

## Develop:

Abstraction  
Computational Thinking  
Data structures  
Security and safety  
Independence

## Sandbach School Computing

# Qualifications Phase Year 2 Computer Science Curriculum Sequence

**Intent:** Studying GCSE Computer Science is the entry point to a career in IT. Learners will gain a foundational understanding of a wide range of skills that are highly in demand and formulate the grounding for future thought and innovation in digital fields. Through careful study and the development of computational thinking, learners will aspire to fulfil the job roles of the next few decades.

### T1

System Software  
Algorithms

### T2

Producing Robust Programs  
Computational Logic

### T3

Translators and Facilities of  
Languages  
  
Ethical, Legal, Cultural and  
Environmental

### T4

Programming Techniques

### T5

Revision

### T6

#### Why these topics?

Systems software completes the learning for Component 1 of the course.  
The standard algorithms can be rote learned and it is effective to do this in Year 11 closer to the exams. This is also an opportunity to revise the programming content by applying it to the algorithms.

#### Why This Topic?

The robust programming unit completes the knowledge portion of the programming skills for the course.  
The computational logic unit is one that students generally pick up quickly. Having it late in the course is an easy win.

#### Why This Topic?

The translators unit involves higher order thinking skills which are best developed when pupils are at their most focussed. The unit also creates an excellent opportunity for showing pupils how to complete the large 9-mark exam questions, a key skill to learn and best learned just before the exam.  
The content for the ethical unit involves higher order thinking skills which are best developed when pupils are at their most focussed. The unit also creates an excellent opportunity for showing pupils how to complete the large 9-mark exam questions, a key skill to learn and best learned just before the exam.

#### Why This Topic?

This unit will be interleaved with other units, providing links to the theoretical aspects of the course.  
Exam style questions will be used throughout to make sure that practical skills translate to better exam answers.  
Practical programming is a requirement of the course

#### Why This Topic?

Teachers can identify gaps in students knowledge and tailor learning to their needs to best support them in the upcoming exams.

#### Curriculum Links

- Technical
- Code
- Algorithms
- Abstraction

#### Curriculum Links

- Code
- Logic
- Responsibility

#### Curriculum Links

- Technical
- Legal
- Environment

#### Curriculum Links

- Code
- Computational thinking
- Abstraction

#### Teaching these topics here supports:

Algorithmic thinking and understanding algorithm complexity. Useful for analysis of algorithms in the GCSE exam.

#### Teaching these topics here supports:

Understanding best practices in code suitable for the GCSE exam and career entry points.

#### Teaching these topics here supports:

Completion of learning for the GCSE exams.  
Study in exam technique to answer level of response questions.

#### Teaching these topics here supports:

Programming topics in the GCSE exam.

#### Teaching these topics here supports:

Direct support for the exams.

#### These topics feed from:

Prior computational thinking topics.  
Hardware and software units.

#### These topics feed from

Previous logic and binary units. All prior programming experience.

#### These topics feed from

New content but based on previous technical backdrop.

#### These topics feed from

Builds on previous Python unit.

#### These topics feed from

All prior learning culminates here.

