



## 100% EXAMINATION

**Paper 1:** Computational thinking and problem-solving - 1hr 30 mins 50% Covers computational thinking, problem solving, code tracing, algorithms, programming and applied computing.

**Paper 2:** Written assessment - 1hr 30 mins 50% Covers data representation, computer systems, networks, cyber security, ethical, legal and environmental impacts of digital technology.



## NON-EXAMINED ASSESSMENT

**Software Development**  
20hrs – The development of a computer program along with the computer programming code itself which has been designed, written and tested by a student to solve a problem. Students will produce a report/log outlining this development.



## FURTHER EDUCATION

This subject is good preparation for any advanced, vocational and degree level courses across Science, Computing and Engineering.



## CAREER OPPORTUNITIES

Professions which utilise this qualification include Programming and Software Development, Software Project Management, Robotics, Artificial Intelligence and Network Engineering.



## OCR WEBSITE

Click on the link above to download full specification/s.

## WHAT IS THIS COURSE ABOUT?

In this technological age, a study of Computer Science, and particularly how computers are used in the solution of a variety of problems, is essential to learners. Computer Science demands both logical discipline and imaginative creativity in the selection and design of algorithms and the writing, testing and debugging of programs; it relies on an understanding of the rules of language at a fundamental level; it encourages an awareness of the management and organisation of computer systems; it extends learners' horizons beyond the school or college environment in the appreciation of the effects of computer science on society and individuals. Students need strong reasoning and logical skills for this subject and should only choose it if they are likely to achieve a grade 6/7 (B) or higher in GCSE Mathematics.

## WHAT SKILLS WILL I DEVELOP?

- Take a systematic approach to problem solving and make use of conventions including pseudo code and flowcharts
- Design, write, test and refine programs, using one or more high-level programming language with a textual program definition, either to a specification or to solve a problem
- Use appropriate security techniques, including validation and authentication
- Evaluate the fitness for purpose of algorithms in meeting requirements efficiently using logical reasoning and test data
- Use abstraction effectively
- Model selected aspects of the external world in a program
- Structure programs into modular parts with clear, well- documented interfaces and apply computing-related mathematics