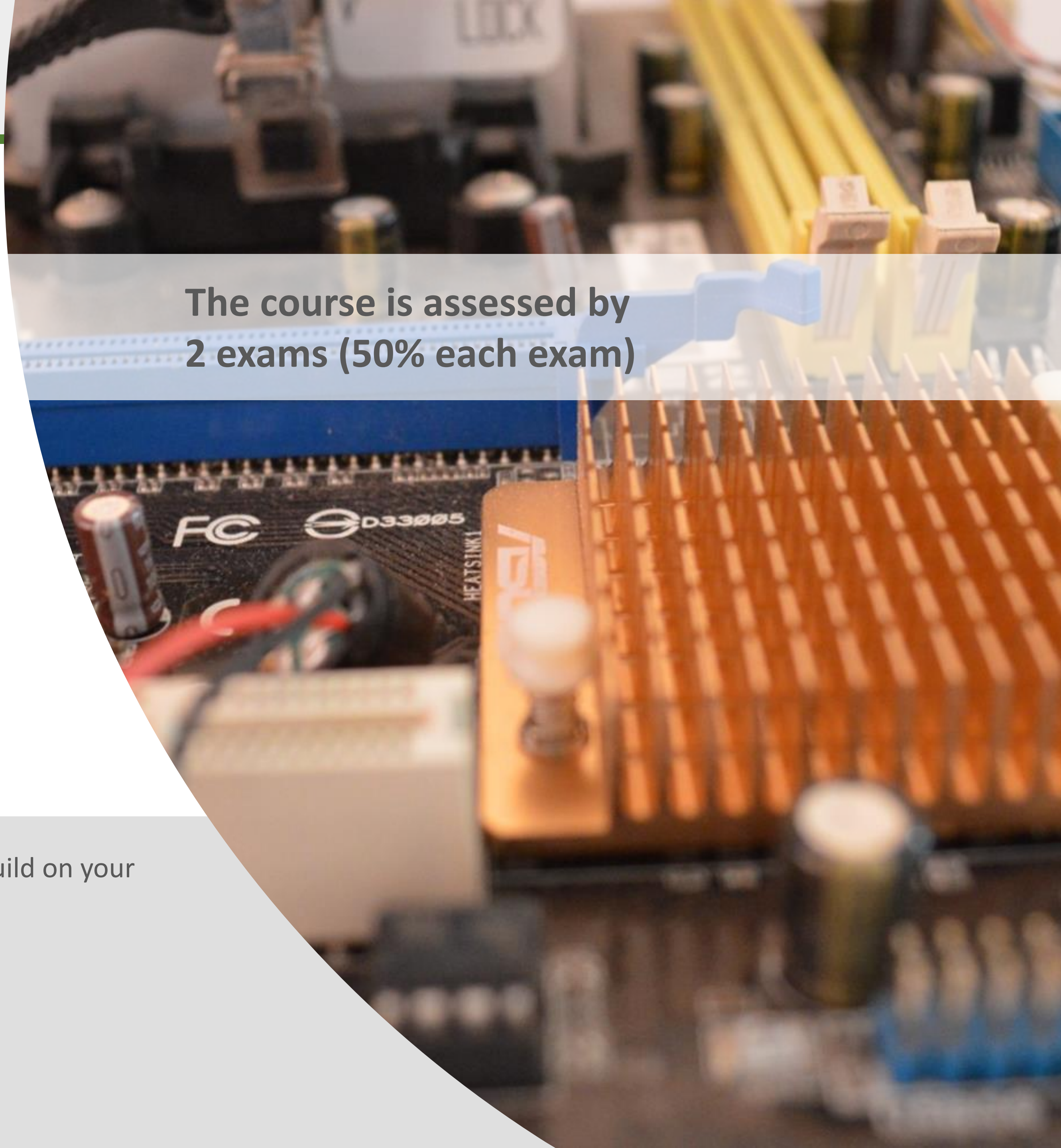
GCSE to A level

Computer Science Transition workbook

- The topic of **Computer Science** is at the heart of the modern world
- Studying it can make you extremely sought after in todays job market
- The transition from GCSE to A level is significant, this includes:
  - An increased emphasis on **technical content**
  - An increased emphasis **independent research**

This workbook is designed to allow you to practice some of these skills and build on your existing knowledge.



The course is assessed by 2 exams (50% each exam)

1 “Tell me about yourself”

Craig’n’Dave

Why did you choose Computer Science?

Expected time to complete: ½ hour

In this simple task you get the opportunity to tell me your choices and reasons behind choosing to study Computer Science. Please answer all questions as best you can.

1. Why did you choose to study A level Computer Science?

2. What other courses have you chosen to study at Key Stage 5, and what made you choose this combination?

3. What are you hoping to achieve from studying Computer Science?

4. How would you describe yourself as a learner at GCSE? What skills where you good at, what areas would you like to improve on?

5. What are your other hobbies and interests outside of school? Anything related to Computing?

2 Independent research task

Craig'n'Dave

Emerging computer technology

Expected time to complete: 2 hours

In this task you get to investigate any area of emerging computer technology which interests you.

You can pick any area which interests you, but examples could be:

- Artificial intelligence
- Robotics
- Automated self driving cards
- Quantum computing

In no more than ONE side of A4 summarise the area you have chosen under the following four headings:

1. What is it?
2. What are the possible Social, Moral, Cultural and Ethical **benefits** of this technology on society
3. What are the possible Social, Moral, Cultural and Ethical **risks** of this technology on society
4. My conclusion on this technology and what it will mean for our world 10 years from now

Additional help:

For additional help and support in structuring your answer you might like to watch some of the videos from the following Craig 'n' Dave playlists:

OCR:
SLR 17 – Ethical, morale and cultural issues
https://student.craigndave.org/videos/slr-17-ethical-moral-and-cultural-issues

AQA:
SLR 19: Moral, social, legal, cultural issues
https://student.craigndave.org/videos/slr19-moral-social-legal-cultural-issues

3 What is “computational thinking”?

Craig’n’Dave

Thinking like a computer

Expected time to complete: 2 hours

At the heart of Computer Science is the ability to look at problems, analyse them, break them down and solve them in a way which involves a variety of “Computational Thinking” skills.

1. Download the “Computational thinking and Computational methods placemats” from Craig n Dave:
    - https://student.craigndave.org/specification-key-terminology-and-cheat-sheets

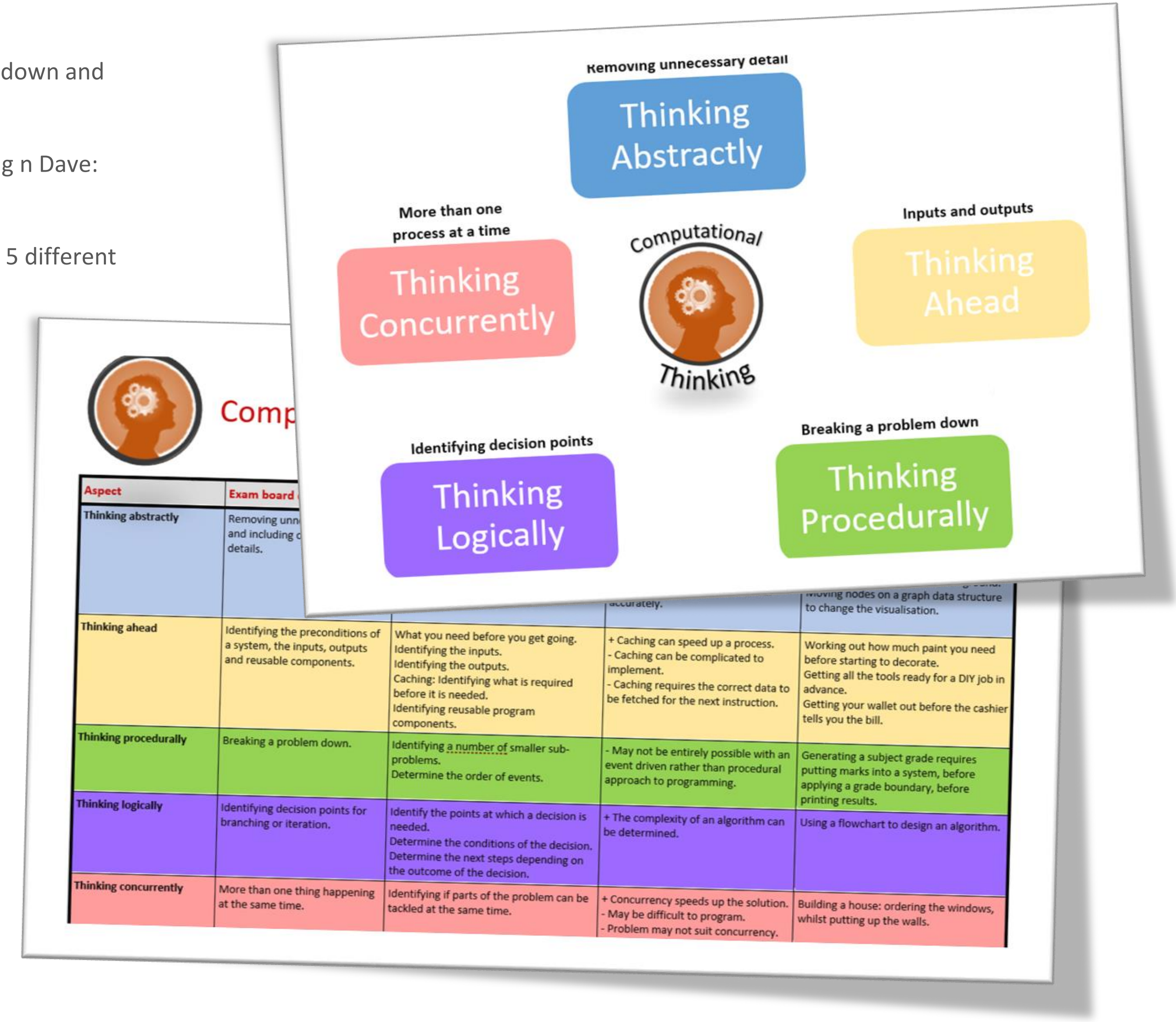
2. Create your own spider diagram / mind map which shows your clear understanding of the 5 different computational thinking strands
    - Keep it to a single side of A4 / A3

3. Your goal is to imagine someone else has to revise from your mind map. Ask yourself:
    - Does it make sense?
    - Is it clear?
    - Does it cover all of the important concepts?

Note:

Although the five strands listed (and the download resources provided for this task) are from the OCR AS / A’Level specification, the concepts of “Computational Thinking” are just as applicable to the AQA course.

Indeed many of the strands listed are explicitly covered in the AQA specification in different locations.



Removing unnecessary detail
Thinking Abstractly

More than one process at a time
Thinking Concurrently

Computational Thinking

Inputs and outputs
Thinking Ahead

Identifying decision points
Thinking Logically

Breaking a problem down
Thinking Procedurally

Comp

Aspect	Exam board			
Thinking abstractly	Removing unn and including c details.		accurately.	Moving nodes on a graph data structure to change the visualisation.
Thinking ahead	Identifying the preconditions of a system, the inputs, outputs and reusable components.	What you need before you get going. Identifying the inputs. Identifying the outputs. Caching: Identifying what is required before it is needed. Identifying reusable program components.	+ Caching can speed up a process. - Caching can be complicated to implement. - Caching requires the correct data to be fetched for the next instruction.	Working out how much paint you need before starting to decorate. Getting all the tools ready for a DIY job in advance. Getting your wallet out before the cashier tells you the bill.
Thinking procedurally	Breaking a problem down.	Identifying a number of smaller sub-problems. Determine the order of events.	- May not be entirely possible with an event driven rather than procedural approach to programming.	Generating a subject grade requires putting marks into a system, before applying a grade boundary, before printing results.
Thinking logically	Identifying decision points for branching or iteration.	Identify the points at which a decision is needed. Determine the conditions of the decision. Determine the next steps depending on the outcome of the decision.	+ The complexity of an algorithm can be determined.	Using a flowchart to design an algorithm.
Thinking concurrently	More than one thing happening at the same time.	Identifying if parts of the problem can be tackled at the same time.	+ Concurrency speeds up the solution. - May be difficult to program. - Problem may not suit concurrency.	Building a house: ordering the windows, whilst putting up the walls.

