



## AIM AND PURPOSE

Computing at TTA aims to prepare students to become fluent in a variety of digital skills. These skills are becoming increasingly valued and important to a wide range of careers, and in some instances are at a critical shortage (DCMS, 2019). The transformation that digital technology has brought and is still yet to bring will have an enormous effect on everyone's lives (Royal Society, 2016). The purpose of the TTA Computing Curriculum is to equip students with the tools to harness technology, using it to solve problems and to keep up with the changes that are taking place with technology in society every day. Every student should have the opportunity to navigate these technological changes safely, effectively, and be well-informed of the opportunities available to them to become effective digital citizens.

## HOW DOES THE CURRICULUM INDUCT STUDENTS INTO THE DISCIPLINE OF THE SUBJECT?

The curriculum at TTA is designed to provide students with a comprehensive introduction to the field of computer science. The curriculum is structured in such a way that students are gradually exposed to the fundamental concepts and skills required to be successful in the discipline. This is achieved through a series of engaging and interactive lessons that are designed to be both accessible and challenging. In addition to core computer science topics, the curriculum also covers a range of related subjects, such as digital literacy and computational thinking, to provide students with a well-rounded understanding of the field. Moreover, the curriculum also includes a range of practical activities and project-based learning opportunities, which help to bring the theory to life and provide students with hands-on experience in real-world applications of computer science. Through these experiences, students can develop their problem-solving skills, creativity, and ability to think critically about complex issues.

Computer science not only provides students with valuable technical skills, but it also helps to develop their critical thinking, problem-solving, and communication skills, all of which are transferable to many other fields and careers (The Royal Society, 2016). This approach to induction helps to provide students with the knowledge and skills they need to succeed in a rapidly changing digital world, both within and beyond the discipline of computer science.





### OVERVIEW

Computing aims to ensure that all pupils: can understand and apply the fundamental principles and concepts of computer; science, including abstraction, logic, algorithms and data representation; can analyse problems in computational terms, and have repeated practical experience; of writing computer programs in order to solve such problems; can evaluate and apply information technology, including new or unfamiliar; technologies, analytically to solve problems; are responsible, competent, confident and creative users of information and communication technology.

Term	Focus	Assessment
Aut 1	<ul style="list-style-type: none"> <li>Introduction to computer science understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns. Using office software to learn how to work efficiently on a computer and record their progress.</li> </ul>	Ongoing assessment through class tasks
Aut 2	<ul style="list-style-type: none"> <li>Being able to categorise computer systems between general-purpose and purpose-built; recognising different hardware components of computer systems and their functions. Identifying examples of software and being able to recommend choices of hardware and software when buying a new computer.</li> </ul>	Ongoing class assessment of tasks
Spr 1	<ul style="list-style-type: none"> <li>Programming basics in Python: sequencing instructions, variables, input/output. Being able to recognize the syntax used in Python and how to recognize errors.</li> </ul>	Ongoing assessment through class tasks and mid-year assessment
Spr 2	<ul style="list-style-type: none"> <li>Programming in Python: Working with integers, selection, and multi-branch selection. Being able to combine programming concepts to begin to solve problems.</li> </ul>	Ongoing class assessment of tasks
Sum 1	<ul style="list-style-type: none"> <li>Programming in Python: Iteration, Boolean variables, random numbers. Being able to write programs based on a problem brief and programming on paper.</li> </ul>	Ongoing assessment through class tasks
Sum 2	<ul style="list-style-type: none"> <li>Revision and further programming practice.</li> </ul>	Ongoing class assessment of tasks and end of year assessment

### Home Learning:

- Homework set every lesson using Microsoft Forms

### Useful resources:

- BBC Bitesize Computing
- Code.org for programming



### OVERVIEW

Computing aims to ensure that all pupils: can understand and apply the fundamental principles and concepts of computer; science, including abstraction, logic, algorithms and data representation; can analyse problems in computational terms, and have repeated practical experience; of writing computer programs in order to solve such problems ; can evaluate and apply information technology, including new or unfamiliar; technologies, analytically to solve problems; are responsible, competent, confident and creative users of information and communication technology .

