Computer Science

AS | AQA | 7516

AS

AQA

Revision Guide

for AS AQA Computer Science

Paper 2

zigzageducation.co.uk

POD 6073

Contents

Thank You for Choosing ZigZag Education ... ii
Teacher Feedback Opportunity ... iii
Terms and Conditions of Use ... iv
Teacher's Introduction ... 1
Revision Progress Tracker: AS Paper 2 ... 2
Topic 5 – Data representation ... 5
5.1 Number systems ... 5
5.2 Number bases ... 5
5.3 Units of information ... 8
5.4 Binary number systems ... 9
5.5 Information coding systems ... 12
5.6 Representing images, sound and other data ... 14
Topic 6 – Computer systems ... 19
6.1 Hardware and software ... 19
6.2 Classification of programming languages ... 20
6.3 Types of program translator ... 21
6.4 Logic gates ... 23
6.5 Boolean algebra ... 25
Topic 7 – Computer organisation and architecture ... 26
7.1 Internal hardware components of a computer ... 26
7.2 The stored program concept ... 28
7.3 Structure and role of the processor and its components ... 28
7.4 External hardware devices ... 34
Topic 8 – Consequences of uses of computing ... 37
8.1 Individual, social, legal and cultural issues and opportunities ... 37
Topic 9 – Communication and networking ... 41
9.1 Communication ... 41
9.2 Networking ... 42
Answers ... 45
Topic 5 – Data representation ... 45
Topic 6 – Computer systems ... 47
Topic 7 – Computer organisation and architecture ... 49
Topic 8 – Consequences of uses of computing ... 50
Topic 9 –Communication and networking ... 51

TEACHER'S INTRODUCTION

This revision guide has been written to support the AQA AS Computer Science specification (first teaching from September 2015, first exams in June 2016).

It summarises the essential theory required for the AS Paper 2 examination; more specifically, topics 5–9 of the AS specification:

5. Data representation
6. Computer systems
7. Computer organisation and architecture
8. Consequences of uses of computing
9. Communication and networking

An equivalent resource is also available for the AS AQA Paper 1 examination (topics 1–4).

Each section includes student notes, examples, diagrams and examination-style questions. Example answers to all of these questions can be found at the back of the resource. *Note that credit should also be given for any valid responses that are not explicitly included in this resource.* There is also a revision progress grid which students may find useful in the lead up to their exams.

P Chapman, January 2016

REVISION PROGRESS TRACKER: AS

Use the grid below to track your progress while revising for your exam. Start by ente the top, and working down the grid, give a rating of between 1 (you really don't know i

This should help you to focus your revision on the areas that require it the most, so tha comes up in the exam. Use the Notes column to record any actions.

Repeat this process until you feel you are confident enough in all areas and are ready f

Specification Topic	Confidence Level (1-5)				
	Date:	Date:	Date:	Date:	
5 – Fundamentals of data representation					
Number Systems					
Number Bases					
Decimal → binary					
Binary → decimal					
Decimal → hex					
Hex → decimal					
Binary → hex					
Bits and bytes					
Units of bytes					
Unsigned binary					
Unsigned binary arithmetic					
Signed binary using 2's complement					
Fixed point binary					
ASCII & Unicode					
Error checking and correction					
Analogue ⇆ digital					
Bitmapped graphics					
Digital sound					

Specification Topic	Confidence Level (1-5)				
	Date:	Date:	Date:	Date:	
Musical Instrument Digital Interface					
Data compression					
Encryption					
Caesar cipher					
Vernam cipher					
6 – Fundamentals of computer systems					
Relationship between hardware and software					
Classification of software					
System software					
Role of an operating system					
Classification of programming languages					
Types of translator					
Logic gates					
Logic diagrams					
Boolean algebra					
7 – Fundamentals of computer organisation and architecture					
Computer system components					
Von Neumann and Harvard architectures					
Stored program concept					
Parts of the CPU					
Fetch-Execute cycle					
Processor instruction set					
Addressing modes					

Specification Topic	Confidence Level (1-5)				
	Date:	Date:	Date:	Date:	
Machine code and assembly language operations					
Factors affecting processor performance					
External hardware devices					
Secondary storage devices					
8 – Consequences of uses of computing					
Awareness of current individual, social, legal and cultural risks					
Awareness of how digital technology can be used					
Responsibilities of computer scientists and engineers					
Challenges facing legislators in the digital age					
9 – Fundamentals of communication and networking					
Serial vs. parallel transmission					
Synchronous vs. asynchronous transmission					
Communication basic definitions					
Network topologies					
Peer-to-peer vs. client server					
Wireless networking					

TOPIC 5 – DATA REPRESENTAT

5.1 NUMBER SYSTEMS

Number System	Examples	Expl
Natural Numbers	N = {0,1,2,3,4, ... }	N is the set of natural numbers (ofter **counting**, e.g. 7 cars, 2 apples.
Integer Numbers	Z = {... -2, -1, 0, 1, 2 ... }	Z is the set of integer numbers that ir negative inverses, as shown in the ex
Rational Numbers	Q includes: 7/2, 8/1, 100/3, 1/2, 16/4 ...	Q is the set of numbers that can be wr integers). Integers such as 7 are ration
Irrational numbers	$\sqrt{2}$, $\sqrt{3}$, π, $\sqrt{99}$	An irrational number cannot be writter all square roots are irrational; for exam
Real Numbers	1, 12.41, $\sqrt{2}$, 0, 12/7, 1/8, 1.23423, 129.6	Real numbers include: whole number numbers; they can be positive, negati Real numbers are used for **measurem** digits to the right of the decimal poir measurement of 1.05 is more accurat
Ordinal Numbers	1st 2nd 3rd S = a, b, c	Ordinal numbers are used to describe list. In the example **S** is an ordered lis second object and **c** is the third or fin

5.1 – Progress Check

1. Explain the difference between the natural number system (N) and the int

5.2 NUMBER BASES

Number base	Description
Decimal (base 10)	The decimal number system is based on the use of 10 digits: 10^2 or 100 / 3 ; 10^1 or 10 / 5 ; 10^0 or 1 / 8 — The numb and 1s as The subscript 10 can be used to indicate the number base us
Binary (base 2)	The binary number system is based on the use of 2 digits: 0 a 2^3 or 8 / 1 ; 2^2 or 4 / 0 ; 2^1 or 2 / 1 ; 2^0 or 1 / 1 — The numb and 1s as So the binary number 1011 represents (1 x 8) + (0 x 4) + (1 x The subscript 2 can be used to indicate the number base use
Hexadecimal (base 16)	The hexadecimal number system is based on the use of 16 di where A=10, B=11, C=12, D=13, E=14 and F=15 16^2 or 256 / 2 ; 16^1 or 16 / E ; 16^0 or 1 / 1 — The numb and 1s as So the hexadecimal number 2E1 represents (2 x 256) + (14 x The subscript 16 can be used to indicate the number base us

Convert binary to decimal

To convert **binary 1011 0101 to decimal**, write the binary numbers into a table in the cor

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

Then add the decimal numbers where 1 is shown, so $1011\ 0101_2$ = 128 + 32 + 16 + 4 + 1 =

Convert decimal to binary

To convert **decimal 133 to binary**, create a table with the binary place values and use the

1. Find the largest place value that is less than or equal to 133.
   In this case, it is 128, so write 1 in row below 128.
2. Subtract 128 from 133 to obtain 5. Find the largest place value that is less than or
   In this case, it is 4, so write 1 in row below 4.
3. Subtract 4 from 5 to obtain 1. Find the largest place value that is less than or equa
   In this case, it is 1, so write 1 in row below 1.
4. Complete the process by adding 0 to all the blanks in the table.

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

Hexadecimal and binary

The **hexadecimal** number system (or **hex**) is based on 16 states. The table on the right ca to convert between binary and hex.

Note that two-digit hexadecimal numbers are the equivalent of eight binary bits or one b of data.

Convert binary to hexadecimal

Example:
To convert binary 11111011 into hexadecimal use the steps below:

1. Treat the 8-bit binary code as two hexadecimal nibbles: 1111 and 1011.
2. Use the table to convert 1111 to F and 1011 to B.
   So, $\mathbf{11111011_2 = FB_{16}}$

Convert hexadecimal to binary

Example:
To convert hexadecimal A9 into binary use the steps below:

1. Convert the two hexadecimal nibbles separately using the table,
   so $A_{16} = 1010_2$ and $9_{16} = 1001_2$.
2. Join them together to form an 8-bit binary number.
   So, $\mathbf{A9_{16} = 10101001_2}$

Convert decimal to hexadecimal

Convert decimal to binary and then convert binary to hex.
Example – convert 189_{10} to hexadecimal

1. 189 decimal is converted to binary; therefore, $10111101_2 = 189_{10}$

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

2. Split the binary number into (4-bit) nibbles

8	4	2	1
1	0	1	1

8	4	2	1
1	1	0	1

3. Add up each nibble (value between 0 and 15) and convert into hex

Giving $1011_2 = 11_{10} = B_{16}$ and $1101_2 = 13_{10} = D_{16}$

Therefore, $189_{10} = 10111101_2 = BD_{16}$

Convert hexadecimal to decimal

Convert hexadecimal to binary and then convert binary to decimal.
Example – convert $5ED_{16}$ to decimal

1. $5ED_{16}$ is converted to binary so $010111101101_2 = 5ED_{16}$

5				E				D			
8	4	2	1	8	4	2	1	8	4	2	1
0	1	0	1	1	1	1	0	1	1	0	1

2. Convert the binary into decimal using the table below:

1024	512	256	128	64	32	16	8	4	2	1
1	0	1	1	1	1	0	1	1	0	1

3. 1024 + 256 + 128 + 64 + 32 + 8 + 4 + 1 = **1517_{10}**

5.2 – Progress Check

2. Convert binary 1110 0111 to decimal (2 marks)
3. Convert decimal 101 to binary (2 marks)
4. Convert hexadecimal 3FA to decimal (2 marks)
5. Convert decimal 7012 to hexadecimal (2 marks)
6. Convert binary 0011 1111 to hexadecimal (2 marks)
7. Convert hexadecimal B7 to binary (2 marks)

5.3 UNITS OF INFORMATION

Bits and bytes

Computer hardware is used to store and process data; hardware components use the binai either 0 or 1.

Computer systems store data and program instructions using binary code, since digital coi Therefore, all data that is input into the computer system needs to be converted into bina

Single binary numbers or digits are usually grouped together in computer systems to form

2^n different values can be represented with n bits as shown in the table below.

n	Bit Number	Bit Patterns	Different Bit Configuration for 2^n
1	2^1	2	0, 1
2	2^2	4	00, 01, 10, 11
3	2^3	8	000, 001, 010, 011, 100, 101, 110, 111
4	2^4	16	0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111

ⓘ **Binary Code** is based on a number system that uses two digits, 0 and 1.

ⓘ **Bit** is a single binary digit that can have the values 0 or 1.

ⓘ **Byte** is a group of 8 bits. The maximum decimal number it can store is 255.

ⓘ **Nibble** of 4 bi a byte'

Units

The number of bytes can be described using binary prefixes representing powers of 2 or us

Binary Powers			
Name	**Symbol**	**Power**	**Description**
kibi	KiB (kibibyte)	2^{10}	1 KiB = 2^{10}B
mebi	MiB (mebibyte)	2^{20}	1MiB =2^{20}B
gibi	GiB (gibibyte)	2^{30}	1GiB = 2^{30}B
tebi	TiB (tebibyte)	2^{40}	1TiB = 2^{40}B

Name	Syml
kilo	k (kilol
mega	M (mega
giga	G (giga
tera	T (teral

5.3 – Progress Check

8. Define the term 'kibibyte' (2 marks)

5.4 BINARY NUMBER SYSTEMS

Signed and unsigned binary

ⓘ **Signed binary** numbers are encoded to include a positive or negative sign.

ⓘ **Unsigned binary** numbers do NOT have a positive or negative sign so are useful for positive whole numbers only.

The advantage of using unsigned integers is that they can be used to store larger numbers than using signed integers.

An unsigned binary number has a minimum of zero and a maximum value of 2n-1, where n is the number of bits available; see the examples on the right.

Number of bit (n)	M
1	
2	
3	
4	
8	

Unsigned binary addition

Binary numbers can be added using the same techniques used to add denary numbers as s

Example: Add the binary numbers 0111 and 0101.