4 Note taking practice task

Craig'n'Dave

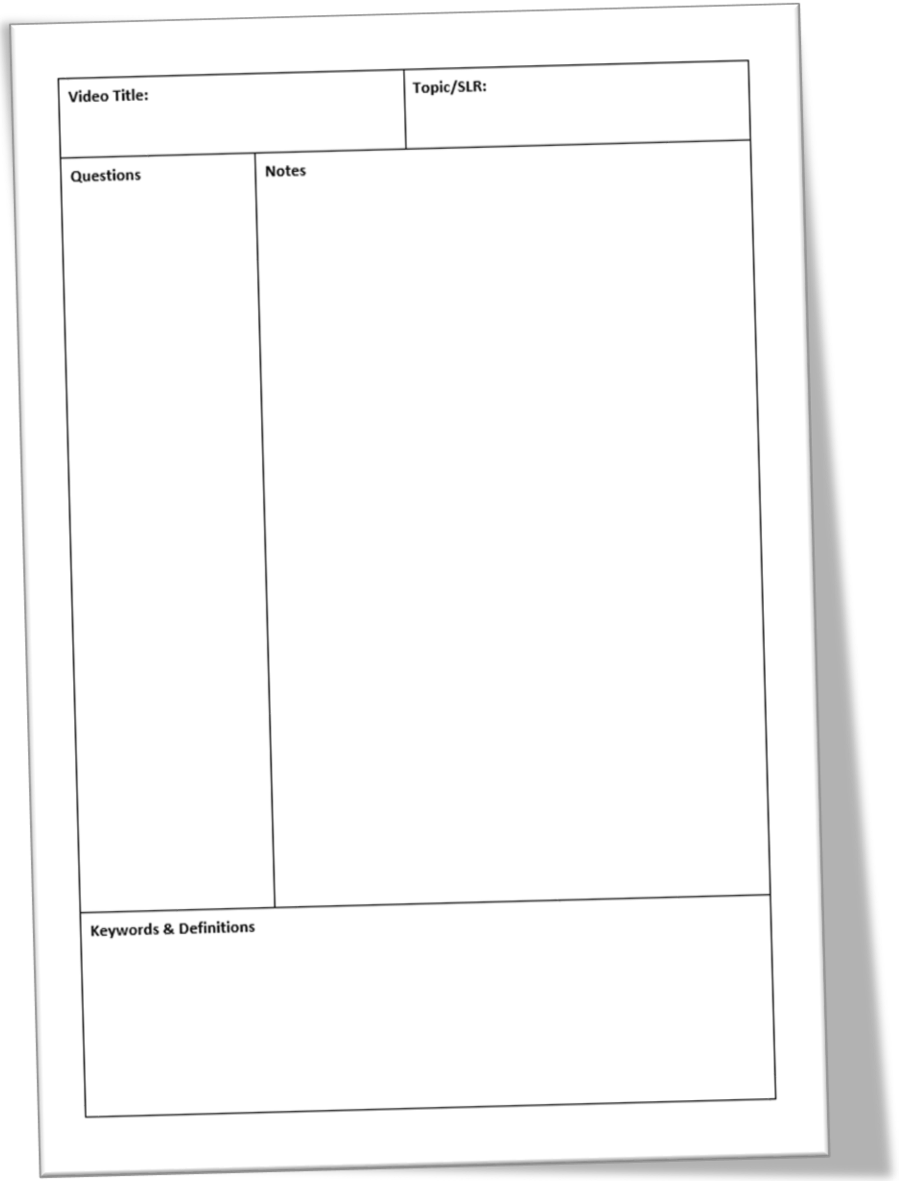
The Cornell method of note taking

Expected time to complete: 1½ hours

The expectation to do independent research at A Level will increase dramatically from GCSE.

There is a real skill to taking decent notes outside of lesson which are of value. Research has proven that one of the most effective methods is the "Cornell" note taking method.

1. To start download the "Cornell note taking template" from Craig n Dave:
   - https://craigndave.org/product/cornell-note-taking-template/
2. Pick any two of the following videos from Craig 'n' Dave:
   - AQA: https://student.craigndave.org/videos/aqa-alevel-slr06-procedural-functional-data-abstraction
   - AQA: https://student.craigndave.org/videos/aqa-alevel-slr13-sample-resolution-and-rate
   - AQA: https://student.craigndave.org/videos/aqa-alevel-slr13-encryption-vernam-cipher
   - AQA: https://student.craigndave.org/videos/aqa-alevel-slr20-bit-rate-baud-rate-bandwidth-and-latency
   - OCR: https://student.craigndave.org/videos/ocr-alevel-slr01-alu-cu-registers-and-buses
   - OCR: https://student.craigndave.org/videos/ocr-alevel-slr04-paging-segmentation-and-virtual-memory
   - OCR: https://student.craigndave.org/videos/ocr-alevel-slr05-stages-of-compilation
   - OCR: https://student.craigndave.org/videos/ocr-alevel-slr14-data-structures-part-2-graphs
3. Write the title of the video and its topic in the top boxes (use a different sheet for each video).
4. In the main "Notes" section, write notes from the video. You can do this in any way you like, a suggestion might be to rewind slightly when the canvas changes, thinking carefully about what was important in the previous few minutes.
5. Having recorded the notes, review them:
   - Turn each part into a question in the section on the left.
   - For example, the notes may say, "The value of the program counter is passed to the memory address register".
   - The question then becomes, "which register is the value of the program counter passed to?"
   - Sometimes these questions are easy, and at times they are more difficult to write.
   - There may also be more than one valid question.
   - You will need to decide for yourself which are the most appropriate questions for revision.
6. Finally pull out all the key words and their definitions words the notes and list them in the bottom section.



5 Key terms task

Craig'n'Dave

Getting to grips with terminology

Expected time to complete: 2 hours

An important aspect of being successful with your study of Computer Science is getting to grips with subject related terminology. There are over 240 specific terms you will need to learn!

Below are a handful of the key terms you will need to become familiar with.

Control Unit **Register** **Busses**

Von Neuman Architecture **Optical Storage** **Operating System**

Intermediate Code **Device Driver** **Compiler**

Assembly Language **Machine Code** **Lossy Compression**

Hashing **Normalisation** **TCP/IP Stack**

Packet Switching **ASCII** **Problem Decomposition**

1. Research each of the key terms and write a definition.
2. Resist the urge to simply cut and paste a definition off of the first website you find. Many definitions found on The Internet are overly complicated and wordy.
3. Ask yourself:
    - Does my definition make sense?
    - Is it succinct, to the point?
    - Does the definition have appropriate depth and detail for A'Level?
    - Could I give this definition to another student so they could revise from it?

Structure and function of the processor
Address Bus
1

Structure and function of the processor
Control Bus
12

Structure and function of the processor
Fetch-Decode-Execute
13

Structure and function of the processor
CPU
14

Structure and function of the processor
Clock Speed
15

Structure and function of the processor
Cores
6

Structure and function of the processor
Cache
17

Structure and function of the processor
Pipelining
18

Structure and function of the processor
Von Neumann Architecture
19

Structure and function of the processor
Harvard Architecture
20

Structure and function of the processor
Contemporary Architecture
1

Types of processor
CISC
22

Types of processor
RISC
23

Types of processor
GPU
24

Types of processor
Multicore System
25

Types of processor
Parallel Processor System
6

Input, output and storage
Input Device
27

Input, output and storage
Output Device
28

Input, output and storage
Storage Device
29

Input, output and storage
Magnetic Storage
30

Input, output and storage
Flash Storage
1

Input, output and storage
Optical Storage
32

Input, output and storage
RAM
33

Input, output and storage
ROM
34

Input, output and storage
Virtual Storage
35

5 Key terms task

Craig'n'Dave

Getting to grips with terminology

Expected time to complete: 2 hours

Term	Definition
Control Unit	"The part of the CPU that manages the execution of instructions. The control unit fetches each instruction in sequence, and decodes and synchronises it before executing it by sending control signals to other parts of the computer."
Register	"Tiny areas of extremely fast memory located in the CPU normally designed for a specific purpose, where data or control information is stored temporarily."
Busses	"A common physical pathway shared by signals to and from several components of a computer."
Von Neuman Architecture	"Traditional computer architecture that forms the basis of most digital computer systems. A single control unit manages program control flow following a linear sequence of "fetch-decode-execute""
Optical Storage	"Storage medium that uses plastic discs on which the data is stored as patterns on the surface in pits and lands."
Operating System	"Collection of systems software that manages the computer. Usually supplied with the computer. Most common operating systems are Windows, Linux, Unix, MacOS, iOS."
Intermediate Code	"Half-way type of code which is standard across machine types. Runs on a virtual machine."
Device Driver	"A computer program that operates or controls a particular type of device that is attached to a computer."
Compiler	"A program that translates a high-level language program, source code, into a computer's machine code."

5

Key terms task

Getting to grips with terminology

Expected time to complete: 2 hours

Term	Definition
Assembly Language	"A language which is related very closely to the computer's own machine code. "
Machine Code	"Set of all possible instructions made available by the hardware design of a particular processor. Closest to pure binary."
Lossy Compression	"A compression scheme where their generally involves a loss of resolution in parts of the image where experiences shows that it will be least noticed."
Hashing	"The process of calculating a numeric value from one or more data items. While this value obviously depends on the value of the data items, it need not depend on the meaning attached to them, simply producing a number that is used within the computer."
Normalisation	"The process of arranging data in tables and setting their relationships to move them through normal forms"
TCP/IP Stack	Transmission Control Protocol / Internet Protocol: "The most common general-purpose standard protocol that allows any networked computers (including those on The Internet) to communicate with each other whatever their equipment."
Packet Switching	"A method of sending data over a wide area network in which the message is broken into a number of parts which are sent independently, over whatever route is optimum for each packet, and reassembled at the destination."
ASCII	America Standard Code for Information Interchange: "A character set devised for early telecommunication systems but proved to be ideal for computer systems. ASCII codes use 7-bits giving 32 control codes and 96 displayable characters (the 8th bit is often used for error checking)."
Problem Decomposition	"The process by which a complex problem or system is broken down into parts that are easier to conceive, understand, program and maintain."

6 An introduction to the basics of programming tasks

Programming basics

Expected time to complete: 6 hours

Learning to "code" is a fun and essential part of A Level Computer Science. This task is ideal if you haven't done the GCSE in Computer Science or you simply want a nice refresher ahead of starting your A Level course.

1. Head over to the web site: https://www.learnpython.org/
2. Complete the following python tutorials under the heading:
   - Hello, World!
   - Variables and Types
   - Lists
   - Basic Operators
   - String Formatting
   - Basic String Operations
   - Conditions
   - Loops
   - Functions
3. Each section presents you with theory, code to run and exercises to try out.
4. If you want to practice writing your own python programs you can download and install a simple python development tool here: https://www.python.org/downloads/

Additional note:

This task is most suited to students who intend to do the A Level and have not previously gained much / or any programming experience from the GCSE Computer Science course.

Although the language chosen here is Python, and that may not be what you will be using at A Level, it is the underlying programming concepts which are important.

