reference = A Both must be correct for one mark	1		2.1a		1
5 *(c)*	Value – a copy of the data is passed into the procedure and cannot be altered Which has the advantage of avoiding unwanted side effects like the value being changed by mistake Reference – a copy of the memory address is passed into the procedure therefore the data in the calling environment can be altered	1 1 1	1.1b 1.1b 1.1b			3

Qu	Answer	Mark	AO1	AO2	AO3	TOT
6	Explanation including four from the points below: A debugger is a software tool used to detect, locate and correct faults in a program and will permit an individual: • to see the progress through the program - which statements / procedures are being executed at any time using program trace (step through / step-into) • to temporarily halt execution at a fixed point in the code in order to ascertain the value of variables at that point (or to step through the program from that point) using a break point • to temporarily to halt execution when a condition is satisfied in order to ascertain the value of variables at that point using a conditional break point • to view the value of a variable at specific points during the execution using variable watch • to view the entire contents of memory at a specific point using store dump • to view error messages relating to errors in the program using error diagnostics • to see the values of variables at the point where the program failed using post-mortem routines. Any other facilities that assist debugging if explained accurately may be worthy of credit.	4	1.1b			4

Qu	Answer	Mark	AO1	AO2	AO3	TOT
7 *(a)*	Lower case Letter  Digit  Email address  Marking of email address One mark for Lower Case Letter repeated twice Two marks for correct sequence of Lower Case Letter, dot, Lower case Letter and digit One mark if mistake in correct sequence (for example dot missing or in incorrect sequence)	1 1 3		2.1b 2.1b 2.1b		5
7 *(b)*	<letter> ::= a|b|c . . . |z <digit> ::= 0|1|2 . . . |9 <name> ::= <letter>|<letter><name> <email_ad> ::= <name>.<name><digit>@nb.co.uk Marking One mark for recursion: Same item Left and Right are needed Cannot gain full marks unless completely correct	1 1 1 1		2.1b 2.1b 2.1b 2.1b		4

Qu	Answer	Mark	AO1	AO2	AO3	TOT
7 *(c)*	The syntax of computer languages follow strict unambiguous rules but the English language can be ambiguous One mark for both comparison to syntax diagrams and BNF Syntax diagrams provide a diagrammatic representation which are easy to follow BNF definitions can easily be coded when writing a compiler	1 1 1	1.1b 1.1b 1.1b			3
8	Perfective is when the system functionality is improved For example reducing the search time for stock control records Adaptive is when the system is changed to run on a new hardware or software platform For example the retail company could switch from a Windows based system to Macintosh or Linux Corrective is fixing bugs in the system that were not detected during development or system testing For example the users might find that some data input for new stock is not validated and causes program to crash or produce inaccurate output	1 1 1 1 1 1	1.1a 1.1a 1.1a 	 2.1a 2.1a 2.1a		6
9 *(a)*	A recursive algorithm is one: That calls itself And has a terminating (base) condition to stop the calling	 1 1	 1.1a 1.1a			2
9 *(b)*	Could run out of memory (stack space) due to high number of recursive calls Difficult to debug (dry run) if producing incorrect results due to difficulty in determining which recursive call produced the error	1 1 1 1	1.1a 1.1b 1.1a 1.1b			4
10	A class is a template defining methods and attributes used to make objects Inheritance is when a new class is created that is based on another class and It can take on the methods and attributes of the parent class	1 1 1	1.1a 1.1a 1.1a			3
11 *(a)*	Input 10 integers into an array Correct control structure – repetition using a loop Identifying (selection using IF) and outputting largest value	1 1 1			3.1b 3.1b 3.1b	3
11 *(b)*	Sort the array of 10 integers into ascending order Output the 10 integers in ascending order Marking of the sort Correct control structures (nested loops) – 2 marks Comparing values for ascending sort - 1 mark Successful swapping of elements – 1 mark Algorithm terminates correctly – 1 mark Algorithm deals with identical values – 1 mark	6			3.1b	6

