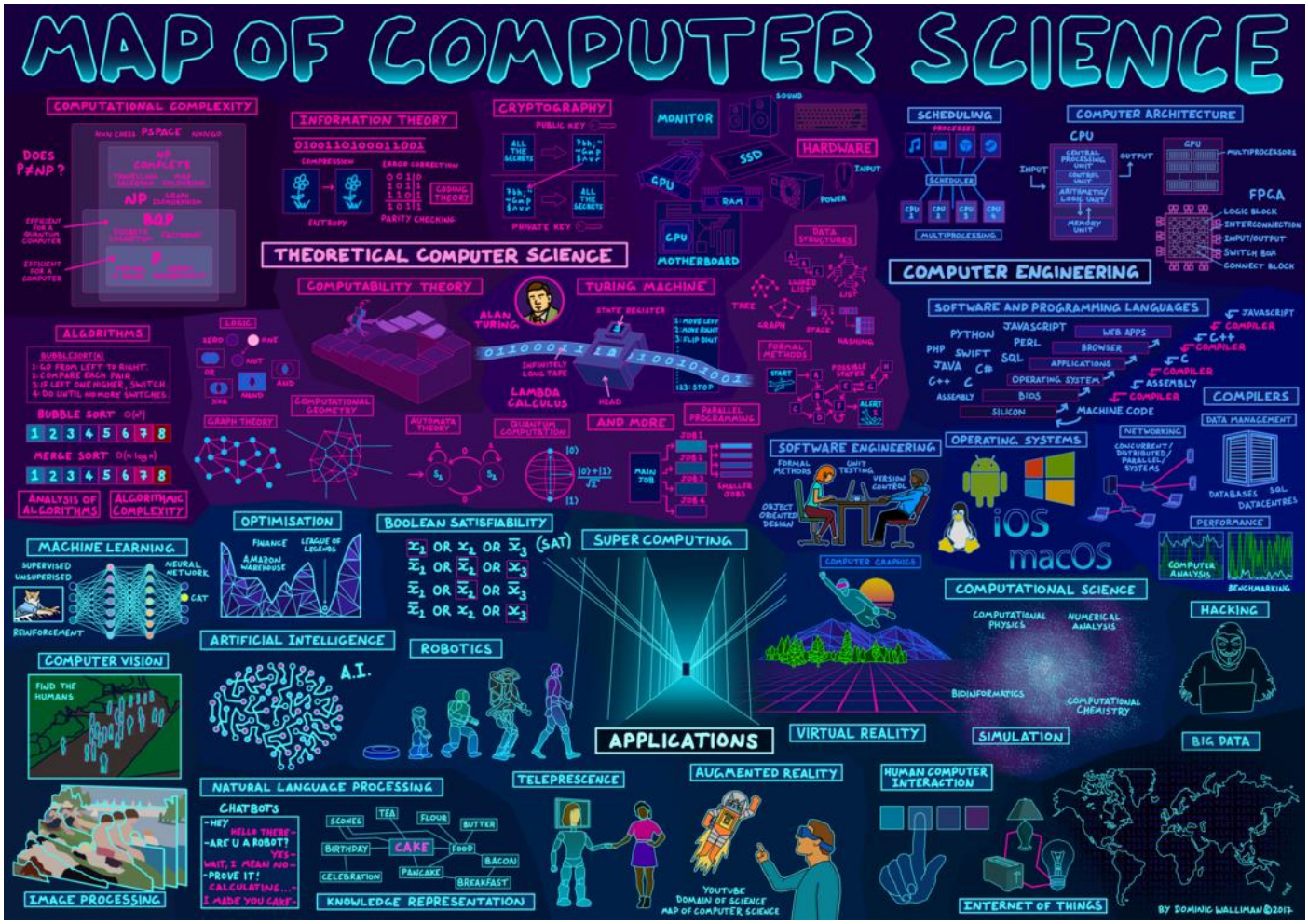
TRANSITION BOOKLET

SIXTH FORM BOLDER

Computer Science

OCR A level H466

Specification : https://www.ocr.org.uk/Images/170844-specification-accredited-a-level-gce-computer-science-h446.pdf



Transition to Sixth Form:

Sixth Form Study

You are likely to study 3 subjects at Bolder Sixth form. Each subject will have six lessons per week. You can expect to engage in a wider range of learning strategies in lessons as well as independently. These could be anything from:

- Making and organising presentations.
- Seminar style reading and group work.
- Use of debate, discussion-based learning, TED-talks, and documentaries.
- Wider reading outside of lesson hours.
- Extended 1-1 practice of practical or experimental work.
- Flip learning – learning in your own time and presenting what you have found to the class.