The list of topics above cover the standard set of programming concepts you would be expected to know having completed a GCSE and Computer Science and so will prepare you well for the A level.

7 Critical thinking task

Craig'n'Dave

Why is Computer Science important?

Expected time to complete: 2 hours

It is easy to say, “Computer Science is essential in todays world”, but are you able to think critically about this statement and back it up? “Thinking Critically” is an essential skill at A Level.
It involves you:

- Looking at a topic / concept in depth
- Taking account of different views / perspectives
- Considering positives and negatives
- Evaluating links and effects on other concepts
- Drawing your own conclusions backed up with evidence

1. On the following slide answer the questions:
    - What is Computer Science?
    - What are the benefits and risks of Computer Science at a local level (think about your local community / town / city / county)
    - What are the benefits and risks of Computer Science at a national level
    - What are the benefits and risks of Computer Science at a global level

Additional help:

For additional help and support in structuring your answer you might like to watch some of the videos from the following Craig ‘n’ Dave playlists:

OCR:
SLR 17 – Ethical, morale and cultural issues
https://student.craigndave.org/videos/slr-17-ethical-moral-and-cultural-issues

AQA:
SLR 19: Moral, social, legal, cultural issues
https://student.craigndave.org/videos/slr19-moral-social-legal-cultural-issues

7 Critical thinking task

Craig'n'Dave

Why is Computer Science important?

Expected time to complete: 2 hours

What is Computer Science?

- Enter your answer here

What are the benefits and risks of Computer Science at a local level

- Enter your answer here
- Try to make at least 4 valid points
- At least 2 of your points should be about the potential risks of Computer Science
- At least 2 of your points should be about the potential benefits of Computer Science

What are the benefits and risks of Computer Science at a national level

- Enter your answer here
- Try to make at least 4 valid points
- At least 2 of your points should be about the potential risks of Computer Science
- At least 2 of your points should be about the potential benefits of Computer Science

What are the benefits and risks of Computer Science at a global level

- Enter your answer here
- Try to make at least 4 valid points
- At least 2 of your points should be about the potential risks of Computer Science
- At least 2 of your points should be about the potential benefits of Computer Science

8 Applying technical knowledge in context task

Craig'n'Dave

Augmented reality

Expected time to complete: 1½ hours

A key skill at A Level is being able to take a topic and then discuss it in the context of different scenarios.

Most theory-based exam questions will be asked in the form of a scenario, simply regurgitating what you know on the topic without contextualising your answer to the scenario will often result in low marks!

The topic for this exercise is "Augmented Reality". It is a truly fascinating area of technology which has the potential to change almost every aspect of our daily lives.

Watch this brief video to learn more:
https://www.youtube.com/watch?v=vQtwWzfzKXI

After watching the video complete the next slide which asks you to discuss the benefits, limitations and risks of augmented reality in the context of:

- Medicine & health care
- Gaming & entertainment
- Schools & learning
- Travel & tourism
- Social media
- Transport & navigation

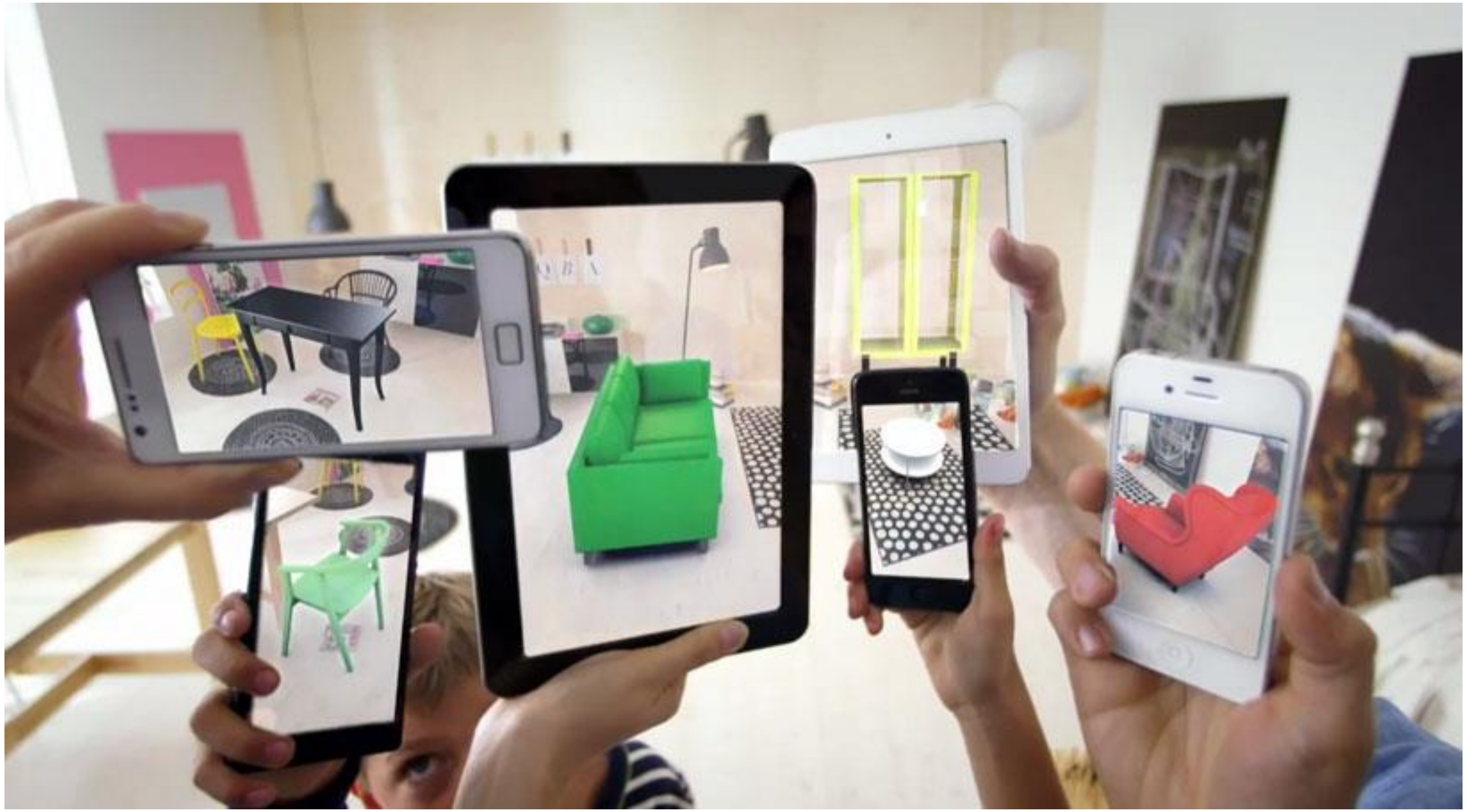


Image by Oyundari Zorigtbaatar (20 March 2016) https://creativecommons.org/licenses/by-sa/4.0/legalcode

8

Applying technical knowledge in context task

Craig'n'Dave

Augmented reality

Expected time to complete: 1½ hours

Medicine & health care

Discuss the **benefits**, **limitations** and **risks** of augmented reality in this context:

- Enter your answer here

Gaming & entertainment

Discuss the **benefits**, **limitations** and **risks** of augmented reality in this context:

- Enter your answer here

Schools & learning

Discuss the **benefits**, **limitations** and **risks** of augmented reality in this context:

- Enter your answer here

Travel & tourism

Discuss the **benefits**, **limitations** and **risks** of augmented reality in this context:

- Enter your answer here

Social media

Discuss the **benefits**, **limitations** and **risks** of augmented reality in this context:

- Enter your answer here

Transport & navigation

Discuss the **benefits**, **limitations** and **risks** of augmented reality in this context:

- Enter your answer here

9 Systems architecture task

Craig'n'Dave

Looking under the hood of the processor

Expected time to complete: 2 hours

The CPU "Central Processing Unit" is the central core of any computer system. You will study what it contains and how it works it in depth at A Level.

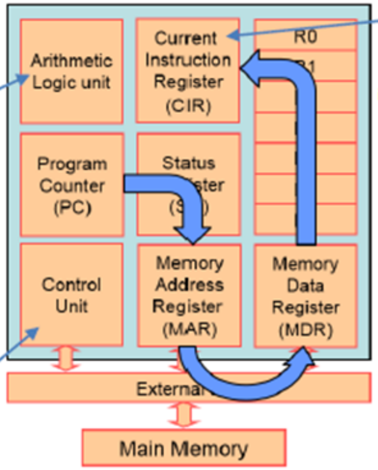
1. Start by watching the following 3 videos from Craig 'n' Dave (choose from OCR or AQA exam board)
   1. **OCR:** https://student.craigndave.org/videos/ocr-alevel-slr01-alu-cu-registers-and-buses
   2. **OCR:** https://student.craigndave.org/videos/ocr-alevel-slr01-fetch-decode-execute-cycle
   3. **OCR:** https://student.craigndave.org/videos/ocr-alevel-slr01-performance-of-the-cpu
   1. **AQA:** https://student.craigndave.org/videos/aqa-alevel-slr17-the-processor-and-its-major-components
   2. **AQA:** https://student.craigndave.org/videos/aqa-alevel-slr17-alu-cu-registers-and-buses
   3. **AQA:** https://student.craigndave.org/videos/aqa-alevel-slr17-performance-of-the-cpu

2. Produce a fully annotated diagram on a single sheet of A4 / A3 paper which shows how the CPU works.

3. Make sure the diagram includes and covers:
   - Major CPU components and what they are for:
     - Arithmetic Logic Unit (ALU)
     - Control Unit (CU)
     - Cache
   - The main registers
     - Program Counter (PC)
     - Memory Address Register (MAR)
     - Current Instruction Register (CIR)
     - Memory Data/Buffer Register (MDR / MBR)
   - Fetch-decode-execute cycle
   - Include annotations which explain how the performance of a CPU can be improved by:
     - Increasing the clock speed
     - Increasing the cache size
     - Increasing the number of cores

Arithmetic Logic Unit: "The part of the CPU where data is processed and manipulated. This processing and manipulation normally consists of arithmetic operations or logical comparisons allowing a program to make decisions."

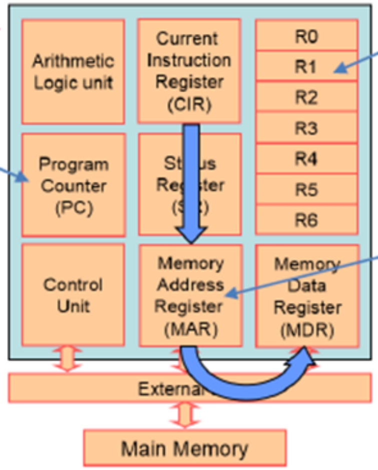
Control Unit: "The part of the CPU that manages the execution of instructions. The control unit fetches each instruction in sequence, and decodes and synchronises it before executing it by sending control signals to other parts of the computer."



Current Instruction Register: "A register in the control unit that stores the address of the next instruction currently being executed and decoded."

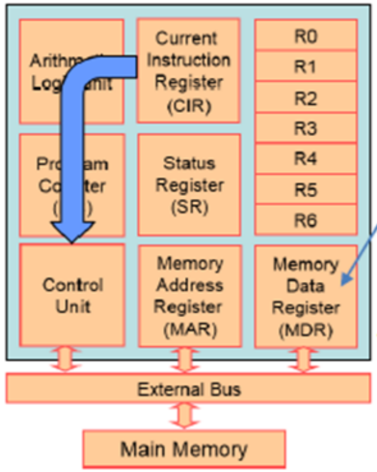
Register: "Tiny areas of extremely fast memory located in the CPU normally designed for a specific purpose, where data or control information is stored temporarily."