Qu	Answer	Mark	AO1	AO2	AO3	TOT
12 *(a)*	One mark for both correct $10^3 = 1000$ $100^3 = 1\ 000\ 000$	1		2.1a		1
12 *(b)*	Evaluation of algorithm Multiplication The only multiplication appears in the k loop. Since this loop will iterate a total of n^3 times, it will execute exactly n^3 multiplication operations Addition There are two addition operations in the k loop: (I) Answer(i, j) = Answer(i, j) + A(i, k) * B(k, j) (II) Incrementing k Each of these will happen n^3 times. Therefore there are $2n^3$ additions generated by the k loop The j loop has one addition operation Incrementing j which happens n^2 times The i loop has one addition operation as well Incrementing i which happens n times. Adding these up we the number of addition operations which is $2n^3 + n^2 + n$ As n gets very big then n^3 will dominate therefore it is $O(n^3)$ NOTE: Calculations might include assignment operations but these will not affect overall time so ignore **Marking** One mark for identifying i loop will execute n times One mark for identifying j loop will execute n^2 times One mark for identifying that k loop will execute n^3 times. One mark for correct number of additions $2n^3 + n^2 + n$ One mark for determining that the order will be dominated by n^3 as n gets very big giving $O(n^3)$ for the algorithm	5			3.1c	5
12 *(c)*	Algorithm will need to store 3 matrices A, B and Answer Each matrix will require N x N items = N^2 Total storage will be 3 x N x N = $3N^2$ As n increases the storage requirements will increase n^2 as constant (3) will be insignificant so storage requirements will be Order (n^2) **Marking** One mark for identifying storage requirements as $3N^2$ One mark for determining the storage requirements will be $O(n^2)$	2			3.1c	2

Qu	Answer	Mark	AO1	AO2	AO3	TOT
13	**Indicative content** When developing a system in modules then it is desirable for them to be highly cohesive and loosely coupled **Algorithms and programs** Use of constants – allow modules to be easily updated, for example when the rate of VAT changes a module that calculates Gross Price can be updated by only changing the constant value for the rate of VAT and likewise fir income tax, NI contributions, pensions, etc… Use of standard functions that have already been written can easily be integrated into a modular system Use of meaningful identifiers are essential for team members to be able to understand and amend modules written by other programmers Use of local variables that are not used outside the module will help ensure the system is highly cohesive and loosely coupled Use of global variables should be avoided as will make the system 'tightly coupled' Use of parameters will help ensure the system is highly cohesive and loosely coupled All variables and parameters should be clearly documented for future and other programmers Modules can be tested individually before integration Ensure there is data independence so underlying data structures can be altered without affecting existing modules **Principles of programming** Many different computer languages facilitate modular programming A class in OO languages can be considered a module Modular programming is suitable for bottom up approach to system development Modular programming is suitable for a structured top down approach to system development (stepwise refinement) Modules must be written to agreed standards to allow easy integration Modules must be written to agreed standards to allow system to be maintained	12	1.1b			12

	Systems analysis Modular programming is suitable for both agile and waterfall approach to system development Modules can be integrated into system one module at a time Modules can easily be replaced with improved modules Modules can easily be replaced with modules with additional or different functionality Existing modules can be re-used Modules have been tried and tested so less chance of future errors Different programmers can work on different modules at the same time so the critical time path is shorter Programmers with expertise in specific areas can write modules in their specialised area **Software engineering** Integrated Development Environment can be used to debug modules which are easier to debug as smaller more manageable amount of code Integrated Development Environment can be used to create a library of modules Program version management software can be used when team are developing modules to ensure only latest version of module is used **Program construction** No need to re-compile modules Modules need to be ‘re-locatable’ in RAM Dynamic link loader loads therefore saving time					

