

## Key Stage 5: Year 12 Computer Science

### Overall Curriculum Goals

- To build on the knowledge, understanding and skills established at key stage 4 and encourage students to develop a broad range of the knowledge, understanding and skills of computing, as a basis for progression into further learning and/or employment.
- Encourage students to develop:
  - an understanding of, and the ability to apply, the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms and data representation
  - the ability to analyse problems in computational terms through practical experience of solving such problems, including writing programs to do so
  - the capacity for thinking creatively, innovatively, analytically, logically and critically
  - the capacity to see relationships between different aspects of computer science
  - mathematical skills related to:
    - Boolean algebra
    - number representations and bases.
  - the ability to articulate the individual (moral), social (ethical), legal and cultural opportunities and risks of digital technology.

Half Term 1	Half Term 2	Half Term 3	Half Term 4	Half Term 5	Half Term 6
Key Vocabulary/Concepts/Ideas	Key Vocabulary/Concepts/Ideas	Key Vocabulary/Concepts/Ideas	Key Vocabulary/Concepts/Ideas	Key Vocabulary/Concepts/Ideas	Key Vocabulary/Concepts/Ideas
<p><b>Programming Concepts</b></p> <ul style="list-style-type: none"> <li>- Introduction</li> <li>- Scanners and types</li> <li>- if; for loops</li> <li>- Test, while and do while</li> <li>- File input and output 1</li> <li>- IO 2</li> <li>- Java Arrays</li> <li>- Classes and OOP</li> <li>- Setters and Getters</li> <li>- Constructors</li> <li>- Test</li> <li>- Progress Grades Exercise</li> </ul> <p><b>Theory of Computation</b></p> <ul style="list-style-type: none"> <li>- Abstraction and Automation</li> <li>- Problem solving</li> </ul> <p><b>Fundamentals of data representation</b></p> <ul style="list-style-type: none"> <li>- Number Systems</li> <li>- Number Bases</li> <li>- Units of Information</li> <li>- Binary Number System</li> </ul>	<p><b>Programming Paradigms &amp; Sub-Routines</b></p> <ul style="list-style-type: none"> <li>- Java GUI: JFrame, Layout Managers</li> <li>- Go over test</li> <li>- JFrame inheritance</li> <li>- Demonstrate containers</li> <li>- Demonstrate ActionListener and creating dist jars</li> <li>- Scope and getText()</li> <li>- Adding Images to JLabels</li> <li>- Multiple Events, HTML</li> <li>- Graphics Contexts and JPanels</li> <li>- Passing Values: traffic lights</li> <li>- Timers: CountFrame and GUIBall</li> <li>- Christmas Tree Challenge</li> <li>- Ditto</li> <li>- Ditto</li> <li>- Christmas Tree Judgement</li> </ul> <p><b>Fundamentals of data representation</b></p> <ul style="list-style-type: none"> <li>- Images, sound and other data</li> <li>- Encryption - Compression</li> </ul> <p><b>Fundamentals of Computer Systems</b></p> <ul style="list-style-type: none"> <li>- Hardware and Software</li> <li>- Classification of programming languages</li> <li>- Types of program translators</li> <li>- Logic Gates</li> <li>- Boolean Algebra</li> </ul>	