Independent Study

A Levels and Applied Qualifications will require more study to be completed by you independently rather than with a teacher. At Bolder we recommend that you spend the same time studying outside of lessons as you do in lessons. Therefore, if you have 6 hours of Biology per week, this means that 6 hours should be spent revisiting notes, revising content, completing practice questions outside of the classroom each week also.

This pack will support you with starting to practice independent study over the summer period which will help you understand what works best for you.

What do you need to complete?

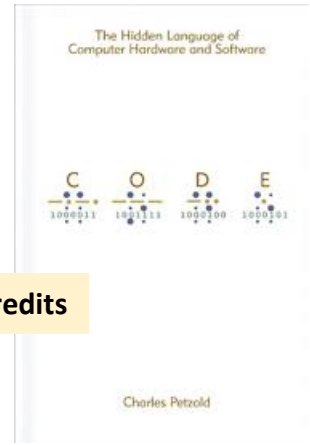
Over the summer it is expected that you engage with this transition booklet to support with your movement into A level Computer Science.

You must make a total of 200 credits through the summer.

The points for each task are listed on the relevant page.

You can decide what combination of tasks to complete, but this must add up to 200 credits. These must also be evidenced on your return in September, preferably hand written though code may be evidenced in screenshots.

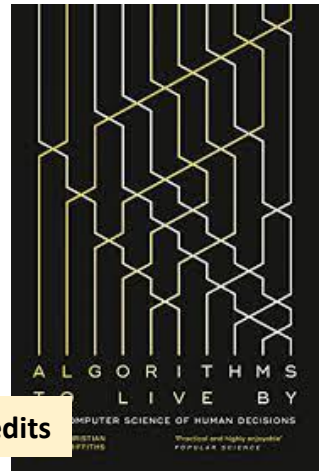
Media Recommendations:



50 Credits

It's a cleverly illustrated and eminently comprehensible story—and along the way, you'll discover you've gained a real context for understanding today's world of PCs, digital media, and the Internet.

Computers, like us, confront limited space and time, so computer scientists have been grappling with similar problems for decades. And the solutions they've found have much to teach us.



50 Credits

25 Credits

TechStuff is a show about technology. And it's not just how technology works. Join host Jonathan Strickland as he explores the people behind the tech, the companies that market it and how technology affects our lives and culture.

75 Credits

Carrie Anne Philbin teaches you computer science! This course covers the history of computing, basic programming structures, elements of hardware and software and much more.

Check this series out on YouTube: shorturl.at/AEU07

Why is Computer Science Important?

25 Credits

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

Computational Thinking

25 Credits

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. At GCSE you will have learned these as abstraction, decomposition, algorithms and pattern recognition. At A level we need to add more detail to these concepts.

1. Review the computational thinking mats below and/or do your own research. Particularly note the different names for the concepts.
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?

Computational Thinking

A LEVEL COMPUTER SCIENCE

Aspect	Exam board definition	Meaning	Theoretical Positives/Negatives	Simple Illustrations
Thinking abstractly	Removing unnecessary details and including only the relevant details.	Identifying what does and doesn't matter to solving the problem. The idea of layering or levels of a problem. Deciding what variables & objects will be needed.	+ Allows you to make predictions. - It may be difficult to predict markets, users, trends and technical influences. - Too many variables may mean the scenario is too complex to model accurately.	Symbols on a map showing buildings, roads etc. Charting where an Oyster card is checked in and out on the London Underground. 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.

Computational Methods

A LEVEL COMPUTER SCIENCE

Aspect	Exam board definition	Meaning	Theoretical Positives/Negatives	Simple Illustrations
Problem recognition	Knowing what the problem is.	Identifying the key requirements of a solution.	- Not all problems can easily be solved by a computer.	Identifying the requirements of a mortgage calculator.
Backtracking	Going back to a previously successful match to find another solution.	Trying an alternative approach if needed.	+ Good for solving logic problems and providing artificial intelligence algorithms. - Only useful for sequential problems.	Mobile phone won't send an email. Going back to check Wi-Fi is enabled, and the internet connection.
Data Mining	The analysis of a large amount of data to provide new information.	Looking for deeper meanings, not obvious conclusions in available data.	+ Advantage can be gained if you can spot unexpected trends and patterns. + Anomaly detection.	Working out that nappies are often bought by men. Putting beer next to nappies increases the sale of beer.
Heuristics	A best guess to problem solving to reduce computation time.	Approximating solutions to ensure a balance between time spent on solving the problem and getting to the best possible solution.	The best solution may take too long, or be too expensive to achieve. Sometimes a solution that may not be the best, works out OK.	Estimating congestion when route planning.
Performance modelling	Carrying out mathematical analysis to assess efficiency.	Knowing how well a solution will perform before full implementation. Building models to test scenarios.	+ Simulations predict outcomes. + Cost effective, time saving & safety first approach. - Requires accurate data. Statistics (relevant data) is used to build the model. Randomisation may be needed to model uncertainty.	Not testing a new exam system on the day exam results are due to be published. Not testing a new London Underground system for train management during rush hour.
Pipelining	The output of one process is the input to another. Queuing up processes.	Some processes must be achieved one after another.	+ Can speed up the execution. - Decisions & branches can mean the pipeline has to be reset, as the next process is no longer the one to be done next.	Mix the cake ingredients. Bake the cake. Let it cool. Put on the icing.
Visualisation	Visualisation is a representation of reality using symbols, charts and colour.	Using diagrams to represent data for analysis. Modelling scenarios and comparing to visible reality.	+ Data is more easy to read if it is presented in a visual way. + May be easier to spot trends, patterns and relationships between different items of data.	Using diagrams to represent programs: systems diagrams, class diagrams and flowcharts.