Band	AO1.1b Max 12 marks
3	**9-12 marks** The candidate has: • written an extended response that has a sustained line of reasoning which is coherent, relevant, and logically structured • shown clear understanding of the requirements of the question and a clear knowledge of the topics as specified in the indicative content. Clear knowledge is defined as responses that provide relevant detailed points of the implications of writing programs in modules, which relate to an extensive amount of the indicative content. • addressed the question appropriately with minimal repetition and no irrelevant material • has presented a balanced discussion and justified their answer with examples • effectively drawn together different areas of knowledge, skills and understanding from all relevant areas across the course of study • used appropriate technical terminology confidently and accurately.
2	**5-8 marks** The candidate has: • written a response that has an adequate line of reasoning with elements of coherence, relevance, and logical structure • shown adequate understanding of the requirements of the question and a satisfactory knowledge of the topics as specified in the indicative content. Satisfactory knowledge is defined as responses that provide relevant points of the implications of writing programs in modules, which relate to the indicative content. • presented a discussion with limited examples • drawn together different areas of knowledge, skills and understanding from a number of areas across the course of study • used appropriate technical terminology.
1	**1-4 marks** The candidate has: • written a response that that lacks sufficient reasoning and structure • produced a discussion which is not well developed • attempted to address the question but has demonstrated superficial knowledge of the topics specified in the indicative content. Superficial knowledge is defined as responses that provide limited relevant points of the implications of writing programs in modules which relate to a limited amount the indicative content. • used limited technical terminology.
0	Response not credit worthy or not attempted.

Component 2 – Computer Architecture, Data, Communication and Applications

Mark Scheme

Guidance for examiners

Positive marking

It should be remembered that learners are writing under examination conditions and credit should be given for what the learner writes, rather than adopting the approach of penalising him/her for any omissions. It should be possible for a very good response to achieve full marks and a very poor one to achieve zero marks. Marks should not be deducted for a less than perfect answer if it satisfies the criteria of the mark scheme.

For questions that are objective or points-based the mark scheme should be applied precisely. Marks should be awarded as indicated and no further subdivision made.

For band marked questions mark schemes are in two parts.

Part 1 is advice on the indicative content that suggests the range of computer science concepts, theory, issues and arguments which may be included in the learner's answers. These can be used to assess the quality of the learner's response.

Part 2 is an assessment grid advising bands and associated marks that should be given to responses which demonstrate the qualities needed in AO1, AO2 and AO3. Where a response is not credit worthy or not attempted it is indicated on the grid as mark band zero.

Banded mark schemes

Banded mark schemes are divided so that each band has a relevant descriptor. The descriptor for the band provides a description of the performance level for that band. Each band contains marks.

Examiners should first read and annotate a learner's answer to pick out the evidence that is being assessed in that question. Once the annotation is complete, the mark scheme can be applied.

This is done as a two stage process.

Stage 1 – Deciding on the band

When deciding on a band, the answer should be viewed holistically. Beginning at the lowest band, examiners should look at the learner's answer and check whether it matches the descriptor for that band. Examiners should look at the descriptor for that band and see if it matches the qualities shown in the learner's answer. If the descriptor at the lowest band is satisfied, examiners should move up to the next band and repeat this process for each band until the descriptor matches the answer.

If an answer covers different aspects of different bands within the mark scheme, a 'best fit' approach should be adopted to decide on the band and then the learner's response should be used to decide on the mark within the band. For instance if a response is mainly in band 2 but with a limited amount of band 3 content, the answer would be placed in band 2, but the mark awarded would be close to the top of band 2 as a result of the band 3 content. Examiners should not seek to mark candidates down as a result of small omissions in minor areas of an answer.

Stage 2 – Deciding on the mark

Once the band has been decided, examiners can then assign a mark. During standardising (marking conference), detailed advice from the Principal Examiner on the qualities of each mark band will be given. Examiners will then receive examples of answers in each mark band that have been awarded a mark by the Principal Examiner. Examiners should mark the examples and compare their marks with those of the Principal Examiner.

When marking, examiners can use these examples to decide whether a learner's response is of a superior, inferior or comparable standard to the example. Examiners are reminded of the need to revisit the answer as they apply the mark scheme in order to confirm that the band and the mark allocated is appropriate to the response provided.