	0	1	1	1	1 + 1 = 2 so write 0 and carry 1
	0	1	0	1	1 + 0 + 1 = 10 binary so write 0 and carry 1
Sum	1	1	0	0	1 + 1 + 1 = 11 binary so write 1 and carry 1
Carried	1	1	1		0 + 0 + 1 = 1 binary so write 1

So 0111 + 0101 = 1100 binary

Unsigned binary multiplication

Binary numbers can be multiplied using the same techniques used to multiply denary numl

Example: Multiply binary numbers 0011011 by 101.

		0	0	0	1	1	0	1	1	Multiply 0011011
		0	0	0	0	0	1	0	1	By 101
			0	0	1	1	0	1	1	Step 1 multiply 00011011 by 1
	0	0	1	1	0	1	1			Step 2 multiply 00011011 by 1 a
		1	0	0	0	0	1	1	1	Binary addition three rows for fin
Carry		1	1	1	1					Use carry bits where necessary

So 00011011 x 00000101 = 10

Binary Multiplication Rules:
0 x 0 = 0
1 x 0 = 0
0 x 1 = 0
1 x 1 = 1 (there are no carry or borrow bits with binary multiplication)

5.4 – Progress Check

9. Add the following unsigned binary numbers 00000101 and 00000100 (2
10. Multiply the following unsigned binary numbers 00010101 and 0000010

Signed binary using two's complement

Signed numbers are represented in computers using the **Two's Complement** method.

The most significant bit will indicate 1 for a negative number and 0 for a positive number. Two's complement subtraction is performed by simply adding a negative number.

Positive Numbers	
A positive number in two's complement form is the same as an unsigned integer except the most significant bit needs to be zero, representing a positive sign.	The two's complem 1. Inverting eacl and 1's to 0's 2. Adding 1 to tl
0000 1111$_2$ = +15$_{10}$	**0000 1111$_2$ = 111**

Subtraction using two's complement

Example: 15 – 12

15$_{10}$ = 0000 1111$_2$ and
12$_{10}$ = 0000 1100$_2$ which is (1111 0011) + 1 = **1111 0100$_2$** in two'

	0	0	0	0	1	1	1	1	15
	1	1	1	1	0	1	0	0	-12
Sum	0	0	0	0	0	0	1	1	15 – 12 = 3
Carry	1	1	1	1	1	-	-	-	

Signed binary using two's complement range

Binary	Decimal
0111 1111	+127
0000 1111	+15
0000 0001	+1
0000 0000	0
1111 1111	-1
1000 0001	-127
1000 0000	-128

The range of an unsigned Byte (8-b

The range of a signed byte using tv -128 to 127, as shown in the table

The range is – (2^{N-1}) to +(2^{N-1} -1) v number of bits

- In an 8-bit byte the first bit is the sign bit, so only 7 bits contain nu
- The range of numbers for an 8-bit byte is – (2^{8-1}) to +(2^{8-1} -1) or -1
  - ➥ Highest positive number in two's complement 8-bit byte is 2^{N-1} -
  - ➥ Lowest negative number in two's complement 8-bit byte is - 2^{N-1}

5.4 – Progress Check

11. Calculate 24 – 12 using two's complement (2 marks)

Numbers with a fractional part

Fixed-point binary

Fixed-point binary is used to represent a fractional part of a number, where the binary poi

512	256	128	64	32	16	8	4	2	1	.	1/2	1/4	1/8	
Integer Part										Point	Fractional			

Convert fixed-point binary to decimal

For example, to convert **1010000111.10010_2** into decimal, write the binary numbers into shown below.

512	256	128	64	32	16	8	4	2	1	.	1/2	1/4
1	0	1	0	0	0	0	1	1	1	.	1	0

Then add the decimal numbers where 1 is shown:

1010000111.10010_2 = 512 +128 + 4 + 2 + 1 + 0.5 + 0.0625 = <u>**775.5625**</u>

Alternatively: $512 + 128 + 4 + 2 + 1 + \frac{1}{2} + \frac{1}{16} = 775\frac{9}{16}$

Convert decimal to fixed-point binary

To convert **decimal 130.25 to binary**, create a table with the binary place values and use

1. Find the largest place value that is less than or equal to 130.25.
   In this case, it is 128, so write 1 in row below 128.
2. Subtract 128 from 130.25 to obtain 2.25. Find the largest place value that is less
   In this case, it is 2, so write 1 in row below 2.
3. Subtract 2 from 2.25 to obtain 0.25. Find the largest place value that is less thar
   In this case, it is 0.25, so write 1 in row below 0.25.
4. Complete the process by adding 0 to all the blanks in the table.

512	256	128	64	32	16	8	4	2	1	.	1/2	1/4	1/
0	0	1	0	0	0	0	0	1	0	.	0	1	(

5.4 – Progress Check

12. Convert fixed-point binary number 1001111100.1010 into decimal (3 ma

5.5 INFORMATION CODING SYSTEMS

Character encoding

When a character is pressed on the keyboard, a binary code for that character is input into the computer; characters can be encoded into either ASCII or Unicode.

There is a need to differentiate between the encoded decimal digits and their pure binary numbers; this is shown in the ASCII table below, where the ASCII code for the decimal 1 is represented by the pure binary number representing 49.

ⓘ **ASCII** is a seven-bit character set which offers 128 different characters. Some of these characters are used for controlling peripherals.

Every character on the keyboard has an ASCII code and lower-case letters have different codes to upper-case letters.

The main limitation with ASCII is that it does not have enough characters to cope with languages that have large character sets.

ASCII	Decimal	Binary
1	049	011 0001
2	050	011 0010
3	051	0110011
4	052	0110100
5	053	0110101
6	054	0110110
7	055	0110111
8	056	0111000
9	057	0111001
:	058	0111010
;	059	0111011
<	060	0111100

ⓘ **Unicode** is 65,536 dif characters

Unicode has be all major web b

Uni
0000 (
0000 (
0000 (
0000 (
0000 (

Note that ASCI Unicode uses m per character; /

Error checking and correction

Parity checking

Error checking is frequently carried out on data that is transmitted and parity is a simple m data that has been received. Parity checking is only able to detect an odd number of erro

ⓘ **Even parity** is where the number of 1s is counted and if the number of 1s is odd then

In the example below the 7-Bit ASCII code for Q is 1010001, which is an odd number of 1

Parity Bit	ASCII Code for Q				
1	1	0	1	0	0

ⓘ **Odd parity** is where the numbers of 1s is counted and if the number of 1s is even the

In the example below the 7-Bit ASCII code for S is 1010011, which is an even number of 1

Parity Bit	ASCII Code for S				
1	1	0	1	0	0

Majority vote

The parity bit approach can only detect errors in data transmission, but the majority vote c

With **majority voting**, each bit is sent three times and checked each time. If it is the s on one occasion there is an error, in which case the majority of two will determine th

In the example shown data sent is 01100, each bit will be transmitted three times.

Data transmitted will be	000	111
With interference errors data received is	100	101
Each triplet is then checked and majority accepted	000	111

Main disadvantage of majority vote correction method is three times the volume of data n

Check digits

A **check digit** is added to binary data to check that the data is accurate; this is an appl bar code printed on an item of shopping has been correctly entered into a computer.

The check digit is a single digit that is calculated using a mathematical algorithm and ins

A typical method used is based on modulo-11 as shown in the example below:

1	2	8	0	2	4	Original number
7	6	5	4	3	2	Weighting – each of the origi
7	12	40	0	6	8	Multiply weights and add the
73 divided by 11 = 6 remainder 7						Divide total by 11 since it is a
11 – 7 = 4						Subtract the remainder from 1
1280244						Original number with check di then an error will be detected

5.5 – Progress Check

13. Compare the use of ASCII and Unicode systems for character encoding (4

14. Explain, with an example, even parity for error detection (2 marks)

5.6 REPRESENTING IMAGES, SOUND AND OTHER DATA

Bit patterns

ⓘ A **bit pattern** is an arrangement of binary digits arranged in a sequence; they can be u
Unicode) as well as images (bitmaps and vector graphics), video and sound.

Analogue and digital

	Analogue	
Data and Signals	Analogue data and signals vary in a continuous way. For example, the output from a microphone into a tape recorder is an analogue signal that varies as a function of the pressure of the sound.	Digital data and sig binary digits or a se For example, a sou and stored as a ser

Conversion between analogue and digital data

It is necessary to convert analogue signals using an **analogue-to-digital converter** (ADC) i
input and processed in a computer. The output from a computer may need to be convertec
digital-to-analogue converter (DAC) to interface with an analogue device.

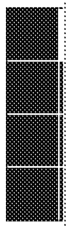
Analogue to Digital Converter (ADC)	Digital to
Analogue to digital converters receive analogue signals input in the form of voltage waveforms which are converted into a digital format so they can be read into a computer.	Digital to analogue co signal into a continuo analogue sound outpu

Bitmapped graphics

Digital images are composed of a series of **pixels** or dots, so a pixel is the smallest element in a digital image.

Images are represented in digital or binary form to be stored or used on a computer. The black and white image of a staircase is shown on the right, where for each pixel 1 = black and 0 = white.

Colour can be shown in digital images by using more bits per pixel.



ⓘ **Colour depth** is the number of bits used for each pixel, where 1 bit is black and white and 8 bits will give 256 colours.

ⓘ **Image resolution** is calculated using the pixel dimensions (height x width).

ⓘ **Display Resolution** is the number of dots per inch (DPI).
More dots per inch give a better image resolution.

ⓘ **Metadata** is 'data about data'; metadata in an image file provides details of properties such as the image dimensions and colour depth.

Bit
8
16

Image metadata and storage requirements

Metadata is used by the computer to help it interpret an image file, since all data is basically stored in binary format.

The image on the right shows some typical metadata properties for a digital image.

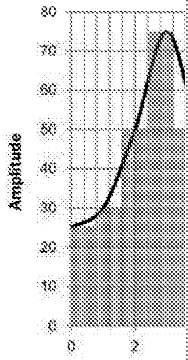
Image storage requirements are calculated using he following formula:
Storage = width (pixels) x height (pixels) x colour depth (bits per pixel)

Digital representation of sound

When sound is input into a computer, it will be converted into a digital data file using an ADC.

In the graph on the right, the analogue sound input is sampled at regular intervals; the converted digital sound is shown superimposed on the analogue wave.

The digital sound is only approximately the same shape as the analogue wave; the sound quality can be improved by increasing the sample rate.



ⓘ **Sampling rate** (measured in Hz) is the number of samples taken per second from the analogue input to create a digital signal.

ⓘ **Nyquist's theorem** indicates that the sound must be sampled at twice the highest analogue input frequency to create an accurate representation of the original input waveform.

ⓘ **Sampling resc** number of bit

Sound Sample Size Calculation

Example: Given that: Sample frequency = 4000 Hz, Sample resolution = 16 bits (2 bytes
How much disk space would a 120-second sound recording require?

File size (bytes) = sample frequency (Hz) x sample resolution (bytes) x leng
= 4000 x 2 x 120 = <u>960,000 bytes</u>

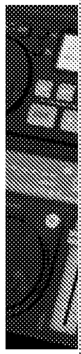
Musical Instrument Digital Interface (MIDI)

ⓘ **MIDI** (Musical Instrument Digital Interface) is a protocol that is used to synthesise mu recording and playing back music input from keyboard, voice and other musical instru

MIDI makes use of **event messages** that are used to control musical parameters such as notation and pitch and volume, and to synchronise the rhythm between a range of other devices.

The main advantages of using a MIDI system are:

- Music data that has been loaded onto the computer can be arranged and manipulated in many different ways.
- A musician or music producer can make use of overlays to create a whole band sound.
- It is straightforward to create a musical score from the edited music.



5.6 – Progress Check

15. Describe the functions of an analogue to digital converter and a digital t
16. Define the following digital image terms:
    (a) Colour depth (2 marks)   (b) Resolution (2 marks)   (c)
17. Calculate the memory used in kilobytes for a digital photographic image pixels high, with colour depth 24 bpp (bits per pixel) where 1 byte = 8 bi
18. Define the following digital sound terms:
    (a) Sampling rate (1 mark)   (b) Sampling resolution (1 mark)

Data compression

Data compression is important in data transmission via the internet, as compressed data creates smaller files so that they can be transmitted faster and require less storage space on the computer system.	ⓘ **Data cor** downloa also be r

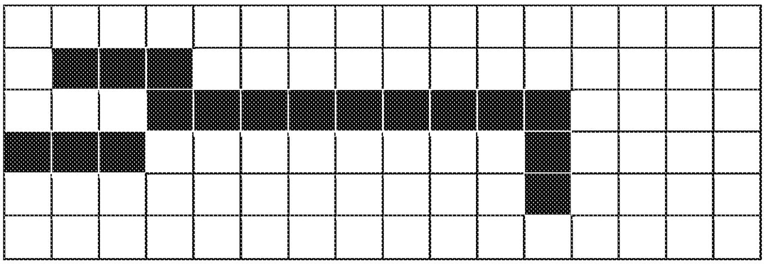
Lossy and lossless compression

Compression Method	Advantages a
ⓘ **Lossless** compression techniques allow the original data to be perfectly reconstructed, resulting in no loss of data.	An image can be compressed to low resolut also be needed in a higher resolution in a p
	A program may be compressed to be downl need to be expanded to an exact copy of th
	A text document might be compressed into transmission using email. Again the docume exact copy of the original to ensure that it c
ⓘ **Lossy** compression techniques result in a loss of data, so the original data cannot be perfectly reconstructed.	A copy of an image might be compressed to website, but there is no need to expand bac
	In sound files unnecessary data can be redu output quality is acceptable for the user.

