The Bishop of Winchester Academy
Sapere Aude

Mallard Road
Bournemouth
Dorset
BH8 9PW

Computer Science Curriculum

Intent

Curriculum Vision

Computer Science is an ever-evolving subject playing a crucial role in securing and improving the world we live in. From designing animations for multimedia projects to preventing cyber-attacks on a national and global scale, and from the use of machine learning to improve productivity to advances in medical science through robot-assisted surgery, computer scientists play an important role in shaping the future of our society.

Students develop an understanding of the application of principles of computer science and computational thinking including abstraction, decomposition, logic, algorithms and data representation. Students are required to think creatively through game design; think innovatively by discussing and researching new technologies; think analytically by evaluating data and tasks; think logically through programming and Boolean algebra/logic; and think critically by exploring an issue to formulate a conclusion. Students will develop the ability to articulate the moral, social, legal and cultural risks and opportunities of digital technologies.

The knowledge, understanding and skills gained will enable students to pursue, with competence and confidence, a variety of diverse Computer Science career pathways.

Concepts and Skills

The overarching concepts for Computer Science are:

- The characteristics of contemporary processors, input, output and storage devices
- Exchanging data
- Data types, data structures and algorithms
- Legal, moral, cultural and ethical issues
- Elements of computational thinking
- Problem solving and programming
- Algorithms to solve problems and standard algorithms

The overarching skills we aim to develop in Computer Science are:

- The ability to demonstrate knowledge and understanding of the principles and concepts of computer

science, including abstraction, logic, algorithms and data representation
- The ability to apply knowledge and understanding of the principles and concepts of computer science, including analysing problems in computational terms
- The ability to design, program and evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions

Disciplinary Literacy

Disciplinary approach

In Computer Science we support the development of disciplinary vocabulary and the students' ability to read, write and communicate at an academic level so that they master the nuances of the curriculum. Understanding computer vocabulary is essential to build a holistic view of the subject area; we recommend reading materials for students to give them the opportunity to build on concepts and techniques visited in the lesson. With the continuous developments in Computer Science, it is important that students understand, through research, the advantages and limitations of these new developments and be able to articulate the moral, ethical, legal and cultural opportunities and risks that these new developments pose. As a department we provide students with a vocabulary list to aid them in understanding and learning terminology used in their research and we encourage them to use the terminology in their written work. In Computer Science students not only write computer programs but analyse problems in computational terms and then articulate their analysis, design and evaluation through report writing so it is important that students use disciplinary literacy to give them the tools to succeed in this subject area.

Interdisciplinary approach

In order to support the mastery of key concepts in Computer Science, our interdisciplinary approach unites elements of digital literacy, such as the ability to communicate and find information online, which is an important discipline that can be transferred into all areas of Computer Science.

Intellectual autonomy

In order to develop intellectual autonomy and confidence, we foster the willingness and ability of students in Computer Science to comprehend challenging texts, assimilate key concepts and synthesise them with prior learning. Students are equipped to think critically and apply strategies independently so that they can form their own cohesive conclusions and be able to express that in writing.
This is facilitated by encouraging students to keep up to date with new and developing technologies and be able to make connections to the concepts they have studied in class. This develops their understanding, vocabulary and inspires the motivation to further their learning.

Students are encouraged to engage in additional independent reading to deepen their Computer Science understanding. Recommended texts include: PG Online *OCR A-level Computer Science Textbook, Artificial Intelligence: A Ladybird Expert Book* by Michael Wooldridge, *Hidden Figures: The American Dream and the Untold Story of the Black Women Who Helped Win the Space Race* by Margot Lee Shetterly, *The Road to Conscious Machines: The Story of AI* by Michael Wooldridge, *Algorithmic Puzzles* by Anany Levitin and *The Information: A History, a Theory, a Flood* by James Gleick.

Application of Mathematics

The curriculum recognises the need for students to be able to apply mathematics effectively. In Computer Science students use and develop their knowledge of Boolean algebra, comparison of complexity of algorithms and number representation and bases to express computational laws and processes.

Independent Study

In Computer Science students undertake both directed and self-directed independent learning activities that support the strengthening of long-term memory and retrieval. Independent study helps our students achieve mastery in Computer Science and prepares them to work at an undergraduate level.