Indicative content is also provided for banded mark schemes. Indicative content is not exhaustive, and any other valid points must be credited. In order to reach the highest bands of the mark scheme a learner need not cover all of the points mentioned in the indicative content but must meet the requirements of the highest mark band. Where a response is not creditworthy, that is contains nothing of any significance to the mark scheme, or where no response has been provided, no marks should be awarded.

Qu	Answer	Mark	AO1	AO2	AO3	TOT
1 *(a)*	-9 (Correct answer only)	1		2.1a		1
1 *(b)*	One method is: • From RHS, rewrite it up to and including the first one • Change other 1 digits to 0 and 0 digits to 1 • Correct working and answer for example 00001000 -> xxxx1000 -> 11111000 Alternatively • Flip the bits • Add one • (Ignore carry (ninth bit)) • Correct working and answer for example (as above) (Other methods equally acceptable)	1 1	1.1b	2.1a		2
1 *(c)*	MANTISSA EXPONENT 0101101100000000 0101 (spacing unimportant) 22 -> 10110 .75 -> .11 Marking: for correct mantissa for correct exponent	 1 1		 2.1a 2.1a		2
1 *(d)*	A greater range of (positive / negative) numbers can be stored in the same number of bits using FP form However, numbers in FP form (unlike integer form); any 1 of: • are not normally stored completely accurately e.g. 7 might effectively be stored as 7.000000000001 • require more complex processing • allow for no exact representation of zero	1 1	1.1b 1.1b			2
2 *(a)*	The address of the next instruction is copied from the PC to the MAR The instruction is copied to the MDR The PC is incremented so that it holds the address of the next instruction	1 1 1	1.1b 1.1b 1.1b			3
2 *(b)*	d1 1001 0110 1110 1100 (original value) d2 1110 1011 1010 0011 (original value) Register 1 1001 0110 1110 1100 (after instruction LOAD 1,d1) Register 2 1110 1011 1010 0011 (after instruction LOAD 2,d2) Register 1 0111 1101 0100 1111 (after instruction XORR 1,2) d3 0111 1101 0100 1111 (after instruction STORR 1,d3) Register 1 1001 0110 1110 1100 (after instruction XORR 1,2)	 1 1 1 1		 2.1a 2.1a 2.1a 2.1a		4

Qu	Answer	Mark	AO1	AO2	AO3	TOT
3 *(a)*	A foreign key is a field in a table which links to a primary key (or part of a primary key) in another table It enables data in different tables to be linked together	1 1	1.1b 1.1b			2
3 *(b)*	Remove any transitive dependencies (or Ensure that each attribute/field depends only the primary key)	1	1.1a			1
3 *(c)* (i)	SELECT StuName, TutNum FROM STUDENTS	1			3.1b	1
3 *(c)* (ii)	SELECT StuName FROM STUDENTS WHERE TutNum = '378'	1			3.1b	1
3 *(c)* (iii)	UPDATE STUDENTS SET TutNum = '345' WHERE TutNum = '378' 1 mark for update; 1 mark for changing TutNum	2			3.1b	2
3 *(c)* (iv)	SELECT StuName FROM STUDENTS WHERE TutNum = (SELECT TutNum FROM STUDENTS WHERE StuNum = '13777') 1 mark for each select (1 mark for main select and one from sub-select)	2			3.1b	2
3 *(c)* (v)	SELECT StuName, TutName FROM STUDENTS, TUTORS WHERE STUDENTS.TutNum = TUTORS.TutNum 1 mark for each select (1 mark for main select and 1 mark from sub-select)	2			3.1b	2
3 *(c)* (vi)	CREATE TABLE PHONES (RoomNum Char(5) PhoneNum Char(5)) 1 mark for table; 1 mark for fields (any suitable field size acceptable)	2			3.1b	2
3 *(c)* (vii)	INSERT INTO PHONES VALUES ('106' , '625') INSERT INTO PHONES VALUES ('113' , '670') 1 mark for insert; 1 mark for values inserted	2			3.1b	2
3 *(c)* (viii)	SELECT StuName, DateBirth FROM STUDENTS ORDER BY DateBirth 1 mark for select; 1 mark for ordering	2			3.1b	2