Program Counter: "A register in the control unit which holds the address of the next instruction to be executed."



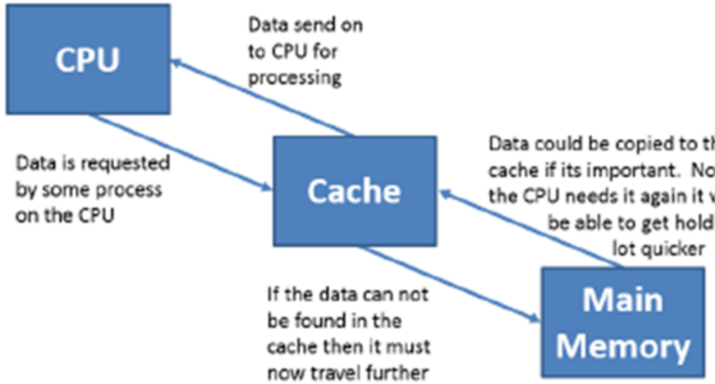
Accumulator: "A special register within the ALU. It is used to hold the data currently being processed by the central processor. Any data to be processed is stored temporarily in the accumulator, the results ending up back in the accumulator being stored in the memory unit."

Memory Address Register: "A register in the CPU that stores the address of the memory location currently in use. In the fetch phase, this would be the address of the instruction being loaded; in the execute phase, it would be the address of the data being used."



Memory Data Register: "A register in the CPU that stores data being transferred to and from the immediate-access store. It acts as a buffer, allowing the central processor and memory unit to act independently without being affected by minor differences in operation. A data item will be copied to the MDR ready for use at the next clock pulse, when it can either be used by the central processor or be stored in main memory."

Cache: "A part of the main store between the central processor and the rest of the memory. It has extremely fast access, so sections of a program and its associated data are copied there to take advantage of its short fetch cycle."



Data send on to CPU for processing

Data is requested by some process on the CPU

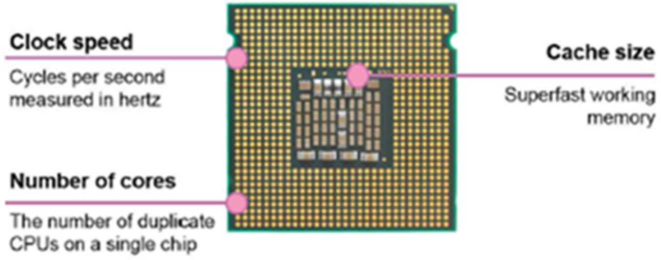
Data could be copied to the cache if its important. Now if the CPU needs it again it will be able to get hold of it a lot quicker

If the data can not be found in the cache then it must now travel further and check in the main memory

PROCESSOR CLOCK SPEEDS

One cycle per second = 1 Hertz (Hz) = 1 instruction carried out each second
1 Kilohertz (KHz) = 1024 cycles per second
1 Megahertz (MHz) = 1,048,576 cycles per second
1 Gigahertz (GHz) = 1,073,741,824 cycles per second (Approximately 1 billion!)

Cores: "A part of a multi-core processor. A multi-core processor is a single component with two or more independent actual CPUs, which are the units responsibly for the fetch-decode-execute cycle."



Clock speed
Cycles per second measured in hertz

Number of cores
The number of duplicate CPUs on a single chip

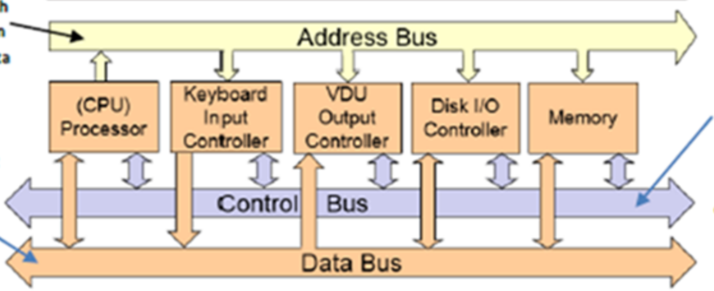
Cache size
Superfast working memory

Clock Speed: "Measured in Hertz, the clock speed is the frequency at which the internal clock generates pulses. The higher the clock rate, the faster the computer may work. The "clock" is the electronic unit that synchronises related components by generating pulses at a constant rate."

Busses: "A common physical pathway shared by signals to and from several components of a computer."

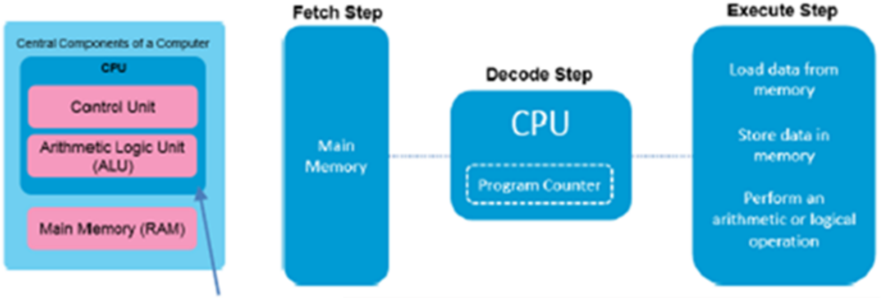
Address Bus: "The part of the bus which carries identification about where the data is being sent."

Data Bus: "The part of the bus which carries the actual information."



Control Bus: "This bus carries command and control signals to and from every other component of a computer."

Von Neumann Architecture: "Traditional computer architecture that forms the basis of most digital computer systems. A single control unit manages program control flow following a linear sequence of "fetch-decode-execute"



Central Processing Unit: "The main part of the computer, consisting of the registers, ALU and control unit."

F-D-E: "The complete process of retrieving an instruction from store, decoding it and carrying it out. Also known as the instruction cycle."

10 Memory task

Craig'n'Dave

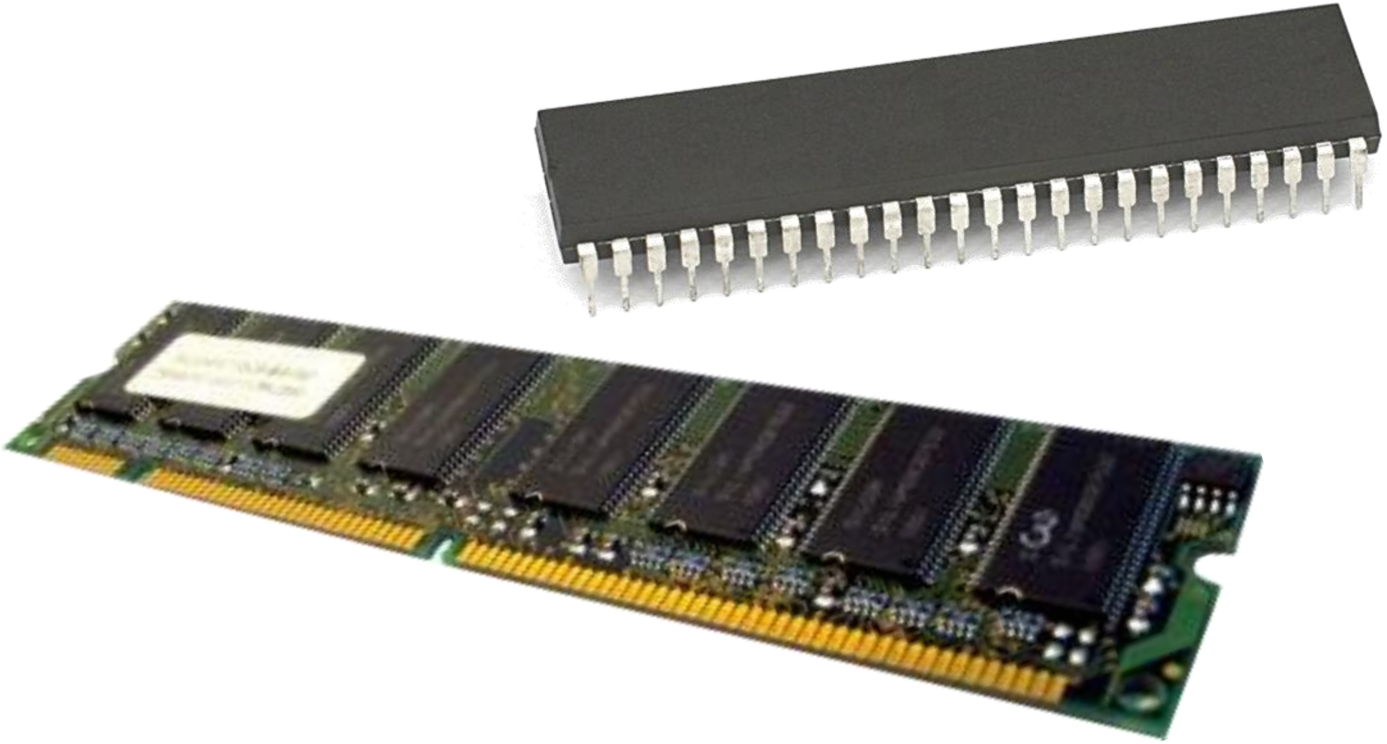
Different types of memory

Expected time to complete: 1 hour

Computer memory comes in many different forms, some of the main ones are:

- Random Access Memory (RAM)
- Read Only Memory (ROM)
- Virtual Memory

Carry out some research into these forms of memory and then complete the tasks on the following slide.



Additional help:

For additional help and support in completing this task you might like to watch some of the following videos from Craig 'n' Dave:

RAM and ROM:
https://student.craigndave.org/videos/ocr-gcse-slr1-2-ram-and-rom

The need for Virtual Memory:
https://student.craigndave.org/videos/ocr-gcse-slr1-2-the-need-for-virtual-memory

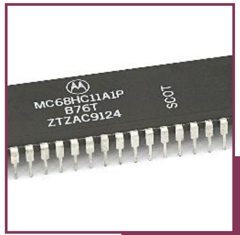
10 Memory task

Different types of memory

Expected time to complete: 1 hour

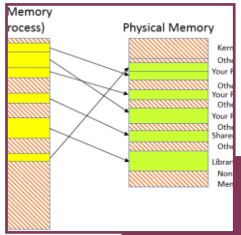
Random Access Memory (RAM)

- A form of primary storage used to store instructions and data which need to be used by the CPU while a computer is in operation.
- Volatile
- Readable and writable
- Typically holds the part of the operating system currently in use, applications being used, the data those applications are using.



Read Only Memory (ROM)

- A form of primary storage which contains essential instructions on who to start up a computer, often written at manufacturing time.
- Non-volatile
- Read only
- Initial instructions your computing first needs when booting up, known as the Bootstrap.



Virtual Memory

- Using part of the hard disk or other secondary storage device as if it where main memory (RAM).
- This allows for more applications to be open and running than physical main memory could normally hold.

11 Storage task

Expected time to complete: 1 hour

Types of secondary storage

Virtually all secondary storage devices in use today fit into one of three broad categories:

- Magnetic
- Optical
- Solid state

Carry out some research into these categories and then complete the tasks on the following slide.

Additional help:

For additional help and support in completing this task you might like to watch some of the following videos from Craig 'n' Dave:

Magnetic, Flash and Optical storage:
https://student.craigndave.org/videos/ocr-alevel-slr03-magnetic-flash-and-optical-storage

Comparing capacity and speed of storage media:
https://student.craigndave.org/videos/aqa-alevel-slr18-comparing-capacity-and-speed-of-storage-media

11 Storage task

Types of secondary storage

Expected time to complete: 1 hour

Optical Storage

How does it work?

