



The English Martyrs Catholic School and Sixth Form College

<u>Y13 Comp Sci</u>	<u>Module 1</u>	<u>Module 2</u>	<u>Module 3</u>
<u>Topic Theme and Intent</u>	Further develop knowledge of software development including compilation and memory addressing. Understand OOP paradigms and the application of these and how to break down a problem using abstraction and decomposition .	Understand network security and the use of search engine indexing, page rank, peer-to-peer and client-server models. Further develop knowledge of data types and structures and complete development of a programming project .	Further develop skill in converting floating point form and be able to carry out bitwise manipulation. Understand how to approach computational problems and the use of key algorithms to embed knowledge of computational concepts.
<u>Knowledge</u>	<ul style="list-style-type: none"> Software and software development, exchanging data and characteristics of contemporary processors. Problem solving and programming, analysis of a problem and design of a solution. 	<ul style="list-style-type: none"> Exchanging data and the data types, structures and algorithms used in computer systems. Design, create and test a solution to a problem using a structured test plan. 	<ul style="list-style-type: none"> Data types and structures and the legal and ethical issues involved in computer science. Computational thinking elements, problem solving and programming solutions to a problem.
<u>Skills</u>	Identify memory addressing methods and use of OOP including compression, encryption and hashing . Software development using C# , decomposing a problem and specifying a solution.	Use of databases, networks and web technologies used for communication and the manipulation of data including floating point values. Complete and test a project developed in C#.	Use of Boolean algebra , particularly around negative exponent and mantissa in floating point form . Understanding of key algorithms including A* algorithm, Dijkstra's and big O notation.
<u>Literacy Links</u>	<p>Reading – Analyse context of questions and carry out software research.</p> <p>Writing – Use subject specific terminology in identifying solutions to a problem.</p> <p>Oracy – Class discussion around key issues, group presentation.</p>	<p>Reading – Carry out research tasks and utilise findings in the project.</p> <p>Writing – Complete testing and evaluate programming project.</p> <p>Oracy – Debate key issues during class discussion and present arguments.</p>	<p>Reading – Read and follow through algorithms taking the correct path.</p> <p>Writing – Extended response questions and use of subject specific terminology.</p> <p>Oracy – Presenting work to a class for feedback and legal discussion.</p>
<u>Essential Vocabulary</u>	Compilation, Dependency, Inheritance, Encapsulation, Normalisation, Paradigm	API, Exponent, Mantissa, Metatag, NSB, Noramaised, Page Rank, VPN, WAP	Absorption, Association, Concatenation, Distribution, Floating Point, Logical Shift

Disciplinary Reading

GCHQ Challenge Book



Reading for Pleasure

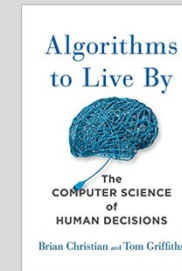
Life 3.0



Crypto Wars



Algorithms to Live By



Outnumbered

