Y9 revision checklist, last updated February 2015

This document is intended to give a list of the main areas of knowledge from y9, and hence to be useful both as you work your way through the year and also for end of year revision.

Please note that Y9 marks the end of KS3 and that you are also expected to be aware of material already assessed in Y7 and Y8.

Sorting/Algorithms

- Understand what an algorithm is
- Understand the importance of accuracy in creating algorithms
- Know at least one algorithm for sorting
- Know at least one algorithm for searching
- Be able to describe advantages of sorting data in advance of searching
- Understand that different algorithms may be available for the same problem, and be able to identify reasons why one might be better than another in a given situation
- Know what a binary tree is, and how algorithms like this allow computers to deal with large volumes of data
- Understand that algorithms are used by many organisations and be able to describe in basic terms how Google's PageRank algorithm works
- Know the difference between alpha testing and beta testing

Thymio II robots

- Be aware of 'event driven' programming (i.e. how 'onevent' is used in Aseba Studio) and how it can be useful in some circumstances
- Most of the work with the robots was designed to consolidate the ideas listed under 'Python' above.

Sound & video data

- Be able to discuss the benefits of multi-tracking when recording music including adding effects, rebalancing/re-mixing music, etc.
- Be able to define analogue and digital music, and the advantages/disadvantages of digitizing music
- Be able to explain the differences between the two main digital music formats: sampled music and MIDI
- Be able to describe how music is digitised and the purpose of DAC and ADC devices
- Be able to define and use digital music terms including sampling rates, frequency, amplitude and Hertz
- Be able to describe the differences and relative advantages of vector and raster graphics formats

- Be able to explain vector graphics, including understanding the terms ‘primative’ & ‘Besier curve’ and listing some properties of objects that might be stored
- Be able to describe at least one technique for reducing the ‘pixelation’ effect that occurs when enlarging a raster graphic (e.g. anti-aliasing)
- Understand the difference between additive and subtractive primary colours and why both systems are used with computers
- Know of the CMYK system used for printing
- Understand what ‘alpha’ represents, and be able to list at least one common file format that supports transparency
- Understand that the size of a graphic file is linked to both the resolution of an image and its colour depth (i.e. how many colours something could be), and be able to explain why this is
- Be able to discuss file compression, including ‘lossy’ vs ‘lossless’ compression and the meaning of the term ‘redundant data’ in this context
- Be able to describe at least one compression technique (e.g. run length encoding)

Python

- Be able to describe/define:
    - Escape character
    - Syntax
    - Variable
    - Constant
    - Assignment operator
    - Arithmetic operators
    - Relational/Conditional operators
    - Looping/iteration
    - Selection (IF)
- Be able to define and distinguish between integers, floats, Boolean variables and strings
- Be able to explain why comments are useful?

Data protection act

- Be able to describe/define:
    - data commissioner
    - data subject
    - data controller
    - data user
    - personal data
- Be able to recognise the principles of the DPA and to list at least three of them

Computer Misuse Act

- Be able to describe the three levels of misuse

Copyright, Designs & Patents Act

- Be able to describe the reasons for this legislation.
- Be able to define plagiarism.

Regulation of Investigatory Powers Act

- Be able to briefly describe the purpose of the legislation, and what is covered by it.

Advanced spreadsheet work

- Be able to format cells, including use of conditional formatting AND to be able to provide examples of when/why this would be useful
- Be able to define what a function is, and discuss and use functions including IF, VLOOKUP, COUNT & RANK.
- Be able to use absolute cell references and/or cell naming techniques where appropriate
- Be able to design 'user friendly' spreadsheets including the use of a range of mouse-activated 'widgets' (such as command buttons, checkboxes, drop down lists, etc), tooltips….
- Be able to define validation & verification, and to be able to use validation in spreadsheets
- Be able to describe at least one use of a macro, and to be able to create a macro in Microsoft Access and run it using a command button
- Be able to apply a range of spreadsheet techniques – i.e. faced with a real problem be able to choose and justify the use of appropriate techniques
- Be able to justify the choice of a particular graph/chart type, and create this

Algorithms / SatNav

- Be able to describe how GPS works (triangulation may be used interchangeably with trilateration)
- Be able to define abstraction, give at least one example, and produce an abstracted diagram/map from real data as an aid to problem solving.
- Be able to follow Dijkstra's algorithm to solve computational problems (must be able to show evidence of working)
- Be able to explain the benefits of an algorithmic approach to problem solving in terms of:
    - A, Taking a complex problem, simplifying it and developing an algorithmic solution that can then be 'scaled up' again to solve complex problems
    - B, Being suited to processing by computers. Why are computers good at solving these sort of problems?