For read only optical storage it stores information in pits and lands which are burnt into disc when it is created. A laser passes over the surfer and at the point when the pits start or end it causes the laser light to scatter. This change in reflection is interrupted as 0's and 1's

Advantages / Positives

- Cheap to produce
- Light weight
- Highly portable

Disadvantages / Negatives

- Slow access times compared to other forms of storage
- Prone to scratches

Examples of typical usage

Multimedia files, e.g. audio, files etc.
Sometimes used for backup and archiving

Typical storage capacity range

CD 720MB~ DVD 4.7-17 GB~ Blu-ray 25GB-128GB~

Magnetic Storage

How does it work?

Uses magnetic fields to store 0's and 1's.
A drive head physically moves over the surface of a hard drive platter reading the different magnetic fields.

Advantages / Positives

- Very cheap to produce
- Very large capacity

Disadvantages / Negatives

- Slow access times as the platters have to spin up to speed and the drive head needs to move to read/write data.
- Fragile as it is made of so many intricate moving parts.

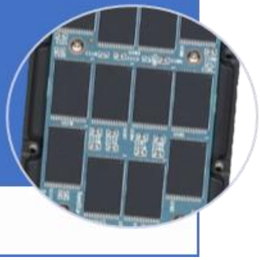
Examples of typical usage

Large mass storage of data, applications and system files
Good for off site backup

Typical storage capacity range

Typically anywhere from 500GB – 10TB and increasing

Solid State Storage



How does it work?

A flow of electricity forces electrons into floating gates between two oxide layers. This causes a change in the charge in the floating gate and this can be measured as a 0 or a 1.

Advantages / Positives

- Durable as it has no moving parts, can cope with knocks and bumps
- Fast access times

Disadvantages / Negatives

- Currently the most costly of the three types of storage Mb for Mb, though costs are coming down all the time.
- Has a limited number of read / write's

Examples of typical usage

Large mass storage of data, applications and system files
Good for off site backup

Typical storage capacity range

Typically anywhere from 250GB – 4TB and increasing

12 Networks task

Craig'n'Dave

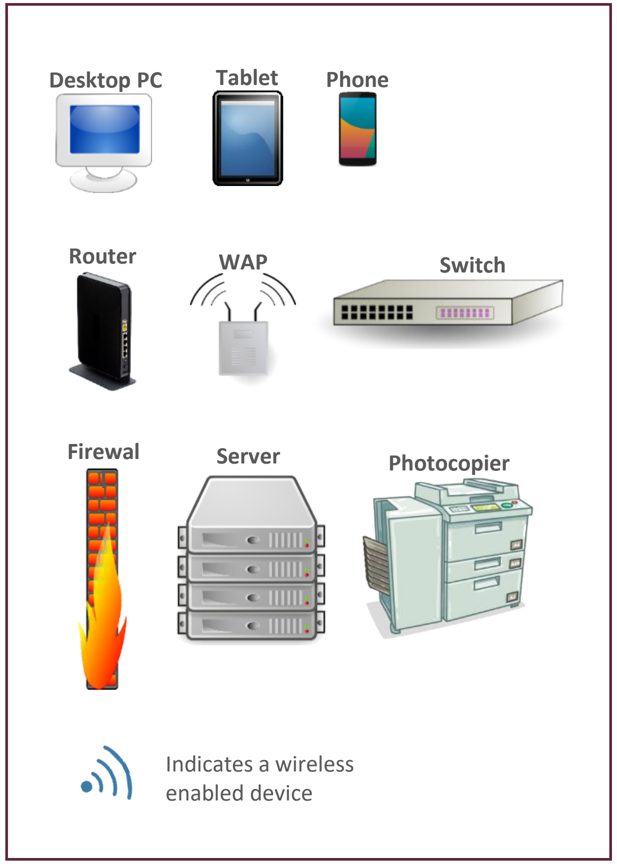
Types of networks

Expected time to complete: 2 hours

Carry out some research on computer networks, in particular LANs, WiFi, Network topologies and connectivity devices.
Use the symbols on the right (feel free to revise them) to create an appropriate network over the floorplan on the next slide.
Make sure your network meets all the following requirements:

1. Each member of the main office needs a desktop PC
2. Angela, Pam, Dwight and Oscar also use an office issued smart phone
3. The following rooms new access to WiFi:
    1. Meeting room (top right)
    2. Reception
    3. Conference Room
    4. Main office
4. Use a circle with a transparent fill (so you can see the network underneath) with a width and height of 12.5cm to provide the WiFi coverage needed to cover the rooms above:
    1. The circles need to have a WAP at the centre
    2. The 12.5cm diameter circles represent the maximum range of each WAP
    3. They WAP icons must be attached to a wall
    4. You must use the minimum number of WAP possible to provide the coverage needed
5. All desktop PCs use wired connections in a star network configuration
    1. The top left server room, conference room and main office need to be on one subnet with its own switch
    2. All other rooms are on a separate subnet and will require its own hardware for this
    3. The two subnets need to be appropriately connected together
6. The top left room needs to have a server placed in it and connected appropriately to the local subnet
7. The server room needs hardware to appropriately connected the LAN to "The Internet"
8. Reception needs a photocopier and it needs connecting to the local subnet
9. A firewall should be placed somewhere appropriate

Use the following symbols:



Indicates a wireless enabled device

Additional help:

For additional help and support in structuring your answer you might like to watch some of the videos from the following Craig 'n' Dave playlists:

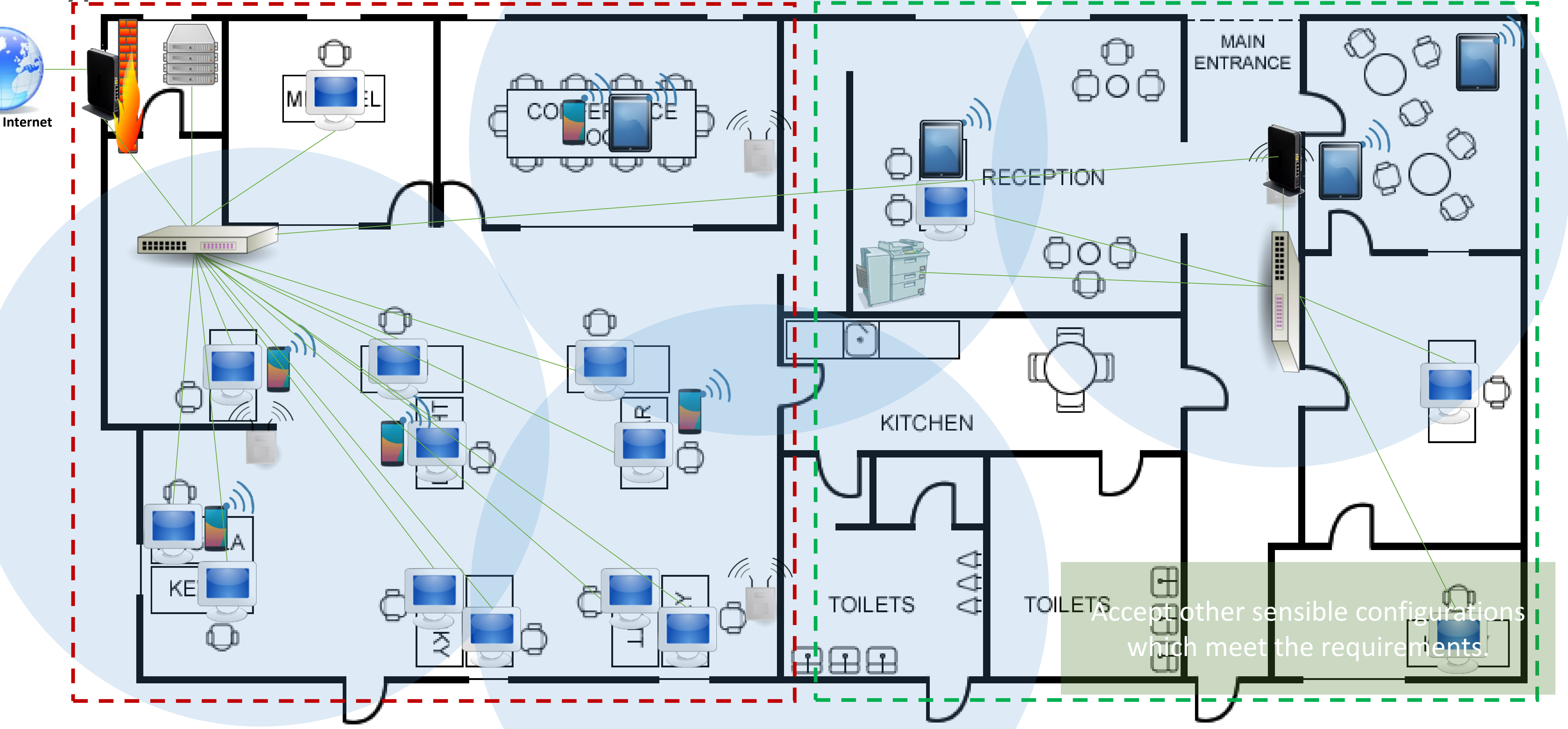
OCR: SLR 11 – Networks
https://student.craigndave.org/videos/slr-11-networks
AQA: SLR21 – Network and the Internet
https://student.craigndave.org/videos/slr21-networks-the-internet

12 Networks task

Craig'n'Dave

Types of networks

Expected time to complete: 2 hours



Accept other sensible configurations which meet the requirements.

13 Systems software task

Craig'n'Dave

Operating systems

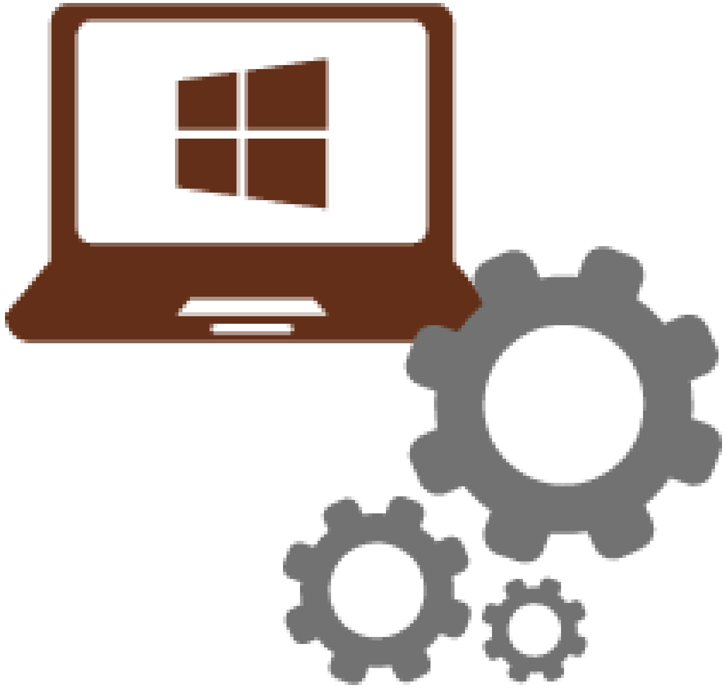
Expected time to complete: 2 hours

Operating systems are arguably the most important piece of software installed on a computer.

Carry out some research into the following areas:

- The purpose of operating systems
- The roles of operating systems
- The purpose of interrupts
- How interrupts work as part of the fetch-decode-execute cycle

Complete the tasks on the following slides.