<p><b>Theory of Computation &amp; Fundamentals of Algorithms</b></p> <ul style="list-style-type: none"> <li>- Pseudocode</li> <li>- Past Paper Questions</li> <li>- Dry Runs</li> <li>- Abstraction and Automation</li> <li>- Start AS PM</li> <li>- FSMs</li> <li>- Skeleton Meet the Code</li> <li>- Turing Machines</li> <li>- Exercise 1</li> <li>- Do 2D arrays</li> <li>- Go over TMS</li> <li>-</li> <li>- Regular and Context Free Exercises</li> <li>- Saving a Field in Skeleton Code</li> </ul> <p><b>Fundamentals of computer organisation and architecture</b></p> <ul style="list-style-type: none"> <li>- Internal hardware components</li> <li>- Stored program concept</li> <li>- Structure of the role of the processor and its components</li> <li>- External hardware devices</li> </ul>	<p><b>Theory of Computation &amp; Fundamentals of Algorithms</b></p> <ul style="list-style-type: none"> <li>- Regular expressions in Java</li> <li>- Seed Exercise 4 and 5</li> <li>- Seed Exercise 5 and 6</li> <li>- Big O Notation</li> <li>- Practice NEA for test next week</li> <li>- Finishing theory</li> <li>- NEA Test , Tuesday and Friday</li> <li>- Return AS Papers</li> <li>- NEA Prep Sheets 1 and 2</li> </ul> <p><b>Consequences of uses of computing</b></p> <ul style="list-style-type: none"> <li>- Individual (moral), social (ethical), legal and cultural issues</li> </ul>	<p><b>Programming Concepts &amp; Programming Paradigms</b></p> <ul style="list-style-type: none"> <li>- Collections</li> <li>- MockSkeleton Meet the code</li> <li>- Exercise 1</li> <li>- Exercise 2</li> <li>- Exercise 3</li> <li>- NEA Prep: database and GUI</li> <li>- Questions on above</li> <li>- Prime Numbers and Dictionary GUI</li> <li>- Bouncy Ball</li> <li>- Ball and Paddle</li> <li>- Ball Classes and Breakout</li> <li>- Object Oriented Programming</li> <li>- Abstract Classes and Interfaces</li> <li>- Database Builders</li> <li>- Choice Interfaces</li> </ul> <p><b>Fundamentals of communication and networking</b></p> <ul style="list-style-type: none"> <li>- Communication basics</li> <li>- Communication methods</li> <li>- Networking Topologies</li> <li>- Types of networking between hosts</li> <li>- Wireless networking</li> </ul> <p><b>Consequences of uses of computing</b></p> <ul style="list-style-type: none"> <li>- Individual (moral), social (ethical), legal and cultural issues</li> </ul>	<p><b>Functional Programming &amp; Fundamentals of Algorithms</b></p> <ul style="list-style-type: none"> <li>- Revision for Finals</li> <li>- Revision for Finals</li> <li>- Haskell</li> <li>- Prime Numbers Question</li> <li>- Haskell Lists, Resits</li> <li>- Recursion</li> <li>- Haskell Recursion</li> <li>- GUI Breakout</li> <li>- Continue GUI Breakout</li> <li>- Project ideas</li> <li>- Making their own 2D games over the summer</li> </ul> <p><b>All Theory Topics</b></p> <ul style="list-style-type: none"> <li>- Revision</li> <li>- Revisit hardest questions</li> </ul>
<b>CIAG</b>	<b>CIAG</b>	<b>CIAG</b>	<b>CIAG</b>	<b>CIAG</b>	<b>CIAG</b>
Guest speaker from industry	BEBRAS Challenge	British Informatics Olympiad	Automated warehouse visit.		

