Y10 Computer Science January Revision Topics

- 1 hour 30 min Paper based Exam
- 1 hour 30 min Practical Programming Assessment in Computer Suite

Each lesson has an assignment on Teams which includes a PowerPoint and video to use as revision material.

Videos can be found here: https://student.craigndave.org/1CP2

Exam Questions and Quizzes here: https://www.bbc.co.uk/bitesize/examspecs/zdqy7nb

Remember it's the Pearson Edexcel GCSE (1CP2) Qualification NOT OCR!

Computational Thinking:

- Define the term 'program'.
- Access an Integrated Development Environment (IDE) to work with a Python program.
- Create simple Python programs that use different data types and arithmetic operators.
- Define the terms 'decomposition' and 'algorithm'.
- Define the term 'sequence' and use sequence in algorithms and program code.
- Interpret error messages.
- Create variables and assign them with correct datatype, View contents of memory (variable) in an IDE.
- Take input and create output.
- Find and fix errors and define different type of errors.
- Translate code into flowchart symbols.
- Represent an algorithm in a flowchart.
- Translate a flowchart into code.
- Use relational operators like while, and 'if' 'else' in flowchart and code.
- Use flowcharts to represent selection and repetition.
- Identify parts of a program (variables, constants, selection, repetition).
- Solve problems using code.
- Use repetition in code.
- Use selection in code.
- Define and construct truth tables for Boolean Operators and combinations.

Programming:

- Define the term 'digital computer' and examples of different types of computers.
- Define 'binary' and 'bit' and explain their relationship.
- Define 'nibble' and 'byte'.
- Convert denary and 8-bit binary numbers.
- Add two 8-bit binary numbers and describe the effects of overflow error.
- Apply a logical binary shift left to a positive 8-bit binary number.
- Apply a logical shift right to a positive 8-bit binary number.
- Explain why binary numbers may become less accurate after a binary shift right.
- Apply an arithmetic binary shift right to a two's complement number.
- Give the hexadecimal equivalent of an 8-bit binary number.
- Give the binary equivalent of a hexadecimal number.
- Explain why hexadecimal is used.
- Describe how characters are encoded in ASCII.
- Derive the code for an ASCII character from that of another.
- Describe the limitations of ASCII.
- Differences between signed and unsigned denary numbers and their conversion to binary