Run-Length Encoding

ⓘ **Run-length encoding (RLE)** is a form of lossless compression where a sequence that replaced by a single value of that data with a count of the number of times it occurs.

The black and white bitmap image (16 x 6 pixels) shown below can be used to demonstrate



This image can be written in the raw format below where a white pixel = 0 and a black pi

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Run-length encoding can be used to represent the data in the following compressed form

Row	RLE Code	Description
1	16,0	16 '0' elements
2	1,0 3,1 12,0	1 '0', 3 '1' and 12 '0' elements
3	3,0 9,1 4, 0	3 '0', 9 '1' and 4 '0' elements
4	3,1 8,0 1,1 4,0	3 '1', 8 '0', 1 '1' and 4 '0' elements
5	11,0 1,1 4,0	11 '0', 1 '1' and 4 '0' elements
6	16,0	16 '0' elements

The RLE bitmap fi bitmap i that the

This is a reformec

Dictionary-based compression

The **dictionary-based** compression method (sometimes known as a substitution technique contains encoded bit strings that contain fewer bits than the original code, so:

- Variable-length strings of symbols are encoded as single symbols or tokens.
- The token is used as the index for the look-up table or dictionary.
- Compression takes place where the tokens are smaller than the original variable

This is a lossless compression method as the original file can be created by using the dicti where large amounts of data can be referenced by a simple token, such as a post code.

Encryption

ⓘ **Encryption** is used to make stored data more secure from hackers, by making it unrea to decrypt or decode it. This method is commonly used to protect data transmitted ov

Encryption operates by modifying plain text using an encryption algorithm; this takes plac stipulates how the message will be encoded. An authorised user is able to decode this me key based on a decryption algorithm.

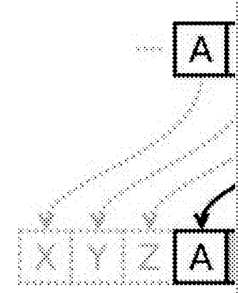
Caesar Cipher

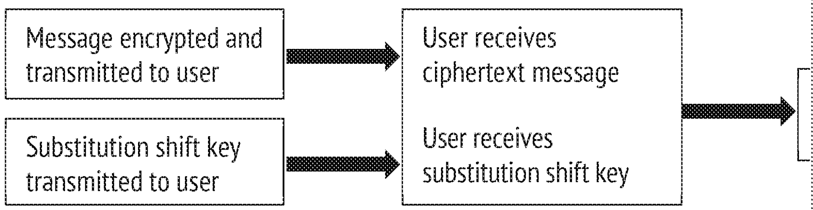
The Caesar Cipher is a commonly used and simple substitution cipher; using this technique each of the plaintext letters in a message is replaced by a different letter of the alphabet a fixed position from the original letter.

The cipher shown on the right is replacing each letter with a different letter using a left shift of 3; therefore, E is represented by B, F is represented by C and so on.

Therefore, the ciphertext for the message will substitute 'B' for each 'E' that appears in the plaintext. Messages are transmitted using the approach shown in the diagram below.

ⓘ A **cipher** algorithn creating ciphertex and turning it bac





The table below can be used to convert between the ciphertext and the plaintext for a lef link between E and B is highlighted.

Plain:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Cipher:	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O

Encrypt using the table above and replace each letter of the plain text message with the l

Ciphertext: **ZLJMRQBO PZFBKZB**

The message is decrypted by using the table to reverse the process, so that each letter of above in the plain text line.

Plaintext: **COMPUTER SCIENCE**

The Caesar is easily cracked as there are only shifts between 1 and 25 so each can be trie then the shift can be applied to all the text to crack the cipher.

Vernam Cipher

ⓘ The Vernam cipher or 'one-time pad cipher', is a digital data stream cipher that is combined with a same length random stream of date to create the ciphertext, based on the Boolean exclusive OR function.

The truth table for the Boolean **Exclusive OR** function is shown on the right.

Example:
To encrypt the ASCII code for 12 using the Vernam cipher use the following steps:

Plaintext – ASCII (1 and 2)	0110001 0110010	14 characters
Random key chosen	1010101 0111100	Any random key of the same length
Ciphertext	1100100 0001110	Created from Exclusive OR Plaintex

Decryption is the reverse of this process as shown in the table below:

Ciphertext	1100100 0001110	14 characters
Random key	1010101 0111100	Use the same random key chosen fc
Plaintext – ASCII (1 and 2)	0110001 0110010	Decrypted from Exclusive OR Ciphe

The Vernam cipher is known as a one-time pad as the key must only be used once.

If the key is randomly chosen and used only once, unlike other ciphers, the resultant ciphe matter how much time is spent trying to crack the code, so the Vernam cipher is a perfect

ⓘ **Computational security** is a term used to indicate the effectiveness of a cipher.

As stated, the Vernam cipher is considered to be 100% mathematically secure.

In theory, every other cipher can be broken given sufficient ciphertext and time; encryption Caesar cipher) can be cracked by the recognition of patterns or by the dictionary process of i

5.6 – Progress Check

19. Explain the difference between lossless and lossy compression (4 marks)
20. Define the terms 'encryption' and 'cryptography' (2 marks)
21. Briefly describe the Caesar cipher algorithm and decrypt the ciphertext:

    CU EQORWVGT UEKGPEG

    that has been encrypted with a Caesar cipher and a right shift of two (6 r

TOPIC 6 – COMPUTER SYSTE

6.1 HARDWARE AND SOFTWARE

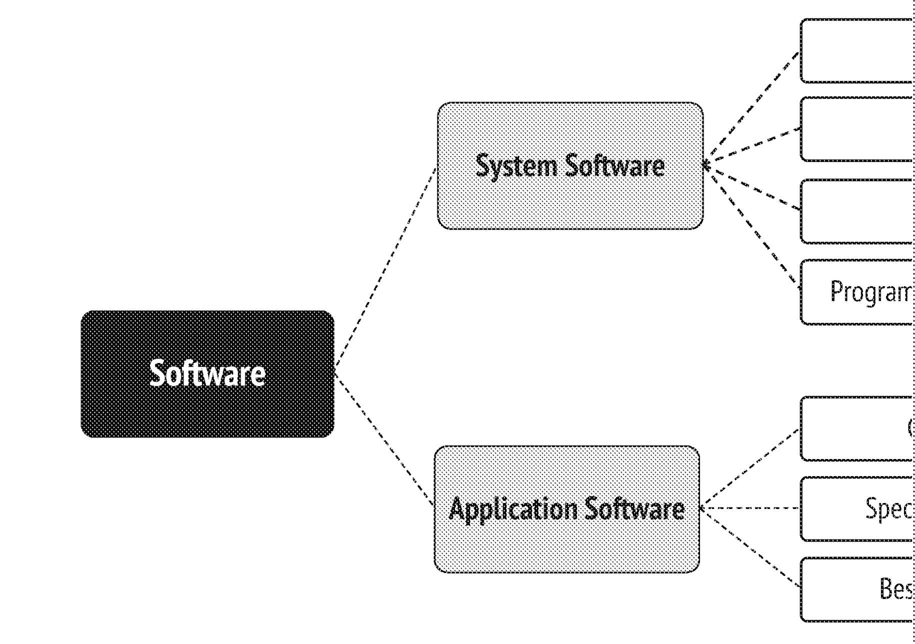
Relationship between hardware and software

A computer system operates with hardware and software to create a functional solution.

Computer hardware is the physical part of a computer, which includes digital circuitry, as distinguished from the applications software that is executed using the hardware. The processing hardware is necessary to gain a useable output from the system.

Software can be classified as systems software and applications software – the diagram below shows the types of software included in each.

ⓘ **Softwar** instructi
ⓘ **Hardwa** up a com
ⓘ **Systems** tasks to
ⓘ **Applicat** out user (GP), Pay



Systems software

Software	Main Features
Operating System (OS)	The operating system (OS) is the software that controls the hardware hardware by creating a platform to run application software. Hiding the complexity of the hardware with a Windows-based operat machine that has a user-friendly interface and operates in the same r of the computer system. Main functions include: resource management and the creation of us conflicts where a resource, such as a printer, is requested by more th time; the OS manages these resource requests in a systematic way. O be managed are processor allocation, peripheral hardware devices, m
Library Programs	Library programs are pre-written software that is stored in compiled f programmer within one or more programs. Library functions are wide interacting with peripherals such as printers. A typical example of a library function from the mathematics library calculate a square root.

Software	Main Features
Utility Programs	Utility programs are a range of systems software that is designed to h optimise the system. Main functions include: disk formatting, file com memory testing and anti-virus protection.
Programming Language Translators	Programming language translators are used to translate a program in different programming language and to maintain the functionality of using one of the following software programs: 1. Assembler – used for assembly language programs 2. Compiler – used for program languages such as: C++, Visual I 3. Interpreter – used for program languages such as: some vers

6.2 CLASSIFICATION OF PROGRAMMING LANGUAGES

Programming languages

Programming languages can be classified as listed below.

Language	Sample Code
Low-level (1st Generation) Machine Code	Load data into register 8, taken from memory cell 68 where location (are listed in register 3: <table><tr><th>Operation</th><th>Register Operations</th><th></th><th>Memory Address</th></tr><tr><td>35</td><td>3</td><td>8</td><td>68</td></tr><tr><td>100011</td><td>00011</td><td>01000</td><td>00000 00001 00100</td></tr></table>
Low-level (2nd Generation) Assembly Language	`SET r1, 12     ; set register 1 to 12` `STORE A, r1    ; store register 1 contents into variab` `LOAD r2, A     ; load variable A into register 2` `STORE B, r2    ; store register 2 contents into variab` The ';' symbol is used to add comments
High-level	`IF (ExamScore > 55) THEN` `      ExamGrade = "Pass"` `ELSE` `      ExamGrade = "Fail"` `END IF` Sample to create database table in SQL: `CREATE TABLE Subjects (` `Subject_Id int NOT NULL AUTO_INCREMENT,` `Student _Id int,` `Subject varchar(20),` `Subject_Mark int,` `FOREIGN KEY (Student_Id) REFERENCES Students` `(Student _Id) )`

Imperative Languages

High-level languages include imperative languages, which is where the program statemen sequence or order as defined by the programmer. These languages, sometimes known as p subroutines and functions to aid readability and hence maintainability.

Low-level languages are imperative as all instructions are executed in a set sequence; eac is translated into numerous machine code statements prior to the execution of the code.

Machine-code and assembly languages compared to high-le[v]	
Advantages	Programming in low-level code can create faster and more efficien[t] performance level of the created code; with high-level programmin[g] create optimised code.
	Low-level language code is memory efficient due to the lack of abs[…] high level languages.
Disadvantages	It is difficult to learn to program in low-level languages whereas hi[…] and training options.
	High-level code can be self-documenting which makes it more und[…] maintenance and debugging.
	Machine-code and assembly programs are specific to a limited rang[…] can be compiled to run on a wide range of processors.

6.2 – Progress Check

1. Explain the difference between hardware and software (4 marks)
2. Describe the following:
   (a) Machine code (2 marks) (b) Assembly language (2 marks)
3. Describe the advantages and disadvantages of machine code and assem[…] high-level languages (6 marks)

6.3 TYPES OF PROGRAM TRANSLATOR

Types of program translator

ⓘ **Source code** is the language instructions that have been written by the computer programmer. The computer cannot execute the source code directly.	➡	ⓘ **Object Code** (or e[…] source code using[…] instructions that c[…]

Programming languages can be classified as listed below.

