CS KS5 Curriculum

Introduction

Studying Computer Science at A Level at Judd focuses on real-world application through theory and practical elements, using emerging technologies to identify approaches and impact, whilst maintaining a strong foundation in the core principles of computer science.

Computational thinking is the mindset we imprint on students to best help them analyse and understand problems to then design, develop, validate and evaluate conceptual and practical solutions to computational problems.

Students emerge at the end of their studies as well-rounded computational thinkers able to take on the problems in front of them; whiteboarding, prototyping and developing future-proofed solutions geared for the real-world.

Examination Board
OCR H446 Computer Science

A-level Specification
OCR H446 Computer Science Specification

Non-Examination Assessment (20%)
OCR H446 Computer Science Component 03

Year 12

Topic	Description
1.4.x - Data types, data structures & algorithms(1) 2.1.x - Elements of computational thinking	Students will walk through the theory behind the more abstract data-types, advanced data structures and the algorithms that are used to interrogate them. Extending on the logical foundations in Key Stage 3 and 4, students will learn the theory and computational computing behind logic and logic circuits. Being a computer scientist is more than just learning to program, students need to understand how computer systems and software are used and in what environments and situations, and how best to approach problems that

	have computational solutions.
1.1.x - Characteristics of contemporary processors 2.2.x - Problem solving & programming(1) 3.x.x - NEA idea generation	Extending on the knowledge gained at GCSE, students will get deeper under the bonnet of the CPU, the way it processes instructions and how tasks are managed and scheduled, along with the most efficient ways to complete these operations. Students will also gain an understanding of the different types of processors, the ways they can process tasks and what functions they best suit. In preparation for the NEA, students will be introduced to the Object Oriented programming paradigm and learn to apply computational thinking and methods to help solve problems. Students will be tasked with identifying a problem with a computational solution that they would like to complete for Component 03 - NEA.
1.2.x - Exchanging data(1) 2.3.x - Algorithms(1) 3.1.x - NEA Analysis	Students will extend their knowledge around encryption and compression whilst being introduced to the concept and purpose of hashing and hashmaps. Databases are used in almost every digital service we interact with, so students will get to know the theory behind databases and database design. Students will also complete some practical programming exercises around database creation and interaction. Students will be introduced to the development and efficiency side of algorithms; ones they can choose to use for their NEA projects. Having chosen an idea in the previous term, students will now go about the task of analysing the problem: breaking it down to understand its individual components, interacting with stakeholders and identifying features from market research.
1.3.x - Software & software development(1) 2.3.x - Algorithms(2) 3.2.x - NEA Design	Students will be introduced to programming and project methodologies and understand what approach works best depending on the problem to solve. Students will further develop their understanding of programming and programming techniques through use of the languages and encoding in the abstraction stack. Students will practise the design and development of algorithms best suited to solve a particular problem with time and space (efficiency) being a key motivator. From the output of the NEA Analysis, students will start the

	process of designing their solutions; from decomposing the problem, wireframes and pseudocode through feature identification to test data.
1.2.x - Exchanging data(2) 2.2.x - Problem solving & programming(2) 3.3.x - NEA Development	Returning to Exchanging Data, students will cover the principles of networking and understand the way in which they are constructed. They will develop an understanding of the various setups and topologies, the way data is transmitted and how networks can be protected. Students will recover and extend upon their understanding of the 3 constructs: Sequence, Selection (Branching) and Iteration, robust programming through modularisation and validation, and where to use recursion over interaction. Students will also start the Development phase of their NEA projects using prototyping to experiment with languages, technologies and methods of implementation.
1.3.x - Software & software development(2) 2.2.x - Problem solving & programming(3) 3.3.x - NEA Development	Students will look at the generation of applications through the 5 stages of compilation (downwards through the abstraction stack) and get to grips with the various concepts and technologies at each stage. Students will then complement their work on the structure and function of processors by familiarising themselves with the systems software and processes used to. Moving on from NEA prototyping, students will now put into practice all that which they have learnt from the theory and practical elements of the course accompanying the lessons learnt from the prototyping phase.

Year 13

Topic	Description
1.5.x - Legal, moral, cultural & ethical issues 3.3.x - NEA Development	Students will deep dive into the legislation, ethics, environment and cultural impacts of the use of technology in a variety of case studies, scenarios and concepts. They will look at the various approaches of adoption of technologies like Artificial Intelligence and Distributed Ledger Technology (eg. Blockchain & Cryptocurrency) in a plethora of industries identifying the positive and negative approaches of each. Students will complete the development of their NEA projects

	and move into final testing to sign-off the Development phase ready for the Evaluation and reflection.
1.4.x - Data types, data structures & algorithms(2) 3.4.x NEA Evaluation & submission	Students will revisit their theory work on data-types, data structures and the algorithms used to interrogate them. Students will reaffirm their understanding of logic and the computational processes behind the complex operations performed by the Central Processing Unit, along with some programmable exercises to simulated adders and other logic circuits. Students will finish off their NEA project by completing usability testing, User Acceptance testing and evaluating their final solution before submitting at the end of the term.
OCR H446 Component 01 & Component 02 revisited Exam technique inc. extended writing	Students will run back through both components of the A Level specification, revisiting keywords and terms, and core subject knowledge. A number of different summative and formative assessment techniques are employed to achieve this allowing students to identify gaps or weaknesses in their subject knowledge and understanding.
Past Paper questions & revision	To finish off the year, students will be guided through a variety of past paper exam questions and their corresponding mark schemes to build confidence and reaffirm examination board expectations.