

## Key Stage 5 Subject Timeline

Subject: Computer Science

Exam Board: OCR

Year 12						
	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Topics	Topic 1.1: The Processor Topic 1.2: Logical thinking	Topic 2.1: Input, output, storage Topic 2.2: Procedural thinking	Topic 3.1: Systems software Topic 3.2: Programming techniques	Topic 4.1: Application generation Topic 4.2: Computational methods	Topic 5.1: Software generation Topic 5.2: Project planning	Topic 6.1: Programming languages Topic 6.2: Programming project
Key skills and Concepts	<ul style="list-style-type: none"> <li>- Role of the special registers, including within the FDE cycle.</li> <li>- Factors affecting CPU performance.</li> <li>- Use of pipelining to improve efficiency.</li> <li>- Determination of logical conditions that affect the outcome of decisions.</li> <li>- The nature, benefits and drawbacks of caching.</li> <li>- Understand the need for reusable program components.</li> </ul>	<ul style="list-style-type: none"> <li>- Features, pros and cons of different options for input/output/storage.</li> <li>- Features of and need for RAM, ROM and virtual storage.</li> <li>- Identification of the steps and sub-procedures necessary to solve a problem.</li> <li>- The need for abstraction.</li> <li>- The pros and cons of concurrent processing.</li> </ul>	<ul style="list-style-type: none"> <li>- Features, pros and cons of different forms of memory management.</li> <li>- Role of interrupts and service routines in FDE cycle.</li> <li>- Approaches to scheduling in operating systems.</li> <li>- Role of BIOS, device drivers and virtual machines.</li> <li>- Programming using a modular approach.</li> <li>- Use of functions, procedures and parameters.</li> <li>- Iterative vs recursive solutions.</li> <li>- Effective use of a range of IDE feature.</li> </ul>	<ul style="list-style-type: none"> <li>- Open- and closed-source applications.</li> <li>- Translators, assemblers and compilers.</li> <li>- The four stages of compilation.</li> <li>- Linkers, loaders and libraries.</li> <li>- Features of a problem which make it solvable by computational methods.</li> <li>- Problem solving through decomposition and divide and conquer.</li> <li>- The role of backtracking, data mining and heuristics in problem solving.</li> <li>- The role of performance modelling, pipelining and visualisation in problem solving.</li> </ul>	<ul style="list-style-type: none"> <li>- Waterfall vs agile project methodologies.</li> <li>- Features of extreme programming, spiral and RAD methodologies.</li> <li>- Production of effective algorithms to aid software development.</li> <li>- Development of simple web databases through SQL and PHP.</li> <li>- Application of computational methods to the creation of a GUI registration system.</li> </ul>	<ul style="list-style-type: none"> <li>- The need for and characteristics of a range of programming paradigms.</li> <li>- LMC assembly language.</li> <li>- Features, benefits and programming with OOP.</li> <li>- Identification of problem to be solved in programming project.</li> <li>- Meeting the requirements of the Analysis section of the Programming Project.</li> </ul>
Threshold Concepts	Students need to understand that the CPU carries out the processing of instructions and data within a computer system.	Building on the previous half term, students will need to have a clear understanding of how the CPU processes data and instructions. This will help them to understand how data is managed and stored within computer systems. Their work on logical thinking last half term will be extended and deepened in their work on procedural thinking in this unit.	Students will need to have a good understanding of how the CPU manages data and how this is stored in a computer system. The ability to understand how to tackle problems logically and procedurally is required as students begin to tackle increasingly challenging programming tasks.	A strong foundation in programming skills and the generation of simple applications is essential to support the work throughout this unit.	This module extends previous work on logical thinking, programming and application generation into more formal project planning for the generation of more substantial software.	Programming skills and project management knowledge required to start tackling the requirements of the Programming Project (20% of final grade).