Translator	Characteristics	
Assemblers	The source code is written in assembly language, which is a series of me[…] machine operational codes; assembly code is the most difficult language[…] expertise to write and debug it. The assembler translates this source code into machine code that the co[…]	
	Advantage: it is an efficient low-level language that can be translated quickly as it has a one-to-one relationship with machine code.	**Disadvantage:**[…] programming t[…]
Compilers	The source code created by the programmer is not understandable by th[…] a compiler converts the source code to object code, which the computer[…] match the target computer that will run the software.	
	Advantage: an executable file is produced which runs without the need of the source code; this makes the source code more secure as it does not need to be distributed to the customer.	**Disadvantages**[…] long time to c[…] code need to b[…] be produced; s[…] to correct the[…]

Translator	Characteristics	
Intermediate Language	Some compilers create a final output in an intermediate language, such a run on a virtual machine (VM) rather than on a central processing unit (C program is portable and not machine dependent when in VM format. A bytecode interpreter is used to translate the VM code into machine lan code is executed. Although the bytecode interpretation takes time, the fi more efficient than if the source code was compiled directly; this approa	
Interpreters	Interpreter software normally executes the source code directly, it is tran to compile the program. Interpreted software runs more slowly than compiled software as each st before it is executed.	
	Advantage: during development the programmer might make frequent changes, which can be tested without going through the time-consuming process of compiling and linking for each change.	**Disadvantage:** the target com machine code Additionally, t customer and t longer than a c

6.3 – Progress Check

4. Define the terms 'source code' and 'object code' (4 marks)
5. Describe the advantages and disadvantages of using an assembler as a tr
6. Describe the advantages and disadvantages of using a compiler as a tran
7. Describe the advantages and disadvantages of using an interpreter as a t

6.4 LOGIC GATES

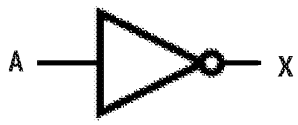
Logic gates

Binary numbers are stored in a computer system as different voltage levels where 0 is low voltage and 1 is high voltage.

The electronic circuits used in computers can be wired together to form logical operations; these circuits are based on logic gates.

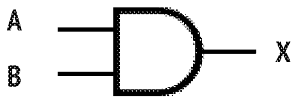
NOT

ⓘ The output is the inverse (opposite) of the input.



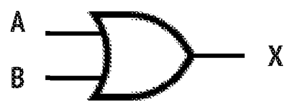
AND

ⓘ Two or more inputs where the output is true if all inputs are true.



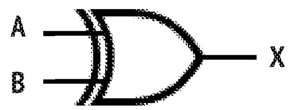
OR

ⓘ Two or more inputs where the output is true if either or both inputs are true.



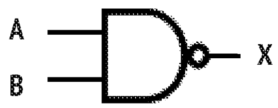
XOR

ⓘ Two or more inputs where the output is true ONLY if one input is exclusively true.



NAND

ⓘ Two or more inputs where the output is false when all inputs are true.

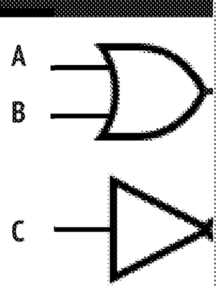


NOR

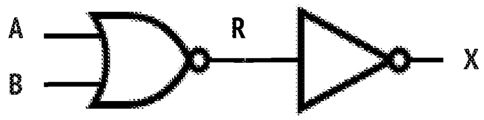
Two or more inputs where the output is true when all inputs are false.

Logic circuits and Boolean algebra

Logic gates can be combined to form **logic circuits**, for example the NOR and NOT gates shown below with the truth table for the circuit.



ⓘ **Boolean expressions** are used in computer programs; when these expressions are evaluated they give a Boolean Logic output of either True or False.



A	
0	
0	
1	
1	

Boolean algebra is used to show logical expressions in the truth table, such as:

'R = (A NOR B)'

The Boolean expression for the complete logic circuit is:

'X = NOT (A NOR B)'

Drawing logic diagrams from Boolean expressions

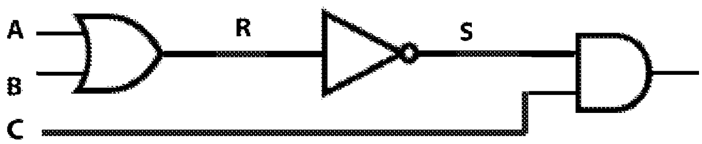
Logic diagrams can be drawn from Boolean expressions, such as: **X = (A OR B) AND (NOT**

Draw complex logic diagrams by first drawing the bracketed terms, so in this case:

1. Draw the logic gate for A OR B
2. Then draw the logic gate for NOT C
3. Finally draw both inputs connected to an AND gate

6.4 – Progress Check

8. Use the following complex logic diagram to work out the outputs for va



(a) State the outputs R, S and T when the following inputs are:
A = 0 B=1 C = 0 (3 marks)

(b) State the outputs R, S and T when the following inputs are:
A = 0 B=0 C = 1 (3 marks)

(c) State the outputs R, S and T when the following inputs are:
A = 1 B=1 C = 0 (3 marks)

(d) State the outputs R, S and T when the following inputs are:
A = 1 B=1 C = 1 (3 marks)

9 Draw the logic circuit for Q = (A OR B) AND (C OR D) (3 marks)

6.5 BOOLEAN ALGEBRA

Using Boolean algebra

De Morgan's laws and Boolean identities can be used to simplify and manipulate Boolean can be produced using NAND or NOR gates.

Some useful Boolean identities are shown below, where + is a logical OR and '.' is a logica

AND logic identities	OR logic identities	
$1 . A = A$	$0 + A = A$	$A . B = \overline{\overline{A}}$
$0 . A = 0$	$1 + A = 1$	The laws a
$A . A = A$	$A + A = A$	1. Char
$A . \overline{A} = 0$	$A + \overline{A} = 1$	2. Char
$A . B = B . A$	$A + B = B + A$	3. Char **A =**
$A . (B . C) = (A . B) . C$	$A + (B + C) = (A + B) +. C$	4. Chan
$A + (B . C) = (A + B) . (A + C)$	$A . (B + C) = (A . B) + (A . C)$	expr **exp**
$A . (A + B) = A$	$A + (A . B) = A$	(and

Example

Simplify the following Boolean expression:

$$\overline{A.B} . (\overline{A} + B) . (B + \overline{B})$$

$\overline{A.B} . (\overline{A} + B) . (B + \overline{B})$	Original Boolean expression
$= \overline{A.B} . (\overline{A} + B)$	Remove identity $(B + \overline{B}) = 1$
$= (\overline{A} + \overline{B}) . (\overline{A} + B)$	Apply De Morgan's law to $\overline{A.B}$
$= \overline{A} + (\overline{B} . B)$	Factorise for $\overline{A}$
$= \overline{A}$	Remove identity $(B . \overline{B}) = 0$

6.5 – Progress Check

10. Simplify the following (3 marks):

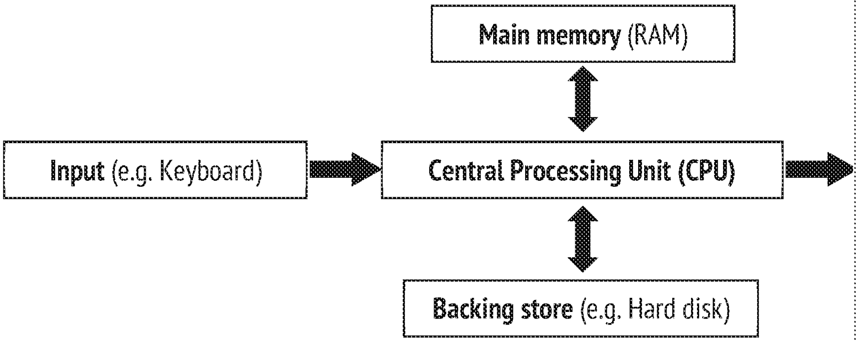
$$\overline{A}.B.C + \overline{A}.C$$

TOPIC 7 – COMPUTER ORGANISATION ANI

7.1 INTERNAL HARDWARE COMPONENTS OF A COMPUTE

Internal hardware components of a computer

All ICT systems have these same essential components and some typical examples of devic
The main part of the system is the CPU, which is a single chip responsible for all the proces



Computer hardware	The physical components of the computer system, which includes The processing hardware is necessary to gain a useable output fr
Backing store	Normally a hard disk, which retains the data written on to it after switched off; it could include other storage devices such as flash
Input devices	Typically includes a mouse, keyboard and microphone.
Output devices	System output can be printed out by the user or displayed on a c
Power supply	Used to convert alternating current (AC) into low-voltage direct c internal components within the computer system.

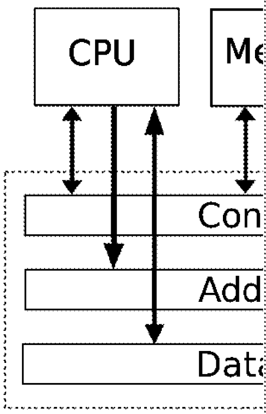
The processor is responsible for following instructions in order to process data and produce

- Sorting and searching data
- Performing calculations
- Logical decision making
- Control of input, output and storage devices

Bus system

The bus system is composed of the communication links which connect the various parts of a computer.

The three buses involved are the address, data and the control bus; the direction of data flow for each bus is shown.



ⓘ **Main memory** stores program instructions and data that are processed by the processor.

ⓘ **Address bus** specifies a physical address in main memory and the value that is to be read or written into that address is communicated via the data bus.

ⓘ **Control bus** manages data processing; for example, sends a signal to either write from the data bus to a memory address or a signal to reac

ⓘ **Input and output** – control signals received from the processor include input (read) or signals are then used to connect a system bus to specific I/O devices; typical I/O device mouse with outputs to monitor and printer.

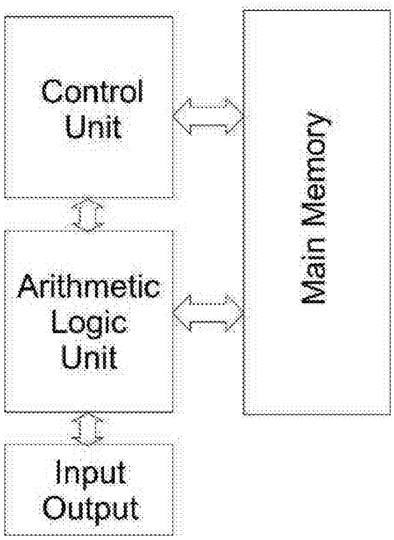
Computer architectures

Van Neumann architecture

The Van Neumann stored program architecture is based on both data and instructions stored in a single memory space.

Data and instructions are transmitted through a shared data bus.

Van Neumann architecture is used extensively in general purpose computing systems.

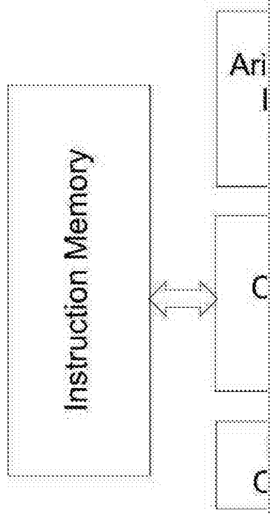


Harvard architecture

The Harvard architecture is based o[...] being stored in separate memories; [...] memory serially and data is fetched [...]

The Harvard technique may be fast[...] and data can take place in parallel.

Embedded systems such as digital s[...] Harvard architecture extensively. T[...] machines and burglar alarms make [...]



Addressable memory

The processor needs to be able to address individual memory locations, so each memory l[...] numeric code.

To select a memory location the following steps are taken:

1. The processor writes the numeric code into the address bus
2. The processor sends a request to the control bus with instructions to read or wri[...]
3. Finally, the data that is being read from or written to will be transferred via the [...]

7.1 – Progress Check

1. Briefly describe the function of the following computer architecture term[...]
   (a) CPU (2 marks)
   (b) Main memory (2 marks)
   (c) Address bus (2 marks)
   (d) Control bus (2 marks)
   (e) I/O controller (2 marks)
   (f) Arithmetic logic unit (2 marks)
2. Describe, with the aid of a sketch, the Van Neumann program architectur[...]

7.2 THE STORED PROGRAM CONCEPT

The stored program concept

The program is stored in main memory and machine code instructions are fetched and executed serially in the processor. The example below shows a typical arithmetic operation using the fetch-execute cycle (described fully in section 7.3).

CPU

Main Memory
Stored in binary

1	ADD	(instruction)
2	9	(data)
3	5	(data)
4		
5		
.		
.		
.		
n		

Arithmetic and Logic Unit
Operations based on
+, - x / AND OR

Fetch – the first instruction fetched is the ADD i
5) fetched are determined by the control unit as

Decode – translate the instruction so the proce
is to be added to the second number.

Execute – Calculate 9 + 5 = 14.