Qu	Answer	Mark	AO1	AO2	AO3	TOT

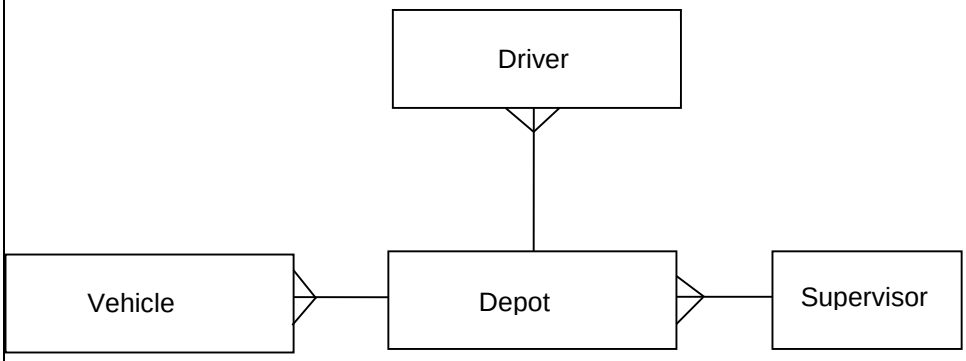
4 *(a)*	<u>Sequential</u>
	
	• Records are stored and accessed in key sequence order
	• Easier to program / fewer overheads than indexed sequential
	• Particularly suitable (and faster) if access only ever needs to be sequential
	
	Addition of a record:
	• Make a new copy of the records until in the correct place to add the new record
	• Add the new record to the new copy
	• Continue until the end of the file
	• If multiple records to be added, these should preferably be sorted before the above process to avoid multiple updates
	
	Deletion of a record:
	• Make a new copy of the records until in the correct place for deletion
	• Do not copy the record to be deleted
	• Continue until the end of the file
	• If multiple records to be deleted, these should preferably be sorted before the above process to avoid multiple updates
	• Allows faster access than sequential because can move directly to individual records
	
	<u>Indexed Sequential</u>
	• Records are stored in key order in the file
	• Records are normally organised into blocks
	• An index allows data to be accessed directly
	• Multilevel index usually used:
	  • There is a main index which contains the location of the next index
	  • This process may extend to several levels and the last index contains the physical address of the record
	• Blocks are originally normally partially filled to allow for more entries
	
	Addition of a record:
	• Place in a block if possible
	• If a block becomes full, an overflow area is used
	• Access may become slow as more records are in overflow area, so re-organisation may become necessary
	
	Deletion of a record
	• Record is normally marked as deleted but not physically removed
	
	[Sample response worth full marks:]
	
	In a sequential file, records are stored and accessed in key sequence order. When a record is to be added, a copy is made of the records until it is in the correct place to add the new record, then the new record is added to the copy. The process is continued until the next new record is detected or the end of the file is reached. If multiple records are to be added, these should preferably be sorted before the above process to
		5	1.1b			5

Qu	Answer	Mark	AO1	AO2	AO3	TOT
	avoid multiple updates. When a record is to be deleted, a copy is made until in the correct place for deletion. The record to be deleted is not copied to the new copy. The process continues to the end of the file as above In an indexed sequential file, records are stored in key order in the file. Records are normally organised into blocks and an index (probably multilevel) allows data to be accessed directly. This indexing process may extend to several levels and the last index contains the physical address of the record. Blocks are originally normally partially filled to allow for more entries. When a record is to be added, it is placed in a block if possible, but an overflow area is used if necessary. Access may become slow as more records are located in the overflow area, so re-organisation may become necessary. When a record is to be deleted, it is normally marked as deleted but not physically removed. The advantages of a sequential file are that it is generally easier to program than the indexed sequential. It is particularly suitable (and faster) if access only ever needs to be sequential. The advantages of an indexed sequential file are that it allows faster access than sequential because it can move directly to individual records. Marking 5 valid points comparing access methods and addition and deletion of records for 5 marks If addition and deletion of records not discussed, max 4 marks Must compare both access types for 3 marks or more If only one access type discussed, max 2 marks					
4 *(b)* (i)	123^2 = 015129, so the output is 51 223^2 = 049729, so the output is 97	2		2.1a		2
4 *(b)* (ii)	123^2 = 015129, so the output is 29 223^2 = 049729, so the output is 29	2		2.1a		2
4 *(b)* (iii)	The second algorithm is weaker As it will always give the same output if the two right hand digits in the key are the same. The first algorithm does not do this so is preferable (further example 323^2 -> 10<u>43</u>29 for first algorithm and 1043<u>29</u> for second)	1 1		2.1b 2.1b		2
5	……. set a = b * c set d = b * e set f = b * g ……. 1 mark for first two statements which can be run in parallel, with second statement excluding first set value (a in this instance) 1 mark for third statement excluding first two set values (a and d in this instance)	2			3.1b	2