## Key Stage 5 Subject Timeline

**Subject: Computer Science**

**Exam Board: OCR**

<b>Endpoints</b>	<p>Students able to describe the role of all components of a CPU.</p> <p>Students demonstrating an ability to think logically in order to tackle problems.</p>	<p>Students can select and justify appropriate input, output and storage devices to suit specific requirements.</p> <p>Students able to combine logical and procedural thinking to design solutions to increasingly challenging problems.</p>	<p>Students able to describe the role of systems software in managing a computer system.</p> <p>Students able to work with increasing independence to create modular programming solutions.</p>	<p>Students able to explain and justify different options for producing and releasing software, using experience to describe the pros and cons of each.</p> <p>Students able to consider an increasing range of computational approaches to solve problems.</p>	<p>Students have developed the project management skills, along with some more advanced programming skills, to enable them to make a positive start to the Programming Project.</p>	<p>Students have developed experience of programming using a range of paradigms.</p> <p>Students have made a strong start to the Programming Project and are ready to make further progress throughout the summer holiday.</p>
<b>Assessment</b>	<p>In-class assessments based on past exam questions.</p> <p>Problem solving challenges to assess practical skills.</p>	<p>In-class assessments based on past exam questions.</p> <p>Problem solving challenges to assess practical skills.</p> <p>PPEs.</p>	<p>In-class assessments based on past exam questions.</p> <p>Problem solving challenges to assess practical skills.</p>	<p>In-class assessments based on past exam questions.</p> <p>Problem solving challenges to assess practical skills.</p>	<p>In-class assessments based on past exam questions.</p> <p>Problem solving challenges to assess practical skills.</p> <p>PPEs.</p>	<p>In-class assessments based on past exam questions.</p> <p>Problem solving challenges to assess practical skills.</p> <p>Assessment of Analysis section of Programming Project.</p>

## Key Stage 5 Subject Timeline

Subject: Computer Science

Exam Board: OCR

Year 13						
	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Topics	Topic 1.1: Compression and databases  Topic 1.2: Programming project development	Topic 2.1: Networks and internet  Topic 2.2: Algorithms part 1	Topic 3.1: Boolean algebra  Topic 3.2: Algorithms part 2	Topic 4: Revision	Topic 5: Final exam prep	
Key skills and Concepts	<ul style="list-style-type: none"> <li>- Features, pros and cons of different forms of compression.</li> <li>- Encryption and hashing.</li> <li>- Flat file and relational databases.</li> <li>- Normalisation of databases.</li> <li>- ACID processing.</li> <li>- Development of programming project.</li> </ul>	<ul style="list-style-type: none"> <li>- Network and internet protocols.</li> <li>- Features, pros and cons of different network structures, and the hardware needed in each.</li> <li>- Programming in HTML, CSS and JavaScript.</li> <li>- Common algorithms including:                             <ul style="list-style-type: none"> <li>- Sorting algorithms</li> <li>- Searching algorithms</li> <li>- Dijkstra's shortest path</li> <li>- The A* algorithm</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>- Definition of problems using Boolean logic.</li> <li>- Interpretation and creation of logic gate diagrams and truth tables.</li> <li>- Simplifying Boolean expressions using Karnaugh maps and Boolean Laws.</li> <li>- Logic associated with flip-flops and adders.</li> <li>- Algorithms for the key data structures (stacks, queues, trees and linked lists).</li> <li>- Evaluating the efficiency of algorithms using Big O notation.</li> <li>- Ability to compare the complexity of algorithms.</li> </ul>	Revision program based on individual and whole-class needs identified through in-class assessments and PPEs.	Revision program based on individual and whole-class needs identified through in-class assessments and PPEs.	
Threshold Concepts	Students need an understanding of the storage of data before working on compression. Programming project work draws on programming and project development knowledge developed throughout Y12.	Knowledge of logical and procedural thinking (developed in Y12) is essential to understanding the key algorithms. Programming skills will be required to create practical applications of a selection of these algorithms.	Students will need a good understanding of common algorithms (covered in previous half term) in order to be able to learn to evaluate them for efficiency and complexity.	Entire curriculum should have been covered at this point.	Entire curriculum should have been covered at this point.	

## Key Stage 5 Subject Timeline

**Subject: Computer Science**

**Exam Board: OCR**

<b>Endpoints</b>	Students demonstrate success in addressing past exam questions on compression and databases. Students have completed a prototype solution to the problem identified in their Programming Project.	Students demonstrate success in addressing past exam questions on algorithms, networks and the internet. Students have completed at least a full draft of their Programming Project.		Students have worked towards developing knowledge in areas of identified weakness. Evidenced by improving performance in practice examinations and assessments.	Students fully prepared to meet and exceed target grades in final examinations.	
<b>Assessment</b>	In-class assessments based on past exam questions.  Problem solving challenges to assess practical skills.	In-class assessments based on past exam questions.  Problem solving challenges to assess practical skills, applied to style of questions faced in the Component 2 exam.  PPEs.	In-class assessments based on past exam questions.  Problem solving challenges to assess practical skills, applied to style of questions faced in the Component 2 exam.	In-class assessments based on past exam questions.  Problem solving challenges to assess practical skills, applied to style of questions faced in the Component 2 exam.  PPEs.	In-class assessments based on past exam questions.  Problem solving challenges to assess practical skills, applied to style of questions faced in the Component 2 exam.	