Additional help:

For additional help and support in structuring your answer you might like to watch some of the videos from the following Craig 'n' Dave playlists:

OCR: SLR 4 – Operating systems – Systems software
https://student.craigndave.org/videos/slr-4-operating-systems-systems-software

AQA: SLR14 – Hardware & software
https://student.craigndave.org/videos/slr14-hardware-software

13 Systems software task

Operating systems

Expected time to complete: 2 hours

1. List at least 8 different roles an Operating System perform.

The purpose and roles of an Operating System

- Managing hardware
- Managing installed programs and those being run
- Managing systems security
- Providing an interface between the user and the computer
- Managing the processor
- Managing memory
- Handling input / output from external peripherals
- Managing networking

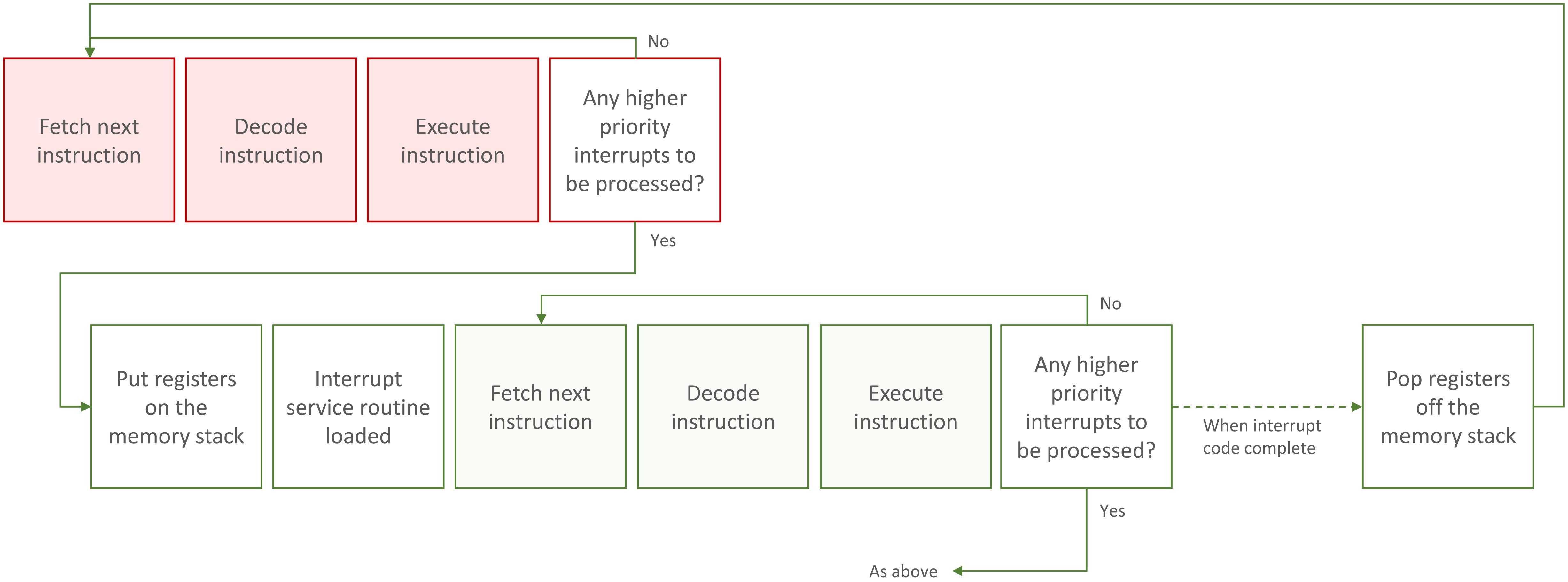
13 Systems software task

Craig'n'Dave

Operating systems

Expected time to complete: 2 hours

2. Rearrange the elements in the diagram below to show hoe the fetch-decode-execute cycle handles interrupts.

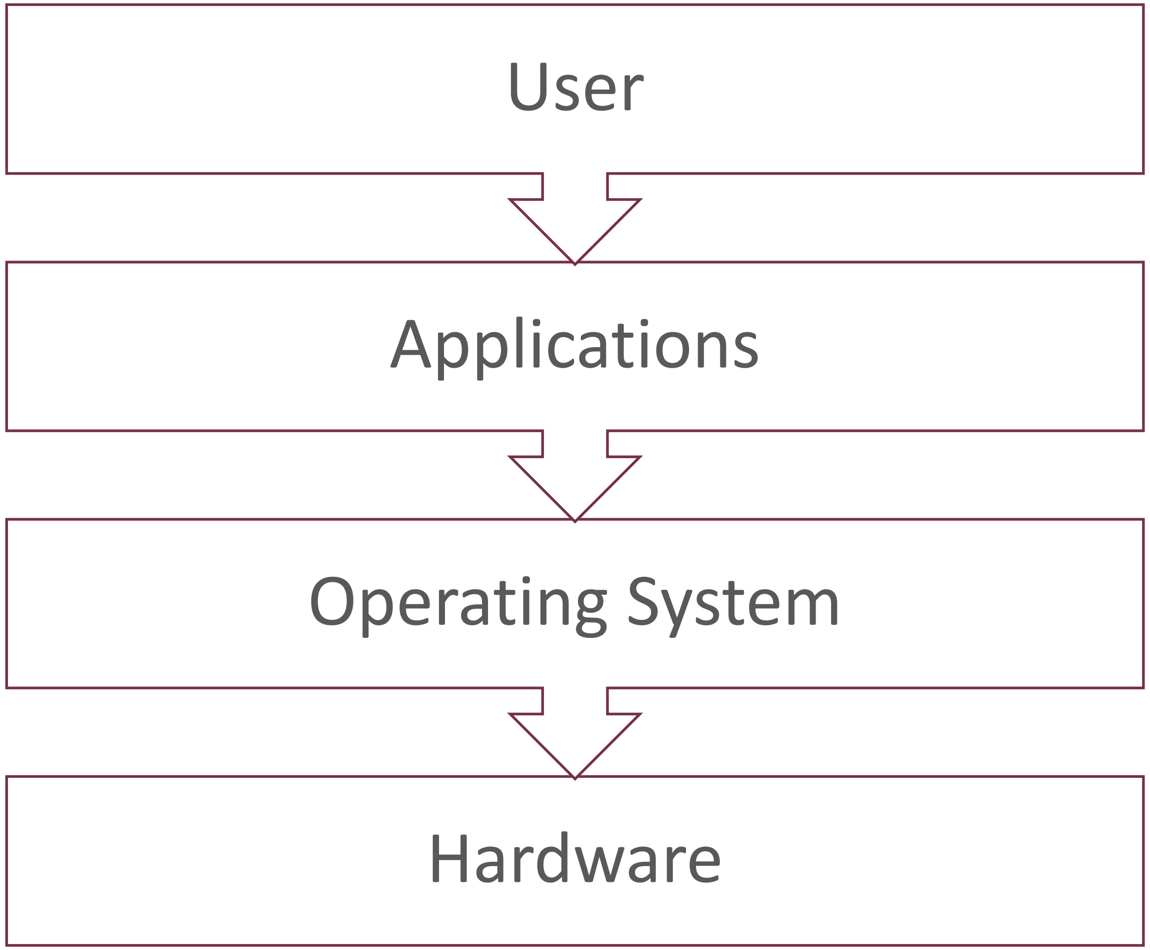


No
Fetch next instruction
Decode instruction
Execute instruction
Any higher priority interrupts to be processed?
Yes
No
Put registers on the memory stack
Interrupt service routine loaded
Fetch next instruction
Decode instruction
Execute instruction
Any higher priority interrupts to be processed?
When interrupt code complete
Pop registers off the memory stack
Yes
As above

Operating systems

Expected time to complete: 2 hours

3. Drag the labels into their correct place on the following diagram:



4. Explain what this diagram is showing:

- The operating system is a layer of software between the applications and the hardware.
- It manages open applications, providing a user interface, multi-tasking, security and memory management.
- It provides a mechanism for the applications to use the hardware, sending outputs to the device drivers, and receiving input from the devices.
- The complexity of storing, retrieving, inputting and outputting to the hardware is transparent to the user.

14 Computational logic task

Truth tables to circuit diagrams

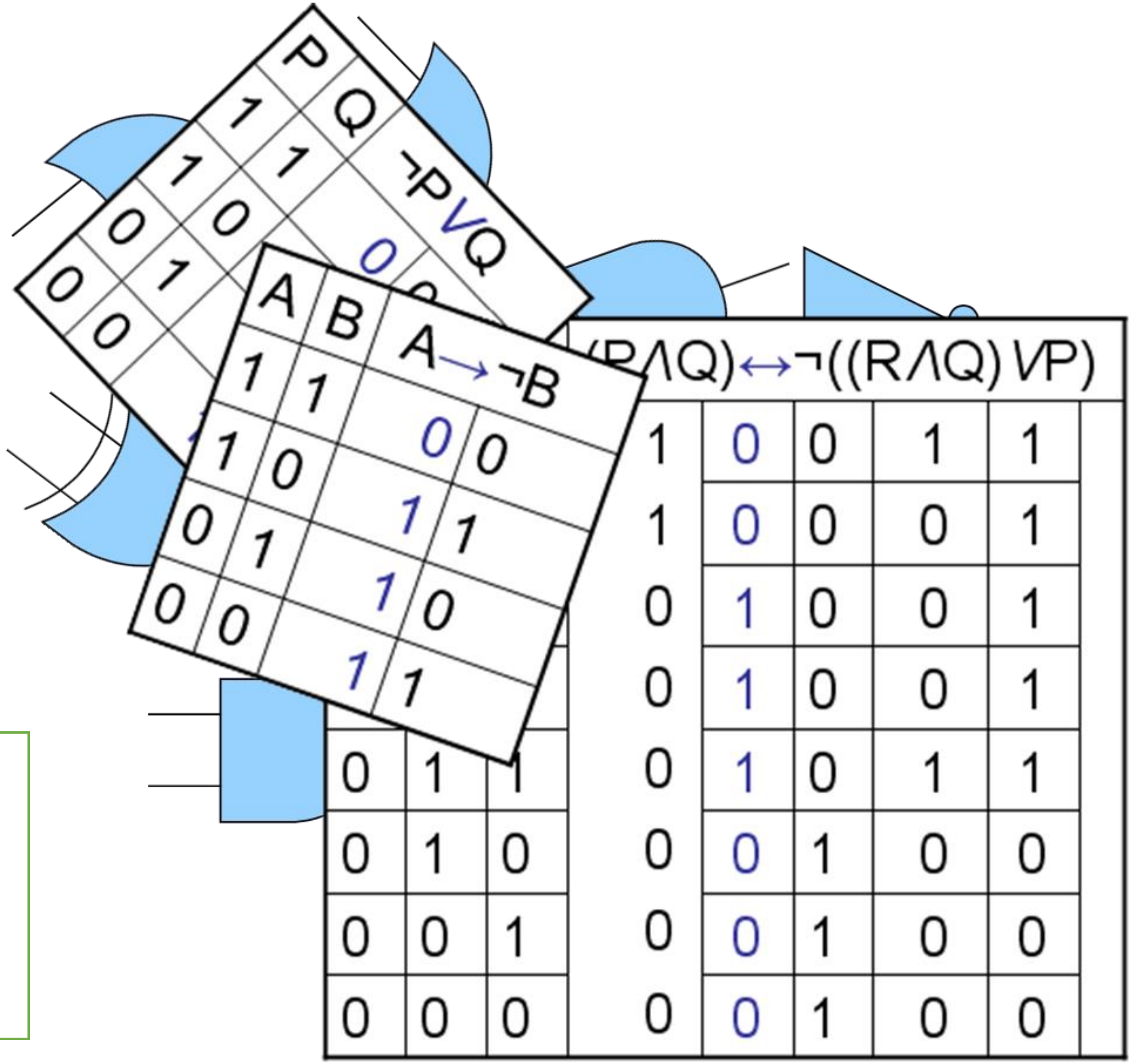
Expected time to complete: 2 hours

An important area of computer science is understanding the logic gates and diagrams which are used to represent the physical circuitry of computer systems.