Qu	Answer	Mark	AO1	AO2	AO3	TOT
6 *(a)*	Simplex: Data transmission is possible in one direction only	1	1.1a			3
	Half duplex: Data transmission is possible in both directions, but only in one direction at a time	1	1.1a			
	Full duplex: Data transmission is possible in both directions simultaneously	1	1.1a			
6 *(b)*	Data collision occurs when two sets of data are transmitted on the network simultaneously	1	1.1b			3
	The network detects the error	1	1.1b			
	Each computer waits for a short / random time then sends the data again	1	1.1b			
7 *(a)* (i)	01111000	1		2.1a		2
	Effect is multiplying by 4 (or doubling twice, or multiply by 100_2 – subscript needed)	1		2.1b		
7 *(a)* (ii)	Number becomes (~~1~~) 00111100 - left hand digit lost as the resulting number is now too large to be contained in the eight bits available	1		2.1b		2
	Any 1 of: • More digits are required • An error should be reported rather than producing an inaccuracy	1		2.1b		
7 *(b)*	Hexadecimal is easier (for humans) to read / copy etc.	1	1.1b			2
	Example eg 2AD -> 0010 1010 1101 or 001010101101	1		2.1b		

Qu	Answer	Mark	AO1	AO2	AO3	TOT
8 *(a)*	The estate agent will hold a huge amount of data altogether. It may be more efficient to store data on a number of different computers to maximise performance. (For instance info on properties in each town could be stored on the computer in that town with links to other branches)	1		2.1a		3
	It will be difficult for the estate agent to ensure that all the data in all the computers is always up-to-date / maintain integrity	1		2.1b		
	Both <u>processing</u> and <u>data</u> are distributed	1	1.1a			
8 *(b)*	Big Data refers to data sets so large and complex that it becomes difficult to process using standard database techniques	1	1.1b			1
8 *(c)*	Data mining: the analysis of (a large amount of) data (in a data warehouse)	1	1.1b			4
	This can provide <u>new information / find patterns / trends</u> in the data	1	1.1b			
	Supermarket: could use data mining on data extracted from loyalty card data to attract customer to make additional purchases	1		2.1a		
	via targeted special offers, etc. placement of products to attract customers etc.	1		2.1a		
9 *(a)*	Real-time control system – a control system which reacts quickly enough to affect behaviour outside the system	1	1.1b			6
	Example: a chemical plant where the computer can react to a (possibly dangerous) temperature rise by employing cooling measures very rapidly	2		2.1a		
	Real-time transaction processing system - a system which quickly locks records or fields to avoid accidental update and possible loss of integrity	1	1.1b			
	Example: an online theatre booking system where records / fields are locked during a transaction to avoid the possibility of double booking	2		2.1a		

Qu	Answer	Mark	AO1	AO2	AO3	TOT
9 *(b)*	More than one job is in memory at same time A time-slice is the amount of time allocated to each job by the operating system Scheduling allocates time-slices to the jobs Polling is the sequential checking of jobs so that each gets its appropriate share of time Partitioning is a division of a computer's memory for different jobs One job is halted if e.g. waiting for a peripheral device – other jobs can now be processed - this is achieved by use of interrupts	4	1.1b			4
9 *(c)*	The operating system would have to halt the execution of the low priority task currently running The operating system (scheduler) would then have to service the required high priority Interrupt Service Routine (ISR) The new medium priority Interrupt Service Routine would be held until the high priority has been dealt with and then the medium priority interrupt would be serviced When both interrupts are serviced then executing the original low priority task is resumed	1 1 1 1		2.1b		4

