Year 8 - Computing

Areas of Learning

- **Algorithms:** Consider aspects of computational thinking: problem decomposition, logical reasoning, pattern recognition, abstraction, pattern generalisation and algorithm design.
- **Programming & Development:** Use at least one textual based programming language to solve a variety of computational problems, using a variety of programming constructs, logic and Boolean operators. Use logical reasoning to predict the behavior of a program, to analyse its performance and to debug logical (semantic), syntax (syntactical) and run-time errors.
- **Data & Data Representation:** Understand how computers store and manipulate data of different forms.
- **Hardware & Processing:** Understand the main functional units of a computer system, how data is generated, stored, processed by a computer, and how its output can be used. Understand the low-level architecture of computer systems, to appreciate how some of the main computer modules function and how the hardware enables such functions.
- **Communication & Networks:** Appreciate the structure and operation of data communication networks, and how the internet works.
- **Information Technology:** Use standard tools to create and repurpose digital content. Use available tools to search for content and to judge content carefully in terms of its reliability.

Approaches to learning

- Use online services, such as Codecademy, that guide learners step-by-step, with interactive feedback, that allow them to learn a programming language.
- To undertake individual and team-based creative projects:
    - To build skills with common software packages that allow pupils to create and repurpose digital content.
    - To work on problem solving activities that promote the use and development of computational thinking and program development skills, within the context of cross-curricular topics and possibly related to real-world applications.
    - To use robotics and other electronic equipment to demonstrate the use of computer technology in measurement and control applications.
    - To simulate digital logic circuits in order to appreciate how circuitry in a computer operates.
    - To collection information and analyse it.
- Build simulations to demonstrate their understanding of a particular topic, such as the operation of a digital adder circuit.
- Document project work using Microsoft Word, process data using Microsoft Excel, and create presentations using Microsoft Powerpoint.
- To create poster displays of their work.
- Exercises and projects aimed at involving pupils in computational thinking.

Examples of learning

- Pupils develop computer programs that demonstrate alternative sorting algorithms, monitor their performance, analyse results, consider scaling equations for the algorithms and compare algorithmic efficiencies.
- Pupils build poster displays on key areas of learning, particularly those that they have built computer simulations for.

References

- P. Kemp, (2014), "Computing in the national curriculum: A guide for secondary teachers", ISBN: 978-1-78339-376-3.
- M. Dorling, et.al, (2014), "Computing progression pathways", Computing at School, download accessed on 07-07-2015 at www.computingatschool.org.uk
- UK Department of Education, (2013), National Curriculum Documents available at: https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study#key-stage-2 , web-document published on 11 September 2013.