Note that the process is serial and line 1 go

7.3 STRUCTURE AND ROLE OF THE PROCESSOR AND ITS C

The processor and its components

Arithmetic logic unit	Performs arithmetic and logical operations, such as fixed point and floati point arithmetic, logical operations (AND, OR, XOR) and shift operations. Inputs are from the internal bus and the accumulator, with outputs route to one of the registers.
Control unit	Main function is to fetch program instructions from memory, to decode them and to execute them serially.
Clock	System clock sends a signal to each of the computer components on a regular basis to synchronise all of the computer operations.
Register	A fast memory location that exists in the processor or the I/O controller. The computer consists of general purpose and dedicated registers.

ⓘ **General purpose registers** (normally named R0, R1, R2, etc.) can be used to hold inst

ⓘ **Dedicated registers** are used by the processor to carry out a specific role:

Accumulator (ACC)	Special register used as fast temporary storage b
Status register (SR)	Used to hold the status of various flags indicatin carry bit used, overflow error, and interrupt statu dependent upon the processor involved.
Program counter (PC)	Register that holds the address of the next instruc
Memory address register (MAR)	Holds the address of the current instruction being
Memory buffer register (MBR)	Holds the instruction from the (MAR) and the asso
Current Instruction register (CIR)	Used to store the instruction that is to be decode and executed.

7.3 – Progress Check

3. Define the term 'register' and the difference between general purpose an
4. Briefly describe the function of the following dedicated registers:
   (a) Status register (2 marks)
   (b) Program counter (2 marks)
   (c) Memory address register (2 marks)
   (d) Memory buffer register (2 marks)
   (e) Current instruction register (2 marks)

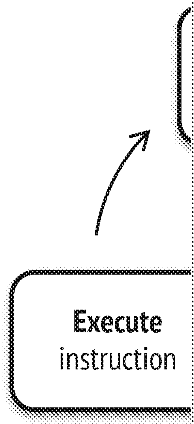
The fetch-execute cycle and the role of registers within it

After the computer is switched on it performs the fetch-execute cycle; this process reads the main memory (RAM) and these instructions are then executed by the processor.

Once the instruction has been executed the process is repeated to read the next instruction are described below:

Fetch:

1. Program counter (PC) points to next instruction to be fetched.
2. Contents of (PC) copied into memory address register (MAR).
3. The instruction in (MAR) is transferred via the data bus to the memory buffer register (MBR).
4. The contents of the (MBR) are copied to instruction register (CIR) and the program counter (PC) is updated to address next instruction.



Decode:

The instruction in the (CIR) is decoded.

Execute:

1. The decoded instruction is executed.
2. The process is repeated.

7.3 – Progress Check

5. Describe, with the aid of a sketch, the fetch-execute cycle and the role of

The processor instruction set

ⓘ **Machine code** instructions are written in binary code that the processor can interpret an

It is convenient to write software in high-level languages; these high-level languages are cor code that can be understood and executed by the processor.

ⓘ The **instruction set** is the set of the machine code instructions that a particular type of

Two different types of processor can only be compatible if they are able to operate the sam the following operations:

- Arithmetic operations (ADD, SUB)
- Data transfer operation (MOV, OUT)
- Logical operations (AND, OR)
- Jump operations (JMP, JZ – jump if zero)

A machine code instruction consists of an operation or op-code and an operand, as shown i

Op-Code	Operand
4 bit	12 bit

ⓘ **Op-code** instructions are the part of a machine code instruction that represents a basic

ⓘ **Operand** is value or memory address that forms part of a machine code instruction.

Instructions consist of an op-code and an operand, where the op-code contains a basic mac a value or a memory address.

Some typical examples are shown in the table below:

Op-code	Operand	Machine Operation	E
0001	0000 0000 1111	Load into accumulator	Load the contents of the ope
0100	0000 0000 0111	Add 7	Add 7 to contents of accumu
1000	0000 0000 0011	Store to main memory	Store the contents of the acc

* **Note**: you will be expected to interpret op-codes rather than define them in the given cor

Addressing modes

Instructions and data are located in memory by using addressing modes; the specification r and direct addressing modes.

ⓘ **Immediate addressing** means that the data in the operand is fixed; in other words, it is

This is a very fast addressing mode since the data is readily available, rather than needing to obt

Typical example: `ADD 12` – this instruction could be used to add 12 to the accumulator.

ⓘ **Direct addressing,** or absolute addressing, means that the code is directly referred to a

Example: `ADD (1302)` – this instruction adds the contents of memory location (1302) to

The disadvantage of using code that directly refers to memory addresses is that the code ca typically used on single-program systems, such as a car engine fuel control system.

7.3 – Progress Check

6. Briefly describe the following processor instruction set terms:
   (a) Machine code (2 marks)
   (b) Op-code (2 marks)
   (c) Operand (2 marks)
   (d) Instruction set (2 marks)
7. Compare the immediate and direct addressing modes (4 marks)

Machine code / assembly language operations based on ARM (Raspberry Pi

ⓘ **Assembly language** is used to make machine code instructions more understandable t

ⓘ **ARM** is a load-store architecture meaning that memory can only be accessed by:

- Loading from memory into a register
- Storing the result back into memory

Basic machine code operations that can be expressed in mnemonic form (assembly languag

Immediate address examples: `LDR r1, #12` `ADD r2, r1, #5` `SUB r3, r2, #3` `STR r3, 1203`	Immediate address loads the data without reference to an add *# indicates that the operand is a number* 1. Loads the number 12 into the register r1 2. Adds the number 5 to value in r1 and stores result ir 3. Subtracts the number 3 from value in r2 and stores r 4. Stores the contents of the register r3 (14) into mem
Direct address examples: `LDR r1, 12` `ADD r2, r1, 13` `SUB r3, r2, 14` `STR r3, 150`	In this case the operand is a memory address 1. Loads data in memory address 12 (4) into r1 2. Adds the data in memory address 13 (5) to r1 and stores result in r2 (making 9) 3. Subtracts the data in memory address 14 (3) from the r2 and stores result in r3 (making 6) 4. Stores the result in r3, which is 4 + 5 − 3 = 6 into memory address 150

Compare is used to compare two values; the result is normally used as a setup for a conditi

`CMP r1, #23`	compares r1 with the number 23 (immediate addressing).
`CMP r1, 23`	compares r1 with the number stored in memory address 23 (di
`CMP r1, r2`	compares r1 with register r2

Branching

ⓘ **Branching** is used for conditional statements or unconditional statements

ⓘ A **label** is a sequence of characters that identities a location in computer source code.

It is commonly used in branch or jump instructions in assembler code.

ⓘ **Unconditional branching** is a command where a section of code is jumped over, with r

```
        ...            some code
        B Next         unconditional branch to label 'Next'
        ...            some more code
Next                   label 'Next'
        ...            code continues
```

ⓘ **Conditional branching** is a command where a section of code is jumped over based on

```
        ...            some code
        CMP r1, r2     compares contents of registers r 1 and r2
        BEQ End        branch to label 'End' if contents of r1 = r2
        ...            some more code
End                    label 'End'
```

Other branches include:

BNE	Branch not equal
BLT	Branch less than
BGT	Branch greater than

High-level to assembly code conversion

Example

```
Total ← 0
FOR X = 1 TO 5
      Total ← Total + 10
END
```

Use r1 to store value of Total and r2 to store the value of X

```
      MOV  r1, #0         initialise Total = 0
      MOV  r2, #1         initialise X = 1
loop  CMP  r1, r2         compare the value of r2 with #5
      BNE  end            r1 = r2 so branch to end of loop label
      ADD  r1, r1, #10    add 10 onto total
      ADD  r2, r2, #1     increment X
      B    loop           branch to start of loop
end   HALT                loop completed
```

Logical bitwise operator

ⓘ **Logical bitwise operator** is where a logical operation is carried out on each column of

In bitwise operations, the '#' symbol in the instructions below indicates that the operand is immediate addressing. Note – where no # symbol is used indicates that the operand is a m

`AND r2, r1, #1101`	Operand1 0011 Operand2 <u>1101</u> **AND** result **0001**	Performs a bitwise logical A register r1 (#0011) and the The result (#0001) is stored
`ORR r2, r1, #1101`	Operand1 0011 Operand2 <u>1101</u> **OR** result **1111**	Performs a bitwise logical O register r1 (#0011) and the The result (#1111) is stored
`EOR r2, r1, #1101`	Operand1 0011 Operand2 <u>1101</u> **XOR** result **1110**	Performs a bitwise logical E the value in register r1 (#00 The result (#1110) is stored
`MVN r2, #0011`	Operand <u>0011</u> **NOT** result **1100**	Performs a bitwise logical N The result (#1100) is stored

ⓘ **Logical shift** is a bitwise operation where all the bits of an operand are shifted left or completed the vacant-bit positions are filled with zeros.

These bitwise operations are performed on unsigned integers as the sign bit is not preserv zero. See examples below for further details.

Logical Shift Left `LSL r1, r2, 2` operand is shifted left by 2 places	The value stored i places and the res So before shift After shift
Logical Shift Right `LSR r1, r2, 5` operand is shifted left by 5 places	The value stored i places and the res So before shift After shift

ⓘ **Halt** simply terminates the program operation

`HALT` no operand needed for halt instruction

7.3 – Progress Check

8. Convert the following program code into assembly language instructions

```
IF (A > B) THEN
      P ← 3
ELSE IF (A = B) THEN
      P ← 1
ELSE
      P ← 0
END IF
```

Use r1 to store A, r2 to store B and r3 to store P

Factors affecting processor performance

The factors and their impact on processor performance are outlined below:

Factor	Effect on processor performar
Clock Speed	The clock speed is a key factor in the performance of a computer. It controls the process of executing an instruction as well as fetchir has increased regularly in recent years. Modern computer devices run at a clock speed in excess of 2.5 GHz second. Computer performance can easily be increased by purchasi the device the more heat it produces and so additional cooling is n
Number of Cores	System performance can be improved by increasing the number of using a multi-core processor. A multi-core processor has more than one processor incorporated i dual-core processor will provide two processors on one chip, which processor.
Cache Memory	Cache is high-speed memory that is fitted close to or on the proces Programs run faster using cache memory as it can be accessed faste regularly used data and instructions to further improve performanc
Word Length	Most modern processors use 32-bit or 64-bit word lengths; the perf an increase in word length, providing that the word length and data So, using a 64-bit word length with a 64-bit bus will allow 64 bits c pulse.
Address Bus Width	The width of the address bus affects the amount of memory that ca allowing more memory to be installed on the system or embedded
Data Bus Width	The width of the data bus directly affects how much data can be ca Where the processor uses a 64-bit data bus, it obviously carries mo processor can work faster. Where word size and data bus width are always be carried out in one single process, which improves perfor

7.3 – Progress Check

9. Describe how clock speed, number of cores and word length impact proc

7.4 EXTERNAL HARDWARE DEVICES

Input and output devices	
Device	**Description**
Barcode reader	A bar code represents a code number for a product and is a set of vertical lines of different thicknesses that represent a number. The lines may give information about: • Country of manufacture • Name of manufacturer • Product code Barcode readers or scanners are used to input data from bar codes: • It scans or 'reads' the barcode by using a visible red light • A light sensor measures the reflected light, where white are black areas so creating an analogue waveform of the bar co • The analogue waveform is converted into digital data that i product data • A check digit on the end of the bar code is used to ensure b
Digital camera	Pictures taken using a digital camera are stored on a memory card a • Connecting the camera to the computer via a USB port or • By plugging the camera memory card into the computer car Digital cameras operate as follows: • When a picture is taken the image strikes the camera senso • The sensor records the amount of (RGB) light received and c • The image sensor records in RGB format so that all other co • The picture is then stored as an image file on a memory car contains a series of digital data for each pixel in the picture • Image files can be stored in RAW uncompressed format that in compressed format such as JPG or TIFF Image editing software can be used to modify images that have bee from the digital camera into the computer; typical improvements in • Red eye removal • Cropping or removing unwanted parts of the image • Modifying the brightness/contrast • Changing the image resolution
Laser printer	Laser printers are used in many workplaces and function as follows: • They operate on a page at a time and files that require prin are sent to the laser printer using a page description langua • The printer processes this file and creates a bitmap pattern for the page • The printer laser beam scans along and across the drum insi the printer creating the same pattern as the page that requi printing; this builds up a pattern of static electricity which attracts toner (powdered ink) onto the page • The toner is then fused onto paper by heat and pressure, us The main advantages of laser printers are their print speed; commer pages per minute and also there is high-quality dry page output.