Getting ahead!

In A level Computer Science you will need to build on your knowledge from GCSE, particularly in the area of programming. We will continue to use Python as our main programming language at A-level. Here are some tasks you can use to help remind yourself, and to get ahead for September.

Ethical, Moral and Cultural Issues

25 Credits

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 cars
- 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

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

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

Terminology

25 Credits

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 from 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?

Programming basics **50 Credits**

Learning to "code" is a fun and essential part of A Level Computer Science. Want a nice refresher ahead of starting your A Level course. If you are already a strong programmer, you may find these tasks too simple, in which case, try the later tutorials.

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/

Challenge

Write out python code 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" and ask the user if they would like to add the item to the array.

Representing negative numbers in Binary **25 Credits**

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 how to represent positive whole numbers between 0-255 in binary.

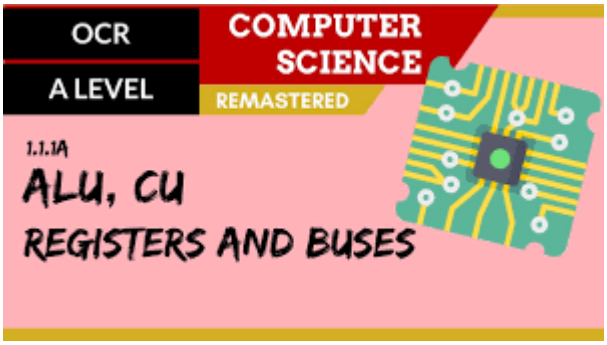
Next, research how to represent negative numbers in binary using the method known as Two's complement. Complete the below task.

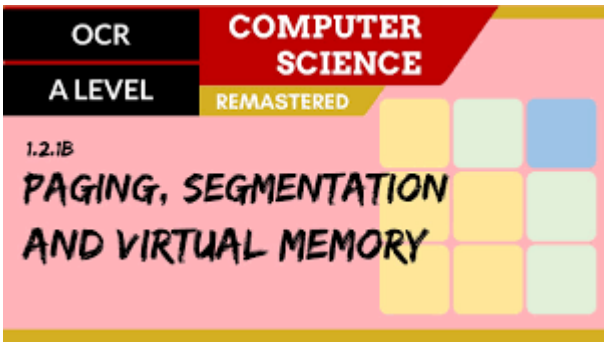
Convert the following into binary using Two's complement: -107, -47, -12
Given you have 8 bits, what is the lowest possible negative number you could represent?

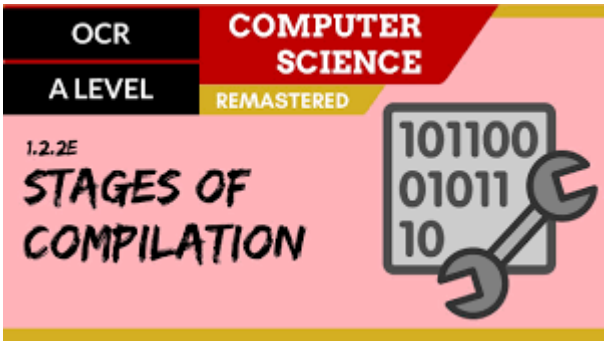
Note taking

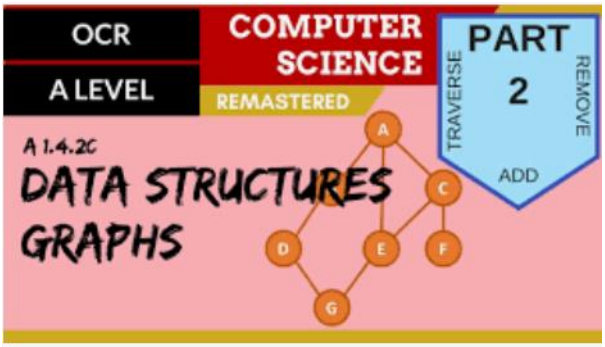
20 Credits each

Use each of the videos linked below to produce one page of Cornell style notes. If you need some guidance on how to take Cornell style notes, check out craigndave.org/cornell-note-taking