Qu	Answer	Mark	AO1	AO2	AO3	TOT
10 *(a)*	 Marking: Diagram 3 marks - each error minus 1 mark	3		2.1b		3
10 *(b)*	DRIVER (DriverID, DriverName, DriverHomeAddress, DepotTown) VEHICLE (RegNumber, Make, Model, DepotTown) DEPOT (DepotTown, Address, TelNumber, SupervisorID) SUPERVISOR (SupervisorID, SupervisorName, SupervisorHomeAddress) Marking: Four suitably named tables 1 mark Each table with suitable primary key shown 2 marks (3 correct = 1 mark) Correct foreign keys 3 marks	6		2.1b		6

Qu	Answer	Mark	AO1	AO2	AO3	TOT
11	**Indicative content:** General issues: • perform the required operations consistently / reliably / accurately • operates in real time so has feedback / can respond to (possibly dangerous) conditions very quickly • human workers will be less subject to dangerous working conditions • system can operate 24 hours / 365 days per year / doesn't get tired etc • system may be very energy / resource efficient • system may effectively monitor operations / provide useful management data • cheap to run once installed (reduced salary costs) • there will be a need for some highly qualified maintenance staff, but generally staff numbers will be cut • system may not react to obvious fault conditions / cannot adapt to changes. Security issues: • security will need to be incorporated from initial analysis and design of system and constantly monitored / updated • techniques such as penetration testing should be employed – deliberately attempting to identify security vulnerabilities • white hat (ethical) hackers employed to carry out penetration testing and other testing techniques • danger of viruses / worms being deliberately introduced by USB sticks, remote access to system • need to consider loss of production / damage to equipment • need to consider injury / death of employees / persons in the community • need to consider company's reputation in the community • motivation of malicious behaviour could be standard criminality such as blackmail or could be terrorism • staff vetted to attempt to prevent internal malicious damage • security staff / physical locks / biometric methods to prevent unauthorised physical access.	13	1.1b			13

Band	AO1.1b Max 13 marks
3	**10-13 marks** The candidate has: • written an extended response that has a sustained line of reasoning which is coherent, relevant, and logically structured • shown clear understanding of the requirements of the question and a clear knowledge of the topics as specified in the indicative content. Clear knowledge is defined as a response that makes six or seven points in both areas signalled in the indicative content. The top of the mark range would require a clear response in both areas. • addressed the question appropriately with minimal repetition and no irrelevant material • has presented a balanced argument and justified their arguments • effectively drawn together different areas of knowledge, skills and understanding from all relevant areas across the course of study • used appropriate technical terminology referring to the indicative content confidently and accurately.
2	**5-9 marks** The candidate has: • written a response that has an adequate line of reasoning with elements of coherence, relevance, and logical structure • shown adequate understanding of the requirements of the question and a satisfactory knowledge of the topics as specified in the indicative content. Satisfactory knowledge is defined as a response that makes four or five points in both areas signalled in the indicative content. The top of the mark range would require a satisfactory response in both areas • presented an argument with limited justification • drawn together different areas of knowledge, skills and understanding from at least two areas across the course of study • used appropriate technical terminology referring to the indicative content.
1	**1-4 marks** The candidate has: • written a response that that lacks sufficient reasoning and structure • produced a discussion which is not well developed, and the justification is weak • attempted to address the question but has demonstrated superficial knowledge of the topics specified in the indicative content. Superficial knowledge is defined as a response that makes two points in both areas as signalled in the indicative content. The top of the mark range would require a superficial response in both areas. • used limited technical terminology referring to the indicative content.
0	**0 marks** Response not credit worthy or not attempted.