RFID reader	Radio frequency identification tags (RFID), are fixed to products and be programmed with information, similar to a barcode. The RFID reader operates as follows: • RFID tags have an antenna (normally a flat spiral aerial) tha communicate with an RFID reader for tracking purposes • The RFID reader transmits a radio signal which is received b and a programmed information code is returned to the read • This code is then decoded and translated into digital data tl identify the information received. RFID tags can be used for security purposes in stores to prevent thef off an alarm when it is removed from the store. **Advantages:** RFID tags do not need a power source as the signal from the 'reader' energises the passive tag and causes it to transmit its ID code. (Second column, cut off:) The RFID not have

Secondary storage devices

Secondary storage devices are used in computer systems to avoid having to re-enter prog non-volatile or permanent copy. Additionally, data can be stored more than once so there from the secondary storage device.

Device	Description
Hard disk	The most common form of secondary storage device is the interr disk drive, where: • Drive consists of a number of metal disks which have been coated with a special magnetic material • The disks are mounted on a common spindle and rotate at high speed • A series of read/write heads move across the disk surface together to access or store the data on the disk • The hard disk mounted in a sealed unit and is connected power supply inside the computer Since the data is stored in this magnetic material the device is k Some hard disk drives are not permanently fixed inside the com the computer; other hard drives are even more portable as they computer via a USB cable. **Advantages:** Hard disks have a high data transfer rate. High storage capacity (typically 500 gigabytes to 10 terabytes) (Second column, cut off:) Where har can be diff computer. Care has t due to the

Optical disk	Optical disks are known as optical storage devices, where: • The optical disk is a plastic disk with a mirrored surface • Binary data is burned or pressed onto the surface of the disk as series of 'pits' and 'lands' • Whichever method is used to create the disk, the 'pit' h binary value of zero, whereas the 'lands' have a binary • Laser beams are used to read the data stored on the di different amounts of light, which can then be decoded There are different types of optical disk available as follows:

Type of Disk	Storage Capacity
CD	650 Mb
DVD	8.4 Gb
Blu-ray	25 Gb

Blu-ray disks can be p increasing their capac

Advantages	
Portable device that is cheap to produce – typical blank disks cost between 10p–50p each.	Care need the shiny
Reasonable storage capacity for home and faster access time than magnetic tape.	Less stora is unsuital
Optical drives are compatible with CD and DVD disks so data can be read from either media.	Access tin SSD devic

Solid-state disk (SSD)	Solid-state disks act in a similar way to a hard drive except are flash memory secondary storage devices where: • They are based on non-volatile NAND flash memo with a controller to manage pages, blocks of data the complexities of writing • The flash memory cells are grouped into a grid tha separated into sections called 'pages', into which t • A block consists of many pages and pages cannot b block has to be erased before the page can be ove

SSD compared to hard di	
Speed	SSDs perform faster than hard drives as there are n consequently they can read, write and access data f Latency is a feature of hard disks as they are slow t position read/write heads over the disk to access da
Cost	SSDs are significantly more expensive to purchase t to be smaller capacity than hard disk drives.

7.4 – Progress Check

10. Describe the operation and uses of an RFID reader (6 marks)
11. Compare the relative advantages of using hard disks or solid-state disks f storage (4 marks)

TOPIC 8 – CONSEQUENCES OF USES OF

8.1 INDIVIDUAL, SOCIAL, LEGAL AND CULTURAL ISSUES A

Individual and social issues

Morals are a set of principles that apply to an individual, but moral standards might be dif
for individuals based on what they consider to be 'right' or 'wrong'.

There are many definitions of ethical standards; for example, 'it is a set of principles to pr
what is good for individuals and society'. In computing, ethical standards are the behavio
expected by the organisations to which the individual belongs.

Ideally individual moral codes would be similar to the ethics practised by computer profe
However, individuals are not well regulated and there is a tendency to act outside of the l
legislation has been introduced to try to prevent this behaviour.

The information below is a shortened version of the widely used **Software Engineering C**
created by ACM (Association for Computer Machinery) and the IEEE (Institute of Electrical

> ***Software Engineering Code of Ethics and Professional Prac***
>
> ***PREAMBLE***
> *Software engineers shall commit themselves to making the analysis, spe*
> *testing and maintenance of software a beneficial and respected professio*
> *commitment to the health, safety and welfare of the public, software eng*
> *following Eight Principles:*
>
> 1. *PUBLIC - Software engineers shall act consistently with the public in*
> 2. *CLIENT AND EMPLOYER - Software engineers shall act in a manner client and employer consistent with the public interest.*
> 3. *PRODUCT - Software engineers shall ensure that their products and highest professional standards possible.*
> 4. *JUDGMENT - Software engineers shall maintain integrity and indepe judgment.*
> 5. *MANAGEMENT - Software engineering managers and leaders shall s approach to the management of software development and mainte*
> 6. *PROFESSION - Software engineers shall advance the integrity and r consistent with the public interest.*
> 7. *COLLEAGUES - Software engineers shall be fair to and supportive o*
> 8. *SELF - Software engineers shall participate in lifelong learning rega and shall promote an ethical approach to the practice of the profess*

See **www.acm.o**

There are many ethnical responsibilities for software engineers and computer scientists t
their disposal is not misused; typical ethical issues to be aware of are:

- To develop software that respects the privacy of those users that will be affected
- To only use accurate data derived by ethical and lawful means.
- To maintain the integrity of data.
- To not knowingly use software that has been obtained illegally.

Legal issues

The following legislation is important when creating computer systems:

- Data Protection Act
- Copyright Law
- Computer Misuse Act

Data Protection Act (1998)

The Data Protection Act was introduced to ensure that the data held on clients conformed
held securely, being up to date and not kept longer than necessary.

Keeping data secure is a key feature of this legislation and so when developing computer
add a range of software methods to help protect client data, such as:

- **Passwords** entered in to software applications should only be accepted if they a
strong passwords are created by using mixtures of numbers, letters, and symbol
that the final password does not look like a word.
- **Encryption** is used to make stored data more secure, by making it unreadable to
decode it. This method is commonly used to protect data transmitted over the In
- **Selective drop-down menus** are sometimes used as a security method to add let
typing them; this prevents key-logging software from viewing the systems and g

Personal data moral issues include:

- Misuse of personal data – selling it to businesses that use the data for a different purpose to the original reason it was collected.
- Data security – personal data is stored on networks that can be broken into and stolen by hackers. In some cases government employees have lost laptop computers with unencrypted personal details of taxpayers which have got into the wrong hands, causing privacy issues.

Copyright Design and Patents Act (1988)

The Copyright Design and Patents Act introduced to protect the intellectual property of in
and produce materials based on their own individual ideas.

The computing industry has grown tremendously in recent years with a great many new co

Copyright legislation is useful in protecting the following aspects of computer technology:

- **Piracy** is the illegal copying of software for either personal use or business use; t
can include illegal downloading of games and music as well as commercial softv
- **Theft** of hardware and software ideas and innovations. In an industry that moves
time to patent your invention before you release it on to the open market. Many
their rivals' products with the express purpose of copying their ideas, which save
expense on research and development.

Protecting your copyright is especially worthwhile when you or your organisation have inv
a new hardware or software concept.

There are many websites that can help the creator to protect and patent their work. The m
that they make the public aware of their intellectual property rights; this can be achieved
with their products, stating that their designs are copyrighted, can't be modified or copied
'reverse engineering' techniques can be used to produce replicas.

Copyright issues include:

- Unauthorised use of software – downloading or copying software illegally witho
purchasing the necessary software licence
- Illegal copying of music and films
- Illegal copying of web content

Computer Misuse Act (1990)

The Computer Misuse Act was introduced with the express purpose of preventing attacks on ICT systems to commit crimes or to damage the system; this legislation made hacking and the introduction of a computer virus into criminal offences.

Hacking is the practice of breaking into computer systems and it is essential that preventative measures are taken. The main techniques are to utilise a **firewall** and to set up some intrusion detection.

ICT systems are constantly communicating with the outside world, which involves connec associated difficulty of effectively policing access to the system. A firewall is a combinatic designed to check the integrity of incoming messages and requests for service from the s

Intrusion detection systems (IDS) are designed to monitor the network or computer syster is detected a report is produced which is sent to the network management for further actio

A computer **virus** is a program designed to cause damage to a computer system. The use c helps to minimise the risk from viruses; this software searches the computer system for vi

Spyware can be loaded into a computer system as a software virus, so it is important to ru prevent and detect spyware from being installed and to remove any spyware that has pre

ⓘ A **firewall** is used to prevent unauthorised access to the network or computer systems via the Internet.	ⓘ **Spyware** refers to programs that run in a computer system to gather information and pass it on to other interested parties.	ⓘ **Hackir** practic into se systerr

Computer misuse issues include:

- Unauthorised access by hackers for fraud purposes or to gain access just for the

Regulation of Investigatory Powers Act (2000)

Regulation of Investigatory Powers Act was introduced to explain the powers that public authorities, such as the intelligence services, the police and Customs and Excise have available when investigating crime or terrorism.

The mains parts of this legislation are that the authorities can obtain information from:

- ISPs providing access to customer communications
- Telephone tapping
- Monitoring individuals' Internet activities
- Accessing protected information, if encrypted the authorities have the right to de means the encryption key must be handed over.

Regulation of Investigatory Powers (RIP) Act issues:

- There is concern that the RIPA powers might be abused by the authorities and no or terrorists.

Cultural and communication issues

In recent years there have been many developments in computer science and digital technology which have had a major impact on information flow and communication throughout the world.

This has led to the ability to analyse, monitor and distribute personal information on a large scale. Although in some cases the consequences of this improved communication can be beneficial to society, there is a risk that it can also be harmful.

Students are expected to have an understanding of the consequences and uses of computin

- One method to gain this knowledge is by making use of current technology news fr
- Reading case studies will develop an understanding of the issues involved
- If the article does not give a balanced view of the issues, it might be useful to consi viewpoint

The following Internet news articles provide an insight into some of the modern issues rega

'Twitter sues US Government over spying' from BBC Technology News (7th Octob
http://www.bbc.co.uk/news/technology-29528665

'Bionic drone bird aims to take flight' video from BBC Technology News (26th Jan
http://www.bbc.co.uk/news/technology-30955444

'Facebook U-turns on phone and address data sharing' from BBC Technology Nev
http://www.bbc.co.uk/news/technology-12214628

'Obama makes push for stronger cyber security laws' from BBC World News (14th
http://www.bbc.co.uk/news/world-us-canada-30807463

'Samsung's smart TVs fail to encrypt voice commands' from BBC Technology Nev
http://www.bbc.co.uk/news/technology-31523497

'Seven million use illegal files' from BBC Technology News (28th May 2009)
http://news.bbc.co.uk/1/hi/technology/8073068.stm

8.1 – Progress Check

1. Compare the terms 'morals' and 'ethics' (4 marks)
2. Briefly discuss some of the ethical responsibilities faced by software eng
3. The Computer Misuse Act was introduced with the express purpose of p commit crimes or to damage the system; describe the following related
   (a) Hacking (2 marks)
   (b) Viruses (2 marks)
   (c) Firewall (2 marks)
   (d) Spyware (2 marks)

Research Tasks

Use the Internet to research the following:

1. Countries that have censorship on Internet activity
2. Cases where the authorities have abused their powers of the RIP Act
3. Situations where a public sector employee has lost a laptop and the sensitive personal

TOPIC 9 – COMMUNICATION AND NE

9.1 COMMUNICATION

Communication methods

Data transmission refers to the transfer of data between two or more digital devices.

ⓘ **Serial** data transmission is where data is transmitted one bit at a time along a single

ⓘ **Parallel** data transmission is where data is transmitted more than one bit at a time al

Serial vs Parallel	
Speed	If a parallel data bus had eight cables it would, in theory, transmit data e cable used in a serial bus.
Interference	There is interference between the parallel wires which reduces the band transmission bit rate is limited by interference.
Connections	Connecting cables are short to minimise the interference effect with par connection there is no interference or noise problem so connecting cabl
Cables	Parallel data transmission uses more cables than serial data transmissio expensive to produce and more complex than serial cables.

ⓘ **Synchronous** data transmission is where a block of data is transmitted along with a c this will ensure that the data transmitter and receiver are synchronised.

Most network protocols make use of synchronous transmission, where data is sent in pack

ⓘ **Asynchronous** data transmission is where a data stream is not synchronised and transmitted at fixed time intervals.

It is only synchronised for the transmission period by making use of start and stop bits; this method is normally used for communications over telephone lines.

Data transmitted asynchronously includes a start bit inserted for each unit of data and a stop bit to indicate the termination of the data transmission.