Directed independent learning tasks set in Computer Science can include background reading to build knowledge and deeper connections to the existing frame of learning, or responding to interlocking questions on a given topic across more than one text source. Self-directed independent study in Computer Science involves retrieval practice which is a crucial component of mastery.

Students practise and develop programming skills in:
Object-Oriented Programming through
- Game development

Data Structure Programming using
- Stacks
- Lists
- Arrays
- Queues
- Records

They write and practise using common search and sort algorithms:
- Binary Search
- Linear Search
- Merge Sort
- Quick Sort
- Bubble Sort
- Insertion Sort

They practise using pathfinding algorithms:
- Dijkstra's Algorithm
- A star Algorithm

They are given questions to research and write about to practise their disciplinary literacy skills.
They are given tasks to complete, for example:
- Design for a network
- Database design

Students undertake self-directed independent study in Computer Science. A crucial component of this involves rigorous self-quizzing. As students encounter challenges and learn to wrestle with demanding concepts and texts they develop not only their knowledge and understanding but also resilience through perseverance. Instead of revision being perceived as something that is crammed into a few weeks, independent study supports spaced practice throughout the curriculum. By repeatedly returning to content covered, students' knowledge has time to 'rest and be refreshed.'

We recognise that not all students process material at the same rate. Students who need extra support to achieve mastery are supported by targeted intervention in Computer Science where a staged or 'scaffolded' process is used to enable students to move from being dependent learners to autonomous ones.

All students have access to our Academy library where a wide range of academic texts, journals and other resources are available.

Implementation

Overview Statement

The curriculum in Computer Science is sequenced coherently so that knowledge, concepts and skills are rigorously developed over time. This supports all students, including the most disadvantaged, and those with high levels of need, especially SEND. Planning is informed by Rosenshine's Principles of Instruction and Cognitive Theory which support students in building secure schemas.

Interdisciplinary links and the application of mathematics are explicitly referenced and exploited in order to deepen understanding. Vocabulary is developed in Computer Science using the principles outlined in the Frayer Model and students are equipped to be able to read, write and speak like a computer scientist. This is done by introducing key terminology in each unit when delivering the

concepts and encouraging students to write and talk using the correct subject terminology; modelling the use of key terminology is also used when teaching and answering questions.

Through the use of independent study resources in Computer Science, students learn at greater depth so that they can become masters in Computer Science and in the skills required to be intellectually autonomous. This is implemented by encouraging students to research and keep themselves updated on new developments in the subject of Computer Science and to think about the environmental, ethical, moral and legal issues which these new developments may bring. They have to design, program and evaluate computer systems, making reasoned judgements about them and presenting conclusions.

Regular retrieval-based activities strengthen long-term memory and aid fluency, as do our cumulative mid-term and end-of-year assessments.

Technology is employed to set online independent tasks using software that gives feedback to students. Students also use Integrated Development Environments to develop their programming skills and to strengthen learning.

Learning character is developed through: resilience – persevering with programming and mathematical challenges; empathy – identifying and respecting the feelings, thoughts and attitudes of others when discussing new technologies and the possible impacts on others; awareness – students are encouraged to link our topic with other topics and to use the skills they have gained to help them to solve problems; collaboration – students are encouraged to draw knowledge from each other and work together to develop each other's strengths and weaknesses; creativity – students develop their own ideas from experiences to develop a computer game; independence – through independent study tasks, students develop their own high expectations and time management skills.

Impact

The Key Stage 5 curriculum builds upon students' initial understanding, extending their knowledge in the areas of: characteristics of contemporary processors, input, output and storage devices; exchanging data; Data types, data structures and algorithms; legal, moral, cultural and ethical issues; elements of computational thinking; problem solving and programming, and algorithms to solve problems. Through developing their knowledge of the principles and concepts of computer science including analysing problems in computational terms, students are able to design, program and evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions. Students also develop an understanding of the interdisciplinary nature of their studies and this is supported though explicit cross-curricular links in Mathematics.

Students use their knowledge of cognitive theory to recognise and use the most impactful methods of revision and retrieval practice.

Academic progress in Computer Science is recognised through the Computer Science A-level which acts as a benchmark of mastery; this provides students with the national currency needed for access to higher education and apprenticeship courses, and prepares them for a career in any workplace.