## Key Stage 5: Year 13

### Overall Curriculum Goals

- Encourage students to develop:
  - an understanding of, and the ability to apply, the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms and data representation
  - the ability to analyse problems in computational terms through practical experience of solving such problems, including writing programs to do so • the capacity for thinking creatively, innovatively, analytically, logically and critically
  - the capacity to see relationships between different aspects of computer science
  - mathematical skills related to:
    - Boolean algebra
    - comparison and complexity of algorithms
    - number representations and bases.
  - the ability to articulate the individual (moral), social (ethical), legal and cultural opportunities and risks of digital technology.

Half Term 1	Half Term 2	Half Term 3	Half Term 4	Half Term 5	Half Term 6
Key Vocabulary/Concepts/Ideas	Key Vocabulary/Concepts/Ideas	Key Vocabulary/Concepts/Ideas	Key Vocabulary/Concepts/Ideas	Key Vocabulary/Concepts/Ideas	Key Vocabulary/Concepts/Ideas
<p><b>Fundamentals of Data Structures</b></p> <ul style="list-style-type: none"> <li>- Skeleton Code 1 Meet Code</li> <li>- Simple Parser</li> <li>- Complex Parser</li> <li>- Sockets</li> <li>- Create Server and Client</li> <li>- Skeleton Code: Meet the Code</li> </ul> <p><b>Fundamentals of Databases</b></p> <ul style="list-style-type: none"> <li>- Conceptual data models and entity relationship modelling</li> <li>- Relational databases</li> <li>- Database design and normalisation techniques</li> <li>- Structured Query Language (SQL)</li> <li>- Client Server databases</li> </ul> <p><b>Big Data</b></p> <ul style="list-style-type: none"> <li>- Big data</li> </ul>	<p><b>Programming, Data Structures, &amp; Algorithms</b></p> <ul style="list-style-type: none"> <li>• NEA analysis</li> <li>• NEA Design</li> <li>• Skeleton Code</li> </ul> <p><b>Fundamentals of data representation</b></p> <ul style="list-style-type: none"> <li>- Number Systems</li> <li>- Signed Binary using two's complement</li> <li>- Rounding errors</li> <li>- Absolute and relative errors</li> <li>- Range and precision</li> <li>- Normalisation of floating-point form</li> <li>- Underflow and overflow</li> <li>- Error checking and correction</li> </ul> <p><b>Fundamentals of Algorithm</b></p> <ul style="list-style-type: none"> <li>- Models of computation</li> <li>- Turing machines</li> </ul>	<p><b>Programming, Data Structures, &amp; Algorithms</b></p> <ul style="list-style-type: none"> <li>• NE Implementation</li> <li>• Skeleton Code</li> </ul> <p><b>Fundamentals of Algorithm</b></p> <ul style="list-style-type: none"> <li>- Maths for regular expressions</li> <li>- Regular expressions</li> <li>- Context free languages – Backus Naur form (BNF)</li> <li>- Classification of Algorithms</li> </ul> <p><b>Fundamentals of communication and networking</b></p> <ul style="list-style-type: none"> <li>- The internet and how it works</li> <li>- Internet Security</li> <li>- The Transmission Control Protocol / Internet Protocol (TCP/IP)</li> <li>- Standard application Layers</li> <li>- IP address structure and standards</li> <li>- Subnet masking</li> <li>- Public and private IP addresses</li> <li>- Dynamic Host Configuration Protocol (DHCP)</li> <li>- Network Address Translation (NAT)</li> <li>- Port forwarding</li> <li>- Client server model</li> <li>- Thick vs thin client computing</li> </ul>	<p><b>Programming, Data Structures, &amp; Algorithms</b></p> <ul style="list-style-type: none"> <li>• NEA Implementation</li> <li>• NEA Testing</li> <li>• NEA Evaluation</li> <li>• Skeleton Code</li> </ul> <p><b>Fundamentals of Algorithms</b></p> <ul style="list-style-type: none"> <li>- Graph traversal</li> <li>- Tree traversal</li> <li>- Reverse Polish</li> <li>- Searching Algorithms</li> <li>- Sorting Algorithms</li> <li>- Optimisation Algorithms</li> </ul> <p><b>Consequences of uses of computing</b></p> <ul style="list-style-type: none"> <li>- Individual (moral), social (ethical), legal and cultural issues</li> </ul>	<p><b>Programming, Data Structures, &amp; Algorithms</b></p> <ul style="list-style-type: none"> <li>• Revision</li> <li>• Skeleton Code</li> </ul> <p><b>All theory topics</b></p> <ul style="list-style-type: none"> <li>- Revision</li> <li>- Revisit hardest questions</li> </ul>	
<b>CIAG</b>	<b>CIAG</b>	<b>CIAG</b>	<b>CIAG</b>	<b>CIAG</b>	<b>CIAG</b>
Guest speaker from industry	BEBRAS Challenge	British Informatics Olympiad			