ⓘ St
u
tr

ⓘ St
u
tr

Communication basics

ⓘ **Baud rate** is the number of symbols (or symbol changes) that are transferred across a

Baud rate is based on the number of bits encoded in each signal change; so, for example, signal change the bit rate will be eight times higher than the baud rate.

ⓘ **Bit rate** is the number of bits transferred across network in a set amount of time; nor

ⓘ **Bandwidth** is the speed of a network or the maximum rate at which data is transferre and indicates the maximum amount of data that can pass from one point to another i proportional to bandwidth.

ⓘ **Latency** is the term used in a packet switched network to measure the time delay bet and the destination receiving that packet.

Ping is a network utility used to measure latency; it works by calculating the time taken f its source and destination.

ⓘ A **protocol is** a set of rules or standards which computers use for communication.

An example of a protocol is **TCP/IP** which provides a unique identifier for a device or com transmission control protocol (**TCP**) sets the standard for the delivery of information pack

9.1 – Progress Check

1. Compare and contrast 'synchronous' and 'asynchronous' data transmiss
2. Describe the following communication terms:
   - (a) Bit rate (2 marks)
   - (b) Bandwidth (2 marks)
   - (c) Latency (2 marks)
   - (d) Protocol (2 marks)

9.2 NETWORKING

Network topology

ⓘ A network **topology** is a way of describing the interconnections and cabling of a grou

ⓘ **Physical topology** is how the devices or a group of computers are physically connecte

ⓘ **Logical topology** is how the network devices communicate with each other.

Logical classification of network topologies has the same characteristics as their physical paths for a logical bus network follow the same route as those in a physical bus network.

Physical Star Topology

ⓘ The physical **star topology** is based on connecting each workstation to a single point such as a hub which may also be linked to a file server.

If one network cable breaks, the workstation connected to it also fails, but it does not affe the rest of the network.

However, if the central hub fails the whole network will be inoperative as each workstatic relies on the central hub for communication.

Advantages	
If one of the network cables breaks only the computer connected to that cable is affected. Fast response time and no problems with data collision as each workstation has its own cable. Security can be installed at the central hub.	All communication take the whole network will Uses the most cable wh install. Also long cable degrade.

Physical Bus Topology

ⓘ When a physical **bus topology** is used, each workstation is connected to a single cabl (often referred to as the 'backbone') which links all of the workstations.

The file server is connected to the main bus for data distribution to all the workstations.

Data can be transmitted in either direction along the main cable, and workstations can communicate with their peers.

A range of peripherals can also be connected to the main bus for shared usage.

The physically wired star topology can be dynamically reconfigured into a logical bus topc protocols and specialist equipment such as routers and switches.

Advantages	
If one workstation fails it does not affect the rest of the network. Is cheaper to install as it uses the least cable as the cost of network cabling (particularly fibre optic) and the cost of network cable installation can be significant.	If the main network cab workstations can acces If there is heavy networ fall off dramatically. If the main bus fails all

9.2 – Progress Check

3. (a) Define the term 'network topology' (2 marks)
   (b) Explain the difference between physical and logical topology (2 mark
4. (a) Explain, with a sketch, the physical star topology (2 marks)
   (b) Describe one advantage of the physical star topology (2 marks)
   (c) Describe one disadvantage of the physical star topology (2 marks)

Client-server and peer-to-peer networking

Client-server networking

ⓘ A **client-server** model is where the workstations in any network design use the server to provide a service.

ⓘ **Client** is a computer or workstation on a typical network.

ⓘ **File server** is the computer with the main processing power and storage that interacts with the clients.

The client-server model has become one of the central ideas of network computing. Clients are the many workstations on the network and the file server is the more powerful system that interacts with the clients.

The client-server system diagram shows a range of client devices connected via the Intern

In a school context: a student logs on to a workstation and opens a word processor applic document for editing (requested from the server) and the document is sent to them from t changes. Whenever the document is saved the changes will be made to the file server loc

Advantages	Disa
The system facilitates a centralised backup.	Server failure can cause the whole
Centralised data which is more consistent as there is only one copy in use. Any lost data can be restored centrally for all users.	Network management is a skilled maintenance.
Security policies for access control and associated usernames are managed centrally.	It is expensive and time-consumin network operating system, which i clients and the labour involved in

Peer-to-peer networking

Peer-to-peer networks are different from client-server networks as they do not have special workstations which are used as servers. Instead any workstation on the network can load information from the hard disk or use a local printer for any other workstation, as all workstations have equal status.

In the diagram it should be noted that all workstations are linked to each other; complexity of cabling can be reduced nowadays by use of Wi-Fi.

Peer-to-peer networks are commonly used as local area networks (LANs) for general administration by either small businesses or in the home.

Note that in peer-to-peer networks all workstations need to be linked but it does not need to be a direct link.

9.2 – Progress Check

5. Explain, with a sketch, peer-to-peer networking (2 marks)
6. (a) Describe the client-server network model (2 marks)
   (b) Describe one advantage of the client-server network model (1 mark
   (c) Describe one disadvantage of the client-server network model (1 m

Wireless networking

ⓘ **Wi-Fi** allows devices, such as smartphones, tablets and laptops to communicate wirel

ⓘ A **wireless access point** is device that allows wireless devices to connect to a wired n

A **wireless adaptor** performs a similar function to a **network interface card (NIC)** but wirelessly. The wireless adaptor can be built into the computer or it can be a portable dev fitted into a USB port.

In a home or office network a wireless router transmits a Wi-Fi signal that can be shared b more than one device. This approach is used to share an Internet connection between sev devices using a mixture of Wi-Fi and Ethernet cable connectivity.

ⓘ **WPA/WPA2** (Wi-Fi Protected Access) are encryption protocols designed to protect a W

This protocol is used whenever an attempt is made to connect to a secured wireless netw key or passcode. This ensures that the connection is secured with encryption.

In some cases the wireless access point has option to choose the security protocol; where is the latest and most secure encryption version.

ⓘ **SSID** (service set identification) is a unique 32-character identifier that is used to nam

It is necessary to ensure that a device is connected to the correct WLAN (wireless local area network); the normal approach to checking for this connectivity is to put a copy of the SSID into the header of each packet of data being sent.

This SSID code is unique to a specific WLAN and only devices that are programmed with t

ⓘ **A MAC (Media Access Control)** address is the physical address of a networking device identifier assigned by the manufacturer and is stored in the device's read-only memo

Security can be improved on a wireless network as follows: the router for a wireless netw to a white list of computers – these computers are identified by their unique MAC address addresses of all the devices that are approved, accepted or recognised by the system.

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

ⓘ **Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)** operates by the c that it is about to transmit data. Other computers can then avoid a collision by delayi

CSMA/CA access methods can be used both with and without the Request to Send / Clear

With RTS/CTS	W
1. RTS signal is sent by the sender of the package 2. then a CTS signal is sent by the intended receiver of the package 3. consequently the sender and receiver are aware of a data transmission and hold off for the duration of the main transmission	1. The computer war the channel is 'idl 2. then it is able to s 3. if the channel is se and try again

The CSMA/CA method is widely used in WLANs although it has the disadvantage of reduci increases the amount of network traffic.

9.2 – Progress Check

7. Describe the following networking terms:
   - (a) Wireless adaptor (2 marks)
   - (b) Wi-Fi protected access (2 marks)
   - (c) Service set identification (S
   - (d) MAC address (2 marks)

ANSWERS

Topic 5 – Data representation

5.1 N is the set of natural numbers (often termed as whole numbers) used for counting
is the set of integer numbers that includes the natural numbers and their negative i
have no fractional part (1).

5.2 Convert binary 1110 0111 to decimal (2 marks)

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

So 128 + 64 + 32 + 4 + 2 + 1 = **231_{10}**

5.3 Convert decimal 101 to binary (2 marks)

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

5.4 Convert hexadecimal 3FA to decimal (2 marks)
First convert to binary, so $3FA_{16} = 001111111010_2$

3				F			
8	4	2	1	8	4	2	1
0	0	1	1	1	1	1	1

Then use the table to convert to decimal

512	256	128	64	32	16	8	4
1	1	1	1	1	1	1	0

So 512 + 256 + 128 + 64 + 32 + 16 + 8 + 4 + 1 = **1018_{10}**

5.5 Convert decimal 7012 to hexadecimal (2 marks)
(1) 7012 decimal is converted to binary; therefore, $1101101100100_2 = 7012_{10}$

4096	2048	1024	512	256	128	64	32	16	8
1	1	0	1	1	0	1	1	0	0

(2) Split the binary number into nibbles and convert into hex to give **$1B64_{16}$**

5.6 Convert binary 0011 1111 to hexadecimal (2 marks)

- Treat the 8-bit binary code as two hexadecimal nibbles: 0011 and 1111.
- Which converts to 3 and B
  So, **$00111111_2 = 3B_{16}$**

5.7 Convert hexadecimal B7 to binary (2 marks)

- Convert the two hexadecimal nibbles separately using the table, so B_{16} = 1011
- Join them together to form an 8-bit binary number.
  So, **$B7_{16} = 10110111_2$**

5.8 A kibibyte (KiB) is a unit used for digital information (1), where 1 kibibyte = 1024 by

5.9 Add the following unsigned binary numbers 00000101 and 11000100 (2 marks)

	0	0	0	0	0	1	0	1
	1	1	0	0	0	1	0	0
Sum	1	1	0	0	1	0	0	1
Carried	0	0	0	0	1	0	0	

So 00000101 + 11000100 = **11001001 binary**

5.10 Multiply the following unsigned binary numbers 00010101 and 00000101 (2 marks)

			0	0	0	1	0	1	0	1
			0	0	0	0	0	1	0	1
			0	0	0	1	0	1	0	1
	0	0	0	1	0	1	0	1		
			0	1	1	0	1	0	0	1
Carry					1		1			

00010101 x 00000101 = 01101001 Binary

5.11 Subtract 24 – 12 using two's complement (2 marks)
24_{10} = 0001 1000_2 and
12_{10} = 0000 1100_2 which is (1111 0011) + 1 = **1111 0100_2** in two's complement form

	0	0	0	1	1	0	0	0	24
	1	1	1	1	0	1	0	0	-12
Sum	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	24 – 12 = 12
Carry	1	1	1	0	0	-	-	-	So **0001 1000_2 - 0000 11**

5.12 Convert fixed-point binary number **1001111100.1010** into decimal (3 marks)
First write the binary numbers into a table in the correct order

512	256	128	64	32	16	8	4	2	1	.	1/2	1/
1	0	0	1	1	1	1	1	0	0	.	1	0

Then add the decimal numbers where 1 is shown;
1010000111. 10010_2 = 512 + 64 + 32 + 16 + 8 + 4 + 0.5 + 0.12
Alternatively 512 + 64 + 32 + 16 + 8 + 4 + $\frac{1}{2}$ + $\frac{1}{8}$ = 636

5.13 ASCII is a seven-bit character set which offers 128 different characters (1) and many controlling peripherals (1); whereas Unicode is a 16-bit character set which offers 6 includes the character for any writing system in the world (1).

5.14 Even parity is where the numbers of 1's in the data to be transmitted is counted an parity bit is set (1) to 1. In the example below the 7-Bit ASCII code for Q is 1010001 parity bit is set (1) to 1.

Parity Bit	ASCII Code for Q				
1	1	0	1	0	0

5.15 Analogue to digital converters (ADC) converts an analogue signal that varies contin
signal.

Analogue signals are input into the device in the form of voltage waveforms (1) tha
format before they can be read and understood by the computer (1).

Digital to analogue converters (DAC) convert a discrete digital signal (1) into a conti
is outputted from a computer (1). For example, sound outputs from a computer ne
signals for use by speakers.

5.16 Digital image terms:

(a) Colour depth is the number of bits used for each pixel (1), where 1 bit is black
colours (1).

(b) Resolution is the number of pixels per inch (1) – often termed dpi or dots per
image resolution (1).

(c) Metadata is included in an image file (1) and gives details of image height, wic

5.17 Calculate the memory used in kilobytes for a digital photographic image that is 8,5C
with colour depth 24 bpp (bits per pixel) where 1 byte = 8 bits (2).

Total pixels	= 8500 * 4020 = 34,170,000
File size (bytes)	= total pixels * (colour depth / bits per byte)
	= 34,170,000 * (24/8) = 102,510,000
File size (MB)	= 102,510,000 / (1,024 * 1,024) = 97.76 MB

5.18 Digital sound terms:

(a) Sampling rate is the number of samples taken per second from the analogue in

(b) Sampling resolution (or audio bit depth) is the number of bits used to store ea

5.19 Lossless compression is where a file can be compressed (1) but the facility exists to
exact original format (1). Lossy compression is where a file is compressed by remov
size (1), so it is not possible to create an exact copy of the original file (1).