Network Design

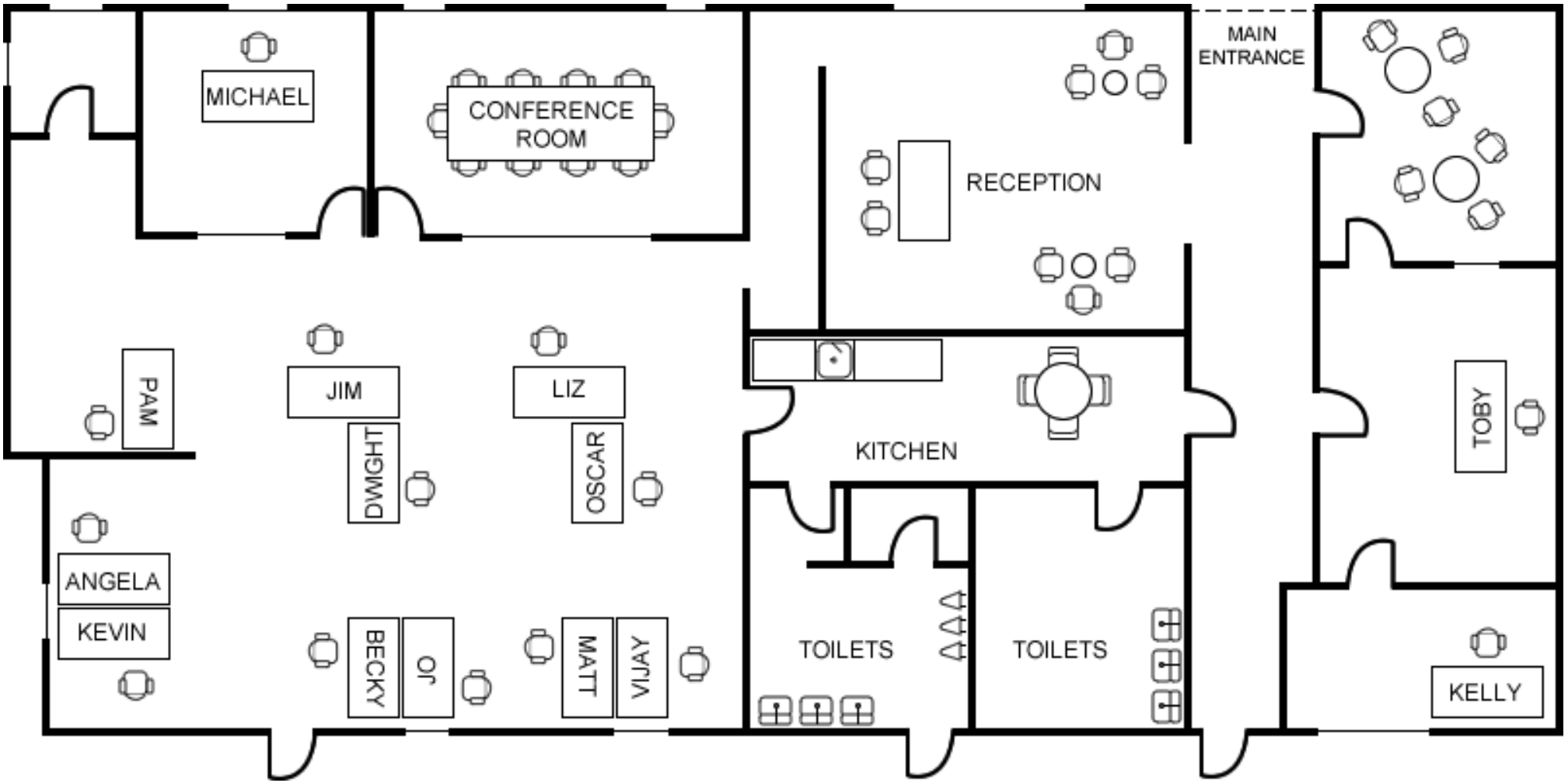
30 Credits

Carry out some research on computer networks, in particular LANs, WiFi, Network topologies and connectivity devices.

Use the symbols on the right (or draw your own) to create an appropriate network over the floorplan on the next page.

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


MICHAEL
CONFERENCE ROOM
MAIN ENTRANCE
RECEPTION
PAM
JIM
LIZ
DWIGHT
OSCAR
KITCHEN
TOBY
ANGELA
KEVIN
BECKY
JO
MATT
VIJAY
TOILETS
TOILETS
KELLY