Carry out some research into the following areas:

- Logic gates:
    - AND
    - NAND
    - NOR
    - NOT
    - OR
    - XOR
- Truth tables
- Boolean expressions
- Circuit diagrams

Complete the tasks on the following slides.



Additional help:

For additional help and support in structuring your answer you might like to watch some of the videos from the following Craig 'n' Dave playlists:

OCR: SLR 15 – Boolean algebra
https://student.craigndave.org/videos/slr-15-boolean-algebra

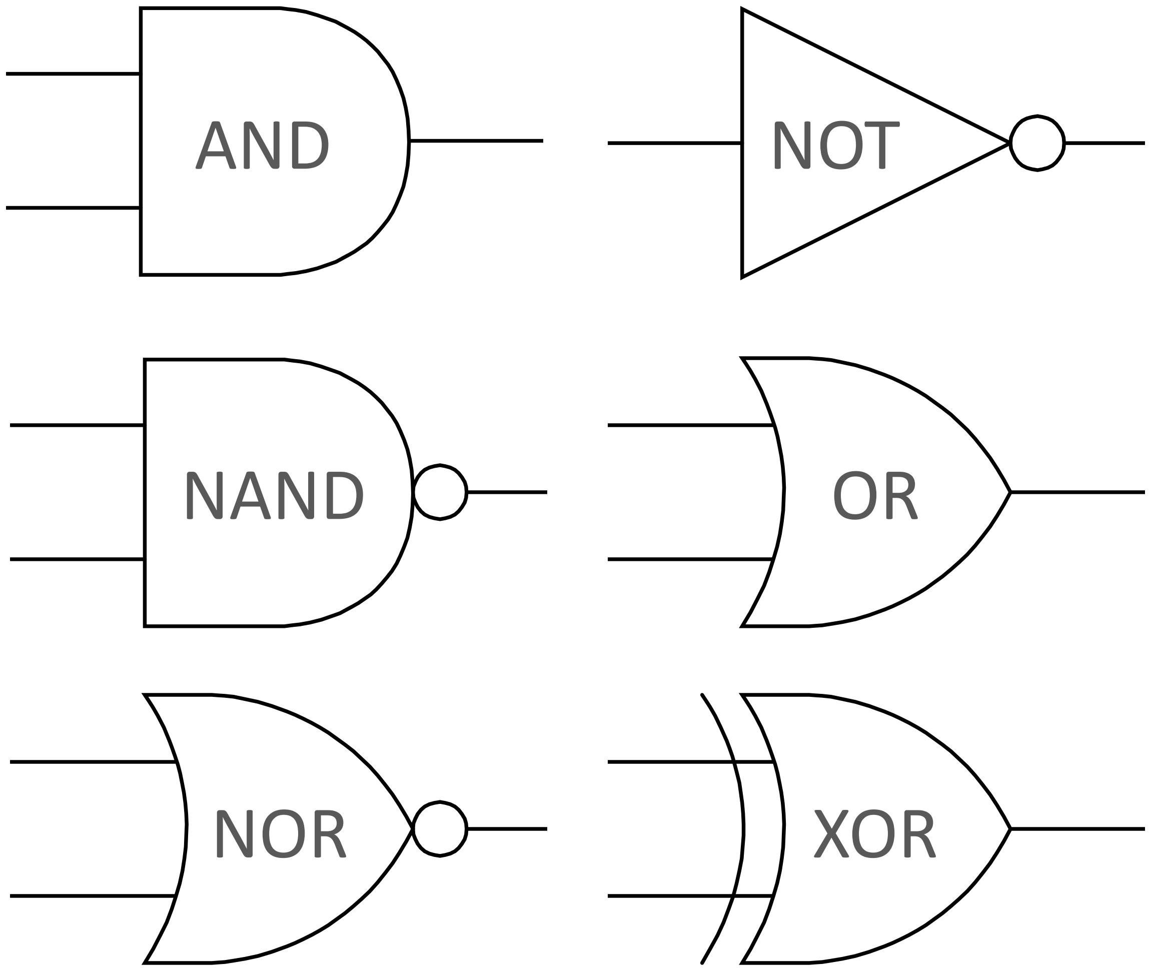
AQA: SLR16 – Logic gates & Boolean algebra
https://student.craigndave.org/videos/slr16-logic-gates-boolean-algebra

14 Computational logic task

Truth tables to circuit diagrams

Expected time to complete: 2 hours

1. Drag the labels into their correct place on the following diagram:



14 Computational logic task

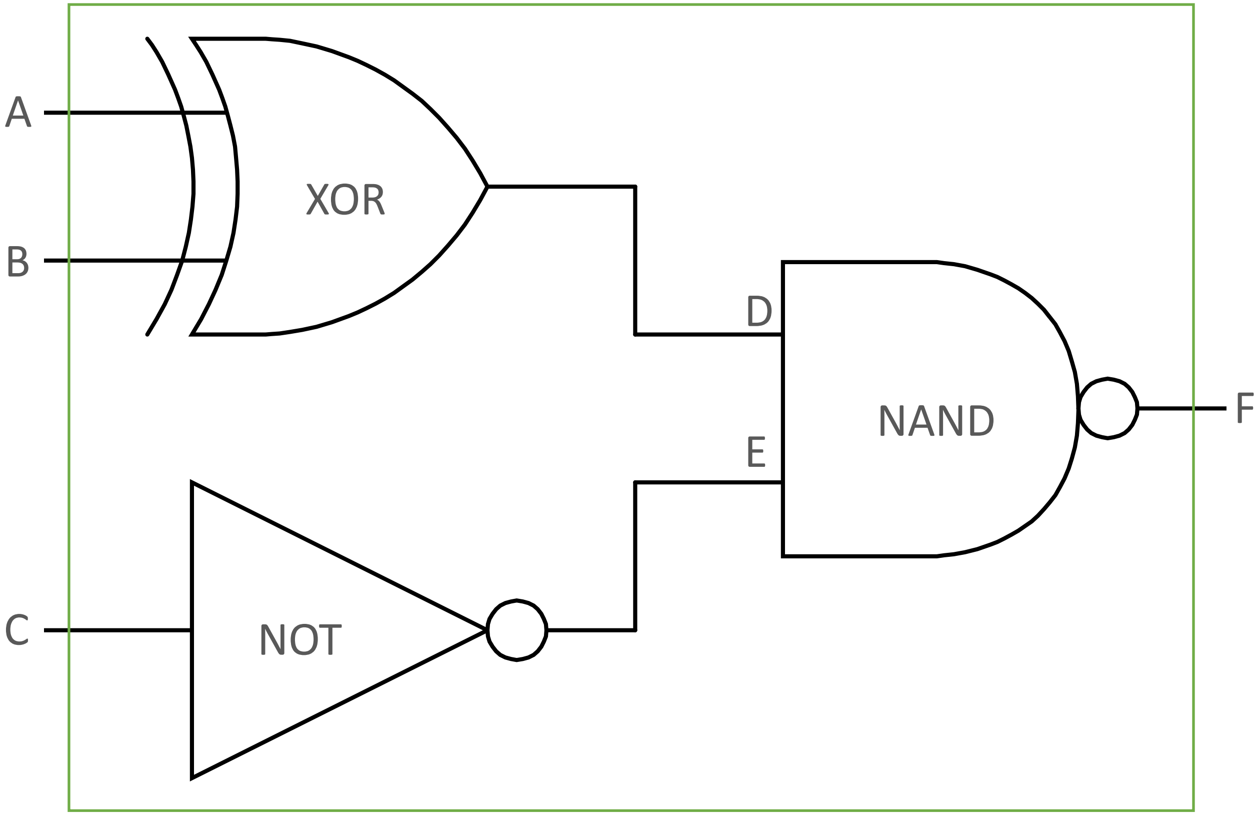
Truth tables to circuit diagrams

Expected time to complete: 2 hours

2. Draw the circuit diagram which would represent the following Boolean expression:

AQA Boolean Expression: $F = \overline{(A \oplus B) \bullet \overline{C}}$

OCR Boolean Expression: **F = ¬(¬C ∧ (A ⊻ B)**



3. Complete the truth table for the circuit diagram you have drawn

A	B	C	D	E	F
0	0	0	0	1	1
0	0	1	0	0	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	1	1	0
1	0	1	1	0	1
1	1	0	0	1	1
1	1	1	0	0	1

15

Binary task

Craig'n'Dave

Representing negative numbers in binary

Expected time to complete: 1½ hours

In GCSE computer science you will have learnt how to represent positive whole numbers in binary e.g. 47

At A Level you will need to know how to represent negative as well e.g. -47

Start to recapping (or learning if you didn't do the GCSE) how to represent positive whole numbers between 0-255 in binary

Now research how to represent negative numbers in binary using the method known as:

- Two's complement

Complete the tasks on the following slides.

Additional help:

For additional help and support in structuring your answer you might like to watch some of the following videos from Craig 'n' Dave:

GCSE recap: How to represent positive binary values 0-255
https://student.craigndave.org/videos/aqa-gcse-slr13-number-bases

A Level: Representing negative binary values using Two's Complement
https://student.craigndave.org/videos/aqa-alevel-slr11-twos-complement

15 Binary task

Representing negative numbers in binary

Expected time to complete: 1½ hours

1. Write out the positive binary number 107, the answer should be displayed in 8 bits.

128	64	32	16	8	4	2	1	Binary weighting
0	1	1	0	1	0	1	1	Binary value

64 + 32 + 8 + 2 + 1 = 107

2. Write out the negative binary number -107 using Two's Complement, the answer should be displayed in 8 bits.

-128	64	32	16	8	4	2	1	Binary weighting
1	0	0	1	0	1	0	1	Binary value

-128 + 16 + 4 + 1 = -107

3. How would you represent the lowest negative number possible using Two's Complement, given 8 bits.

-128	64	32	16	8	4	2	1	Binary weighting
1	0	0	0	0	0	0	0	Binary value

4. How would you represent the largest positive number possible using Two's Complement, given 8 bits.

-128	64	32	16	8	4	2	1	Binary weighting
0	1	1	1	1	1	1	1	Binary value

16 Binary task

Craig'n'Dave

Converting between base-2, base-10 and base-16

Expected time to complete: 1½ hours

As humans we have use the decimal or denary number system (base-10), made up of the unique digits 0-9.

Computer systems at the most basic level use only binary 1's and 0's (base-2).

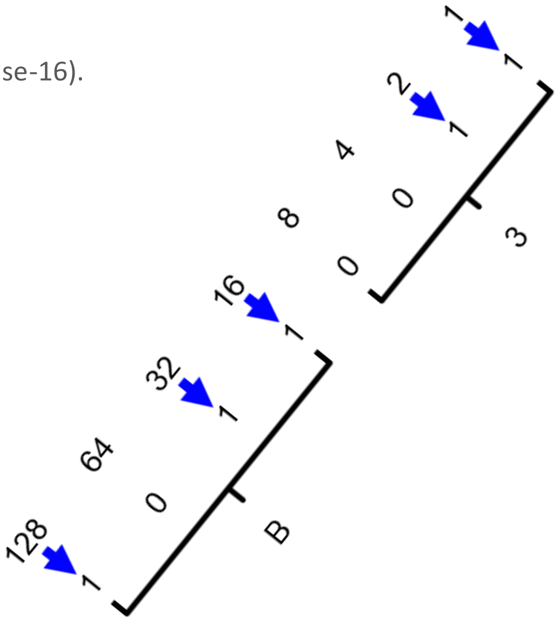
As a computer scientist you will also need to become familiar with the hexadecimal number system (base-16).

You will also need to be comfortable with converting numbers between these three base systems.