5.20 Encryption is used to make stored data more secure from hackers by making it unre
have the key to decrypt or decode it (1). Cryptography is the process of creating cip
(known as encryption) and turning it back to the original plaintext (known as decry

5.21 The Caesar cipher is a commonly used and simple substitution cipher; using this tec
letters in a message is replaced (1) by a different letter of the alphabet, a fixed posi

The ciphertext: CU EQORWVGT UEKGPEG that has been encrypted with a Caesar cy
the decryption key of a right shift by two places in the alphabet decrypts the messa

Topic 6 – Computer systems

6.1 Software is the actual programs or coded instructions (1) that make the computer r
physical parts (1) that make up a computer system (1).

6.2 Programming languages:

(a) Machine code is the set of binary instructions (1) that are used by the CPU to

(b) Assembly language is a low-level computer programming language (1) which i
instruction is one machine operation (1).

(c) High-level language is a computer programming language (1) based on natura
notation (1).

6.3 The advantages of machine code and assembly languages compared to high level la

- Programming in low-level code can create faster and more efficient code as th
performance level of the created code (1); with high-level programming there
create optimised code (1).
- Low-level language code is memory efficient due to the lack of abstraction in t
level languages (1).

The disadvantages of machine code and assembly languages compared to high leve

- It is difficult to learn to program in low-level languages whereas high-level lan training options (1).
- High-level code can be self-documenting which makes it is more understandal and debugging (1)
- Machine-code and assembly programs are specific to a limited range of proce be compiled to run on a wide range of processors (1).

6.4 Source code is the language instructions that have been written by the computer p run or execute the source code directly (1).

Object Code (or executable code) is translated from the source code (1) using an as instructions that can be understood and executed by the computer (1).

6.5 The assembler translates this source code into machine code that the computer ca

The advantage of using assembly language is it is an efficient low-level language (1) has a one-to-one relationship with machine code (1).

The main disadvantage of using an assembly programming language is that it is con programming time and expertise (1).

6.6 A software application called a compiler converts the source code to object code w

The main advantage of using a compiler on high-level language code is that an exec without the need of the source code (1).

The disadvantages of this method are that the compilation of a large program take errors in the source code need to be corrected before an executable file can be pro

6.7 Interpreter software normally executes the source code directly; this avoids the ne

The advantage of using an interpreter is that during development the programmer which can be tested without going through the time-consuming process of compilir

The disadvantage of using an interpreter is that it needs to be loaded on the target efficient machine code at runtime (1). Additionally, the source code is available to t translation method takes longer than a compiled program to run (1).

6.8 Use the complex logic diagram to work out the outputs for various values of A, B ar

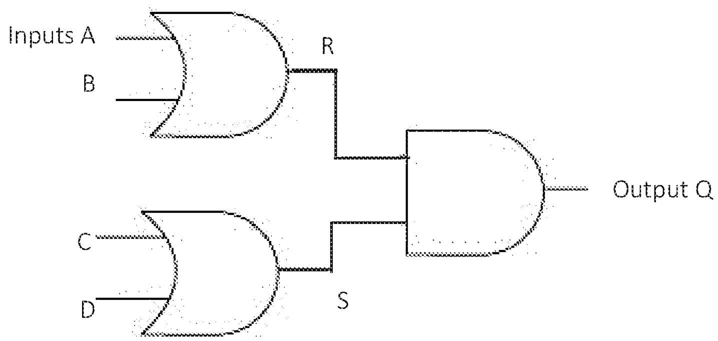
(a) When A = 0 B=1 C = 0 (3 marks)
then R = 1, S = 0, T= 0

(b) When A = 0 B=0 C = 1 (3 marks)
then R = 0, S = 1, T= 1

(c) When A = 1 B=1 C = 0 (3 marks)
then R = 1, S = 0, T= 0

(d) When A = 1 B=1 C = 1 (3 marks)
then R = 1, S = 0, T= 0

6.9 Logic circuit for the Q = (A OR B) AND (C OR D) (3 marks)



6.10 Simplify $\overline{A}$.B.C + $\overline{A}$.C (3 marks)

$\overline{A}$.C (B + 1)	Take common terms outside of brackets
(B + 1) = 1	So B can be removed
$\overline{A}$.C (1)	Remove bracketed 1
$\overline{A}$.C	Final Answer

Topic 7 – Computer organisation and architecture

7.1 Computer architecture terms:

(a) The central processing unit (CPU) processes the data in a computer system (1); t sorting and searching data, performing calculations, logical decision-making and

(b) Main memory or immediate access store (1) is storage for program instructior CPU (1).

(c) The address bus is used to specify a physical address in main memory (1) and into that address is communicated via the data bus (1).

(d) The control bus manages data processing (1) so, for example: by sending a sig to a memory address or a signal to read data from a memory address (1).

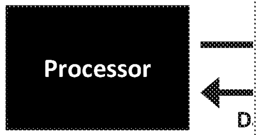
(e) I/O controllers send control signals to connect a system bus (1) to specific I/O output (write) requests they receive from the processor (1).

(f) Arithmetic logic unit – performs arithmetic and logical operations, such as fixe logical operations (AND, OR, XOR) and shift operations (1). Inputs are from the with outputs routed to one of the registers (1).

7.2 Van Neumann program architecture

The Van Neumann stored program architecture is based on both data and instructions stored in a single memory (1).

Data and instructions are transmitted through a shared data bus (1). Sketch (2).



7.3 A register is a very fast memory location that exists in the processor or the I/O cont General purpose registers are available for the programmer to store temporary dat Dedicated registers are used by the processor to carry out a specific role (1).

7.4 Dedicated registers:

(a) Status register (SR) is used to hold the status of various flags indicating such th result, carry bit used, overflow error, and interrupt status. The details of the s upon the actual processor involved (2).

(b) Program counter (PC) is the register that holds the address of the next instruc

(c) Memory address register (MAR) holds the address of the current instruction b

(d) Memory buffer register (MBR) holds the instruction from the (MAR) and the a

(e) Current instruction register (IR) is used to store the instruction that is to be de

7.5 Fetch-execute cycle

Fetch:

1. Program counter (PC) points to next instruction to be fetched (1).
2. Contents of (PC) copied into memory address register (MAR) (1).
3. The instruction in (MAR) is transferred via the data bus to the memory buffer register (MBR) (1).
4. The contents of the (MBR) are copied to instruction register (CIR) and the program counter (PC) is updated to address next instruction (1).

Decode:

The instruction in the (IR) is decoded (1).

Execute:

1. The decoded instruction is executed (1).
2. The process is repeated.

7.6 Processor instruction set terms:

(a) Machine code instructions are written in binary code (1) that a processor can

(b) Op-code instructions are the part of a machine code instruction (1) that repre

(c) Operand is that part of the machine code instruction (1) that contains an item which the binary data is stored (1).

(d) Instruction set is the set of machine code instructions (1) that the processor h execute (1).

7.7 Immediate addressing means that the data in the operand is fixed (1); in other wor

This is a very fast addressing mode since the data is readily available (1), rather tha from a memory address, whereas direct addressing, or absolute addressing, means a memory location (1).

A typical example is the instruction ADD (1302) – this instruction could be used to a (1302) to the accumulator (1).

7.8

```
         CMP   r1, r2       ; compares value of A with B
         BGT   greater      ; branch if A > B
         BEQ   equal        ; branch if A = B
         MOV   r3, #0       ; set P = 0
         B     end          ; branch to end
greater  MOV   r3, #3       ; set P = 3
         B     end          ; branch to end
equal    MOV   r3, #1       ; set P = 1
end      HALT               ; end of code
```

7.9 The clock speed is a key factor in the performance of a computer; it controls the pr well as fetching data in another cycle (1). Clock speed has increased regularly in rec run at a clock speed in excess of 2.5 GHz, or 2.5 thousand million times a second. C increased by purchasing a faster device (1). However, the faster the device the mor cooling is necessary to prevent CPU damage.

System performance can be improved by increasing the number of processors; this c processor (1) A multi-core processor has more than one processor incorporated into core processor will provide two processors on one chip, which will operate faster tha

Most modern processors use 32-bit or 64-bit word lengths; the performance of a p word length, providing that the word length and data bus are the same size (1). So, bit bus will allow 64 bits of data to be handled in one clock pulse (1).

7.10 RFID readers communicate with RFID tags, or radio frequency identification tags; th as a bar code; the tag will be programmed with information (1). Each tag has an an that can be used to communicate with a RFID reader or scanner for tracking purpo

The RFID reader transmits a radio signal which is received by the RFID tag and a coc is sent to the reader (1). The code is then decoded and translated into digital data t identify the information received (1). These tags can be used for security purposes on checkouts where they have the advantage of being scanned remotely and are re

7.11 The main secondary storage device on most computers is a hard disk, which consis which have been coated with a special magnetic material, and a series of read/writ surface to access or store the data on the disk (1). Since the data is stored in this m as a magnetic storage device. Typical internal hard disks supplied with modern com range 500 gigabytes to 10 terabytes (1).

Solid-state disks act in a similar way to a hard drive except they are based on non-v controller to manage pages, blocks of data and the complexities of writing. The flas grid that is separated into sections called 'pages', into which the data is stored. A b pages cannot be overwritten individually; the whole block has to be erased before

SSDs perform faster than hard drives and optical drives as there are no moving par write and access data faster although they are significantly more expensive to purc tend to be smaller capacity than hard disk drives (1).

Topic 8 – Consequences of uses of computing

8.1. Morals are a set of principles (1) that apply to an individual (1), whereas ethics is a group to which the individual belongs (1).

8.2. There are many ethnical responsibilities for software engineers and computer scie have at their disposal is not misused. Some of the typical ethical issues to be aware respects the privacy (1) of those users that will be affected by that software (1), to by ethical and lawful means (1), to maintain the integrity of data (1) and to not kno obtained illegally (1).

8.3. The Computer Misuse Act terms

(a) Hacking – the practice of breaking into (1) secure computer systems (1).
(b) Viruses – are programs designed to cause damage (1) to a computer system (1
(c) A firewall is used to prevent unauthorised requests from hackers (1) to gain ac systems via the Internet (1).
(d) Spyware – programs that run in a computer system (1) to gather information a parties (1).

Topic 9 –Communication and networking

9.1 Synchronous data transmission is a data transmission method where the data signa clock-generated (1) timing signal; this will ensure that the data transmitter and rece network protocols make use of synchronous transmission, where data is sent in pa

Asynchronous data transmission is where a data stream is transmitted at intermitte intervals; this method of data transmission is normally adopted for communication

9.2 Communication basic terms:

(a) Bit rate is the number of bits that are transferred across a computer network normally measured in bits per second (bps).
(b) Bandwidth is the speed of a computer network or the maximum rate at which measured in Mbps (Megabits per second) and basically bandwidth indicates tl pass from one point to another in a unit of time (1).
(c) Latency is the term used in a packet switched network to measure the time de a data packet and destination receiving that packet (1).
(d) Protocol simply means a set of rules or standards (1) which computers use for
(e) Baud rate is the number of symbols (or signal changes) that are transferred ac amount of time (1).

9.3 (a) A network topology is a way of describing the interconnections (1) and cablin
(b) Physical topology is how a group of computers are physically connected (1), w network devices communicate with each other (1).

9.4 (a) Physical star topology:
The physical star network topology is based on connecting each workstation t may also be linked to a file server (1).
(b) Fast response time and no problems with data collision (1) as each workstatio
(c) All communication takes place via the central hub (1) and if it fails the whole network will break down (1).

9.5 Peer-to-peer networking:
Peer-to-peer networks are different from client-server networks as they do not hav used as servers (1).

Instead, any workstation on the network can load information from the hard disk o workstation, as all workstations have equal status (1).

9.6 Client-server:

(a) The client-server model is where the workstations in any network design use t where the client is a computer or workstation on a typical network and the fil that interacts with the clients (1).
(b) Security policies for access control and associated usernames are managed ce
(c) Server failure can cause the whole network to be inoperative (1).

9.7 Networking terms:

(a) Wireless adaptor performs a similar function to a network interface card (NIC adaptor can be built into the computer or it can be a portable device fitted int
(b) Wi-Fi protected access uses encryption protocols (1) such as WPA/WPA2 that connection (1).
(c) SSID stands for service set identification; it is a unique 32-character identifier network (1).
(d) MAC stands for media access control and it is a physical address (1) assigned t manufacturer (1).