



## Computer Science Curriculum Intent

At BVGS in the Computer Science department we aim for all students to become:

**Problem solvers** – have the ability to analyse and solve problems and to design, implement and deploy algorithms to solve real problems

**Logical thinkers** – apply the concepts of abstraction, decomposition and logic

**Creative** – innovate, analyse and think critically in solving problems

**Aware** – understand the impact that technology has on people and on the wider society

**ICT Capable** – able to use computer software applications and programs with confidence and skill

## GCSE Computer Science

Head of Department: Mr K. Kirk

Like it or not, you're living in it – this is the Digital Age. Computer programmes have all but infiltrated every aspect of our lives. Computer scientists theorise, design, develop and apply the software and hardware for the programmes we use day in day out – sounds pretty important to us! Every industry uses computers so naturally computer scientists can work in any. Problems in science, engineering, health care, and so many other areas can be solved by computers. It's up to the computer scientist to figure out how, and design the software to apply the solution. You'll learn how to design and write real computer programs, create apps to run on your smart-phones and tablets and discover how to spot errors in all those dodgy freeware games you insist on running on your laptop. Even better, you'll learn about the latest cutting edge technology and possibly be inspired to becoming the next Mark Zuckerberg or Grace Hopper.

A significant amount of our GCSE Computer Science students go on to study Computer Science at A-Level. Other students find that their practical skills are invaluable in supporting further study in science, mathematics and engineering courses at A-Level.

### **Topics covered:**

Paper 1: Principles of Computer Science

50% of the total GCSE marks

This unit is externally assessed (unseen examination).

- Understanding of what algorithms are, what they are used for and how they work; ability to interpret, amend and create algorithms.
- Understand the requirements for writing program code.



- Understanding of binary representation, data representation, data storage and compression, encryption and databases; ability to use SQL to insert, amend and extract data stored in a structured database.
- Understanding of components of computer systems; ability to construct truth tables, produce logic statements and read and interpret fragments of assembly code.
- Understanding of computer networks, the internet and the World Wide Web; ability to use HTML and CSS to construct web pages.
- Awareness of emerging trends in computing technologies, the impact of computing on individuals, society and the environment, including ethical, legal and ownership issues.

## Paper 2: Application of Computational Thinking

50% of the total GCSE marks

This unit is externally assessed (unseen examination).

- Understanding of what algorithms are, what they are used for and how they work; ability to interpret, amend and create algorithms.
- Understanding how to develop program code and constructs, data types, structures, input/output, operators and subprograms.
- This component may also draw on:
- Understanding of binary representation, data representation, data storage and compression, encryption and databases; ability to use SQL to insert, amend and extract data stored in a structured database.
- Understanding of components of computer systems; ability to construct truth tables, produce logic statements and read and interpret fragments of assembly code.
- Understanding of computer networks, the internet and the World Wide Web; ability to use HTML and CSS to construct web pages.
- Awareness of emerging trends in computing technologies, the impact of computing on individuals, society and the environment, including ethical, legal and ownership issues.

## Programming Project

20 hour programming project.

This unit will be internally assessed but will not count towards the final GCSE.

This is a practical 'making task' that enables students to demonstrate their computational techniques using the Python programming language. Students will:

- decompose problems into sub-problems
- create original algorithms or work with algorithms produced by others



- design, write, test, and evaluate programs.

How do we assess the progress of pupils:

1. Pupils have regular assessment tests to familiarise them with the subject content and style for the examination papers.
2. There is regular peer and self-assessment throughout the entire GCSE course to enable students to rectify any weaknesses they may have.

Keeping up with and extending subject knowledge:

- We recommend that students consolidate and read through relevant VLE material after the lesson.
- Working through programming challenges in Python at home.
- Encourage them to explore the Python programming language and set their own goals and projects.
- All the resources from the Computing department are available on the school's VLE.

Improving examination skills:

We recommend that pupils enhance their examination skills through extra GCSE past paper question practise. Past papers are available from the department's VLE or from the Edexcel website. Writing these answers in exam conditions and practicing how long to spend on each question is greatly beneficial.