Research the following areas:

- Base-2 binary number system
- Base-10 decimal/denary number system
- Base-16 hexadecimal number system
- How to convert between base-2, base-10 and base-16

Complete the tasks on the following slides.



Denary	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Additional help:

For additional help and support in structuring your answer you might like to watch some of the following videos from Craig 'n' Dave:

Base 2, 10 and 16 number systems:
https://student.craigndave.org/videos/aqa-alevel-slr10-base-2-10-and-16-number-systems

Converting between binary, hex and decimal:
https://student.craigndave.org/videos/aqa-alevel-slr11-aqa-converting-between-binary-hex-and-decimal

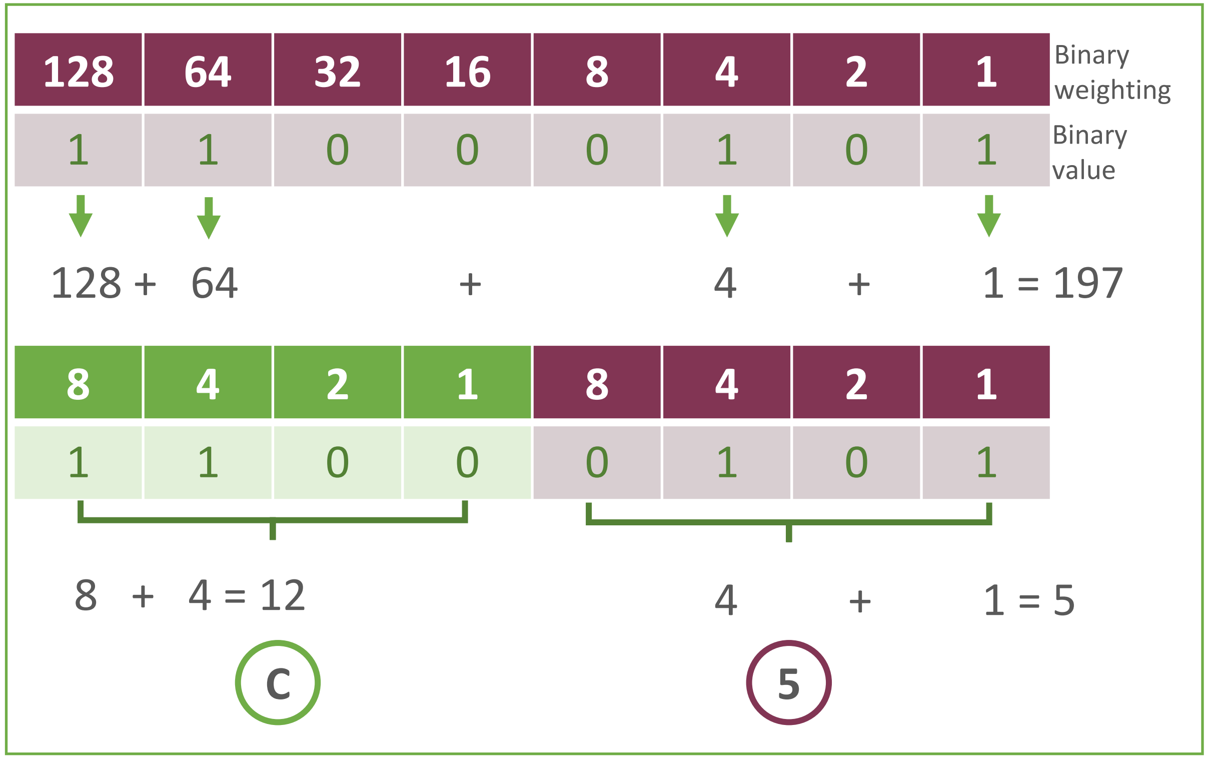
16 Binary task

Craig'n'Dave

Converting between base-2, base-10 and base-16

Expected time to complete: 1½ hours

1. Convert the base-2 binary number 11000101 into base-10 and base-16.
(Show all your working)



128	64	32	16	8	4	2	1	Binary weighting
1	1	0	0	0	1	0	1	Binary value

128 + 64 + 4 + 1 = 197

8	4	2	1	8	4	2	1
1	1	0	0	0	1	0	1

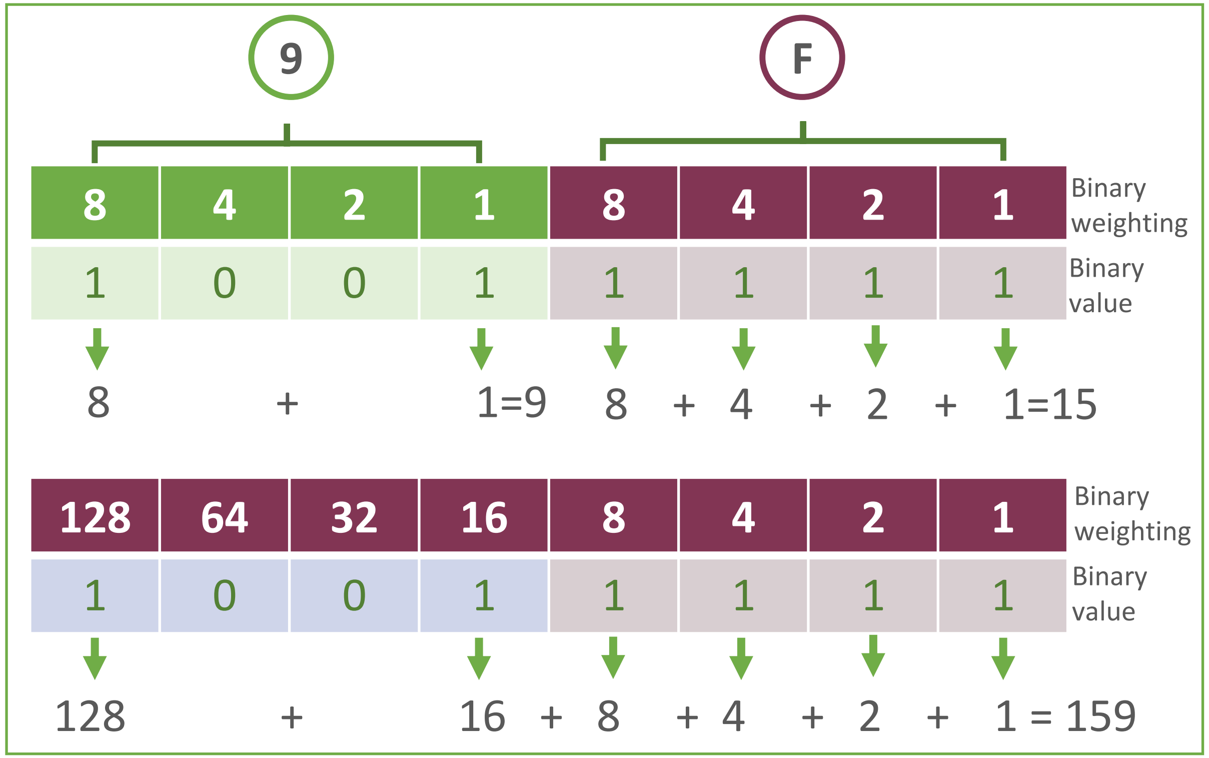
8 + 4 = 12

4 + 1 = 5

C

5

2. Convert the base-16 hexadecimal number 9F into base-2 and base-10.
(Show all your working)



9

F

8	4	2	1	8	4	2	1	Binary weighting
1	0	0	1	1	1	1	1	Binary value

8 + 1=9 8 + 4 + 2 + 1=15

128	64	32	16	8	4	2	1	Binary weighting
1	0	0	1	1	1	1	1	Binary value

128 + 16 + 8 + 4 + 2 + 1 = 159

17 Linear search programming task

Craig'n'Dave

Algorithms: from theory to practice

Expected time to complete: 4 hours

A core concept of computer science is that of data structures and algorithms.

It is also an area which many students struggle with during examinations.

Probably the most basic algorithm is that of the "linear search".

If you have done the GCSE course you will have learnt about this searching algorithm already.

Start by learning or refreshing your knowledge of the linear search algorithm by using the videos on this page:

- https://www.craigndave.org/algorithms-linear-search

Once you are happy with the theory complete the exercises on the following slides.

17 Linear search programming task

Craig'n'Dave

Algorithms: from theory to practice

Expected time to complete: 2 hours

1. Describe what the linear search algorithm does.

- The linear search finds an item in a sorted or unsorted list.
- To perform a linear search, start at the first item in the list and check each item one by one.
- Think about searching for a card in a shuffled deck, starting with the top card and checking each one until you find the card you want.

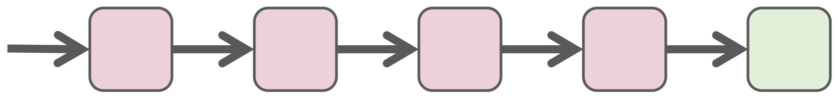
2. What are the applications of the linear search algorithm,?

- The linear search is ideal for finding items in small data sets and performing searches with unordered data such as settings files.
- It is the easiest searching algorithm to implement but usually the most inefficient.

3. Write out the steps of the linear search algorithm in simple-structured English.

1. Start at the first item in the list.
2. If the item in the list is the one to be found, the search is complete.
3. If it is not, move to the next item.
4. Repeat from step 2 until the item is found or there are no more items in the list.
5. If the item has been found, output item data. If it has not, output “not found.”

4. Draw a simple diagram which illustrates the linear search algorithm.



Linear search programming task

Algorithms: from theory to practice

Expected time to complete: 2 hours

5. Write out pseudocode for the linear search algorithm.

- The algorithm should use an array called items which is pre-populated with the following values: "Florida","Georgia","Delaware","Alabama","California“
- The algorithm should ask the user to “Enter the state to find:”
- If the algorithm locates the state entered by the user in the array it should report back to the screen “Item found at position n”
- If the algorithms can not locate the state entered by the user in the array it should report back to the screen “Item not found”

```
items = ["Florida","Georgia","Delaware","Alabama","California"]
item_to_find = input("Enter the state to find: ")
index = 0
found = False
While found == False and index < items.Length
        If items[index] == item_to_find then
found = True
        Else
                index = index + 1
        End If
End while
If found == True then
        Print "Item found at position ",index
Else
        Print "Item not found"
End If
```

17 Linear search programming task

Algorithms: from theory to practice

Expected time to complete: 2 hours

6. Have a go at coding the linear searching algorithm in a programming language of your choice.

- The program should work use an array called items which is pre-populated with the following values: "Florida","Georgia","Delaware","Alabama","California“
- The program should ask the user to “Enter the state to find:”
- If the program locates the state entered by the user in the array it should report back to the screen “Item found at position n”
- If the program can not locate the state entered by the user in the array it should report back to the screen “Item not found”

Coded solution in Python:

```
items = ["Florida","Georgia","Delaware","Alabama","California"]
item_to_find = input("Enter the state to find: ")
index = 0
found = False
while found == False and index < len(items):
    if items[index] == item_to_find:
        found = True
    else:
        index = index + 1
if found == True:
    print ("Item found at position",index)
else:
    print ("Item not found")
```

Coded solution in Visual Basic:

```
Dim items() As String = {"Florida", "California", "Delaware", "Alabama", "Georgia"}
Dim item_to_find As String
Dim index As Integer = 0
Dim found As Boolean = False
Console.WriteLine("Enter the state to find: ")
item_to_find = Console.ReadLine
Do While found = False And index < items.Count
   If items(index) = item_to_find Then
      found = True
   Else
      index = index + 1
   End If
Loop
If found = True Then
   Console.WriteLine("Item found at position " & index)
Else
   Console.WriteLine("Item not found")
End If
Console.ReadLine()
```