Computer Science

Curriculum Principles

Computer Science Faculty Vision

We teach both transferable problem-solving skills and highly specialised computing expertise. We inspire students to consider a fulfilling career in computing or STEM. We keep strategy simple and everyone is consistent. We will work hard to develop computational thinking skills and increase participation in Computer Science.

By the end of their all-through education, a student of Computer Science at Dixons Kings will:

- Know the theory of:
  - Hardware, networks, cyber security, data representation, logic, programming, algorithms, translators, relational databases, ethical issues
- Understand:
  - Computational thinking and problem solving
  - Tracing, interpreting and creating algorithms
  - How to code complete programs in Python

In order to truly appreciate the subject and create deep schema, topics within Computer Science have been intelligently sequenced with the following rationale:

- Programming fundamentals are tightly linked to hardware and the low-level implementation details of code execution. Code in the context of problem solving and program creation has already been introduced in KS3. At the beginning of KS4, a more technical understanding is developed early on. Programming topics and practice is then interleaved with theory topics throughout year 9 and 10. These sequencing requirements led to covering variables, data types, selection and iteration at the start of year 9, followed by theory for computer systems, data representation and then standard algorithms. This provides the theoretical underpinning needed before students encounter more advanced code in the second programming unit which covers nested iteration, string manipulation and arrays. Finally, in year 9, the fundamentals of databases and networks are covered. Both topics are revisited in more depth in year 11. Covering databases at this point consolidates what has been learned about data types and storage and provides forward links for year 10 when students learn more about how programming languages implement data structures (records).
- When students move into year 10, they have already been exposed to a large number of algorithms and are developing the programming literacy needed to implement algorithms in code. At this point, we cover the theory of algorithm design, abstraction and decomposition, and also skills needed to investigate and analyse algorithms. The remaining theory topics in year 10 build on ideas introduced in year 9: cyber security relies on basic networking knowledge, symbolic Boolean logic expressions build on Boolean expressions used in programming, and machine code, assembly and translators expand on ideas in computer systems, especially the fetch-execute cycle. Covering software at this point allows students to consolidate understanding of the von Neumann model, which students often find difficult to learn in isolation.
- Programming skills are built gradually over the whole course, with the content split into four programming modules, supported by additional modules covering standard algorithms, algorithm design, and robust and secure programming. Hardware, databases and networks are also split into two modules, to flatten the learning curve in year 9 and reduce cognitive load for students.
- Advanced aspects of databases, networks and programming are revisited in year 11, and students learn more about robust and secure programming, which is dependent on all the programming skills students have developed earlier in the course.

The Computer Science curriculum at Dixons Kings has been influenced by academic research:

- Our approach relies heavily on Sue Sentance's work on structured approaches to teaching programming. We have developed our own resources using these concepts. All programming tasks follow the PRIMM methodology, giving students an opportunity to interpret and engage with code prior to manipulating and extending it. This has proven to be an effective method of scaffolding code creation.

- Lesson content is influenced by past and present GCSE specifications, with the latest revisions tailored to the requirements of the AQA 8525 specification. We draw on the examiners reports for in-depth analysis of potential misconceptions.

Our Computer Science curriculum ensures that social disadvantage is addressed:

- As a STEM subject, Computer Science is critical in providing independent learning and problem-solving skills, particularly higher-order thinking skills required for any science or engineering career. All intervention is based around closing individual topic and skill gaps as early as possible with small groups of students. This addresses inequalities fairly without singling out students for a particular characteristic.
- Revision guides and practice workbooks are subsidised 100% for disadvantaged students, to support with retrieval practice for homework tasks using the workbook.
- SEN students are given tailored support according to individual needs, and also additional support through:
    - Frequent book checks
    - Computer-based learning opportunities
- Specialised knowledge of computational and algorithmic thinking, and deeper understanding of curriculum areas in Computer Science (powerful knowledge, Young 2014) empowers students, particularly disadvantaged students, by going beyond everyday knowledge. Curated information (Knowledge Navigators) allows students to build on lesson content and revise linked knowledge efficiently. This is further supported with retrieval practice.
    - Young M, Lambert D, Roberts C, et al. (2014) Knowledge and the Future School: Curriculum and Social Justice. London: Bloomsbury.

We fully believe Computer Science can contribute to the personal development of students at Dixons Kings:

A true love of Computer Science is developed by teaching beyond the domain of the GCSE specification.

Examples of such content:

- Context of the development and future of computing technologies.
- Detailed background information about hardware operation, such as SDRAM and flash memory.
- Support to use libraries and syntax discovered during independent research

Our belief is that homework is used for deliberate practice of what has been taught in lessons. We also use retrieval practice and spaced revision to support all students with committing knowledge to long term memory.

- Homeworks use a range of formats, including:
    - Exam-style questions that practice what is taught recently and revise previous topics.
    - Programming and problem-solving questions that require independent thought.
    - Retrieval practice quizzes using multiple choice or short-form answers.
    - Extended writing tasks, usually based on case-studies or research.
- Do Now activities are used for spaced practice of key skills, and also as an opportunity for low-stakes testing and reteaching.

Opportunities to build an understanding of social, moral and ethical issues are developed alongside links to the wider world, including careers:

- Study of ethical and legal impacts of computing, especially privacy concerns and environmental concerns.
- Learning about the digital divide and unequal access to technology.
- Computer Science students can access cyber-security extracurricular opportunities through GCHQ, which build on whole-school opportunities in KS3.

Remote working in our subject supports students to access the full curriculum in the following ways:

- Providing the same content used in lessons through the Online Learning Platform.
- Linking teaching to video content, especially video series by Craig'n'Dave.
- Referring students to gamified online programming courses such as CoderDojo, CodeWars, and Hour of Code.
- Using teaching resources provided by Oak National Academy.
- Synchronous and asynchronous video lessons providing the same explanations and tasks used in lessons.
- Assignments set and marked through Teams. This has been used as an element of home learning and is also being developed as part of our homework provision. This allows tailored feedback to be provided and gaps to be identified early on.

Further Information can be found in:

- Long term plans
- Schemes of work
- Knowledge Navigators
- Online resources through the Online Learning Platform