

COMPUTER SCIENCE

KS4 Years 10-11 Curriculum rationale

The GCSE Computer Science course (Edexcel 1CP2 Specification) provides students with a comprehensive understanding of the principles and practices that underpin modern computing. As we live in an increasingly digital world, this course equips students with the knowledge and skills to navigate, innovate, and excel in technology-driven environments. This rationale outlines the structure of the course, the skills and learning outcomes, the benefits of studying computer science, and how parents can support their child's learning journey.

1. Course Overview

The GCSE Computer Science course focuses on both theoretical knowledge and practical programming skills. It provides students with the opportunity to develop critical thinking, problem-solving, and analytical skills. The course is divided into two main components:

1. Computer Systems (Theory)

- Understanding the structure and function of computer systems, including hardware, software, and peripheral devices.
- Learning about data representation, binary, and how data is processed and stored.
- The role of operating systems and the management of resources.
- Cybersecurity, threats to computer systems, and ethical issues related to computing.

2. Computational Thinking, Algorithms, and Programming (Theory & Practical)

- Developing problem-solving skills and the ability to break down problems into smaller, manageable parts.
- Understanding algorithms and their efficiency, as well as the process of designing, writing, testing, and refining code.
- Developing practical programming skills in a high-level language (such as Python).
- Testing and debugging programs to ensure they function as expected.
- An understanding of how software development works in real-world contexts.

Skills and Learning Outcomes

The Edexcel GCSE Computer Science course fosters a range of key skills that are crucial for the modern workplace and further academic study:

Key Skills Developed:

- **Problem-Solving and Analytical Thinking:** The ability to think logically, break problems into smaller sub-problems, and find efficient solutions using computational methods.
- **Programming:** Gaining proficiency in writing, testing, and debugging code. Students will learn to design algorithms, implement them in a programming language, and adapt existing programs to solve new problems.
- **Data Handling and Representation:** Understanding how data is represented within a computer system (e.g., binary, ASCII, images, and sound) and how this impacts storage, transmission, and processing.
- **Cybersecurity Awareness:** Knowledge of common threats to computer systems, including malware, hacking, and data breaches, and how to protect against them.
- **Ethical Considerations in Computing:** Critical thinking around the ethical issues related to the use of technology, including privacy, surveillance, and the impact of technology on society.

Learning Outcomes:

By the end of the course, students will be able to:

- Demonstrate a clear understanding of how computer systems operate and how data is processed, stored, and communicated.
- Design, write, test, and evaluate computer programs to solve problems.
- Understand and apply algorithms, data structures, and computational thinking in real-world scenarios.
- Critically analyze the impact of digital technologies on individuals, organizations, and society.
- Recognize and address ethical and security issues in the digital world.
-

	Autumn	Spring	Summer
Year 10	<p>Topic 2: Data</p> <p>Topic 6: Problem solving with programming</p>	<p>Topic 1: Algorithms and Computational Thinking</p> <p>Topic 3: Computers</p> <p>Topic 4: Networks</p> <p>Topic 6: Problem solving with programming</p>	<p>Topic 4: Networks</p> <p>Topic 5: Issues and Impact</p> <p>Topic 6: Problem solving with programming</p>
	<p>Topic 2: Data</p> <p>2.1 Binary 2.2 Data representation 2.3 Data storage and compression</p> <p>Topic 6: Problem solving with programming</p> <p>6.1 Develop code 6.2 Constructs 6.3 Data types and structures 6.4 Input/output 6.5 Operators 6.6 Subprograms</p>	<p>Topic 1: Algorithms and Computational Thinking</p> <p>1.1 Decomposition and abstraction 1.2 Algorithms 1.3 Truth tables</p> <p>Topic 3: Computers</p> <p>3.1 Hardware 3.2 Software 3.3 Programming languages</p> <p>Topic 4: Networks</p> <p>4.1 Networks 4.2 Network Security</p> <p>Topic 6: Problem solving with programming</p> <p>6.1 Develop code 6.2 Constructs 6.3 Data types and structures 6.4 Input/output 6.5 Operators 6.6 Subprograms</p>	<p>Topic 4: Networks</p> <p>4.1 Networks 4.2 Network Security</p> <p>Topic 5: Issues and Impact</p> <p>5.1 Environmental 5.2 Ethical and legal 5.3 Cybersecurity</p> <p>Topic 6: Problem solving with programming</p> <p>6.1 Develop code 6.2 Constructs 6.3 Data types and structures 6.4 Input/output 6.5 Operators 6.6 Subprograms</p>

<p>Year 11</p>	<p>Topic 1: Algorithms and Computational Thinking</p> <p>Topic 6: Problem solving with programming</p>	<p>Revision of all topics</p> <p>Topic 4: Networks</p> <p>Topic 6: Problem solving with programming</p>	<p>Revision of all topics</p> <p>Topic 6: Problem solving with programming</p>
<p>How parents / carers can support</p>	<p>Topic 1: Algorithms and Computational Thinking</p> <p>Algorithms and computational logic</p> <p>Topic 6: Problem solving with programming</p> <p>Advanced programming concepts Robust and secure programming Classification of programming languages</p>	<p>Topic 4: Networks</p> <p>Networks and Cyber Security</p> <p>Topic 6: Problem solving with programming</p> <p>6.1 Develop code 6.2 Constructs 6.3 Data types and structures 6.4 Input/output 6.5 Operators 6.6 Subprograms</p>	

What is Expected of Students?

To succeed in GCSE Computer Science, students are expected to demonstrate the following:

1. **Engagement and Curiosity:** Students should be actively engaged in exploring the topics covered, developing a genuine curiosity about how computers and technology work.
2. **Independent Learning:** While the course is supported by structured lessons, students are expected to take responsibility for their own learning, including completing homework, practicing programming, and reviewing key concepts outside of class.
3. **Consistent Practice and Application:** Students will need to practice their programming skills regularly and apply theoretical concepts to practical programming challenges.
4. **Problem-Solving Mindset:** Computer Science is about finding solutions to complex problems. Students should approach problems with persistence, creativity, and a willingness to learn from mistakes.

Assessment

The Edexcel GCSE Computer Science course is assessed through two main components:

1. **Paper 1: Computer Systems.**
The paper is 1 hour and 30 minutes.
 - The paper consists of five questions (one question per topic, multiple-choice, short-, medium- and extended-open-response, tabular and diagrammatic items).
 - The paper is out of 75 marks.
2. **Paper 2: Computational Thinking, Algorithms, and Programming**
The paper is 2 hours.
 - The paper consists of six compulsory questions.
 - The paper is out of 75 marks.
 - Students must answer all questions.
 - The questions are practical in nature and require students to design, write, test and refine programs in order to solve problems.
 - Students will complete this assessment onscreen using their Integrated Development Environment (IDE) of choice.
 - Students will be provided with coding files, a hard copy of the question paper, and the Programming Language Subset (PLS)