Term	Focus	Assessment
Aut 1	<ul style="list-style-type: none"><li>• Binary: converting numbers to and from binary, binary addition.</li><li>• Logic: Boolean logic operators, logic gates, solving logical expressions</li></ul>	Ongoing assessment through class tasks
Aut 2	<ul style="list-style-type: none"><li>• Computational thinking – abstraction, decomposition, algorithms</li><li>• Flowcharts (symbols and flowchart design)</li><li>• Sorting and searching algorithms</li></ul>	Ongoing class assessment of tasks
Spr 1	<ul style="list-style-type: none"><li>• Cybersecurity – Malware, encryption, authentication methods</li><li>• Networks – network types, network hardware, network topologies, network security</li></ul>	Ongoing assessment through class tasks and mid-year assessment
Spr 2	<ul style="list-style-type: none"><li>• Python programming – user interaction, data types, arithmetic operators.</li><li>• Practical programming practice</li></ul>	Ongoing class assessment of tasks
Sum 1	<ul style="list-style-type: none"><li>• Python programming – definite vs indefinite iteration, functions, introduction to lists/arrays.</li><li>• Practical programming practice</li></ul>	Ongoing assessment through class tasks
Sum 2	<ul style="list-style-type: none"><li>• Python programming – list functions, string manipulation, converting data types.</li><li>• Practical programming practice</li><li>• Revision</li></ul>	Ongoing class assessment of tasks and end of year assessment

### Home Learning:

- Homework set every lesson using Microsoft Forms or Carousel learning

### Useful resources:

- BBC Bitesize Computing
- Code.org for programming



### OVERVIEW

Computer Science will encourage learners to understand and apply the fundamental principles and concepts of Computer Science, including abstraction, decomposition, logic, algorithms, and data representation; analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs; understand the components that make up digital systems, and how they communicate with one another and with other systems; and understand the impacts of digital technology to the individual and to wider society.

Term	Focus	Assessment
Aut 1	<ul style="list-style-type: none"><li>• Data representation intro to numbers and formats</li><li>• System Architecture</li><li>• Introduction to Programming Techniques</li></ul>	Ongoing assessment through class tasks
Aut 2	<ul style="list-style-type: none"><li>• System Architecture</li><li>• Storage</li><li>• Introduction to Programming Techniques</li></ul>	Ongoing assessment through class tasks.
Spr 1	<ul style="list-style-type: none"><li>• Wired &amp; Wireless Networks</li><li>• Network Topologies, protocols</li><li>• Producing Robust programs</li></ul>	Ongoing assessment through class tasks. Mid year assessment
Spr 2	<ul style="list-style-type: none"><li>• System Security</li><li>• Producing Robust programs</li><li>• Computational Logic</li></ul>	Ongoing assessment through class tasks..
Sum 1	<ul style="list-style-type: none"><li>• System Software</li><li>• Data representation intro to numbers and formats</li><li>• Translators and facilities of language</li></ul>	Ongoing assessment through class tasks
Sum 2	<ul style="list-style-type: none"><li>• Data storage and compression</li><li>• Ethical, legal, cultural and environmental concerns</li><li>• Internet and WWW</li></ul>	Ongoing assessment through class tasks. End of year paper

### Home Learning:

- Weekly exam questions.

### Useful resources:

- AQA GCSE (9-1) Computer Science textbook.
- Seneca Learning. Codecademy (for learning Python)



### OVERVIEW

Computer Science will encourage learners to understand and apply the fundamental principles and concepts of Computer Science, including abstraction, decomposition, logic, algorithms, and data representation; analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs; understand the components that make up digital systems, and how they communicate with one another and with other systems; and understand the impacts of digital technology to the individual and to wider society.

Term	Focus	Assessment
Aut 1	<ul style="list-style-type: none"><li>• Compression</li><li>• Programming structures</li></ul>	Ongoing assessment through class tasks
Aut 2	<ul style="list-style-type: none"><li>• Standard algorithms</li><li>• Hand tracing Algorithms</li></ul>	Ongoing assessment through class tasks. Mock exams.
Spr 1	<ul style="list-style-type: none"><li>• Database design</li><li>• Revision</li></ul>	Ongoing assessment through class tasks
Spr 2	<ul style="list-style-type: none"><li>• Revision</li></ul>	Ongoing assessment through class tasks. Mock exams.
Sum 1	<ul style="list-style-type: none"><li>• Revision</li></ul>	
Sum 2		

### Home Learning:

- Weekly exam questions.

### Useful resources:

- AQA GCSE (9-1) Computer Science textbook.
- Seneca Learning. Codecademy (for learning Python)