We want Pupils to be able to code the next Facebook or Xbox blockbuster, to found the next Pixar or Wikipedia, to

publish a magazine, to make a film, to connect with others around the world, to use all the wonderful new tools to

unleash your creativity and hack a better future for all of us!

The IT department wants every child to be able to do three things by the time they leave. Students should

understand:

how to build computational tools,

how to use those tools effectively and

how to use those tools responsibly.

To that end we have divided the curriculum into three strands, each being taught in a different term. These strands

are:

Computational Thinking

Working Life

Connected World

This is where you roll your sleeves up and play around with how things work. You’ll open up a computer and look

inside. You’ll begin to see how code works, how it turns electricity into ones and zeroes into MMRPGs, how

important hacking is and more. Many people still think it’s magic. You’ll be a wizard to them. Like Merlin, just

without the beard.

Computational thinking is fundamental to computer science. It is a particular type of approach to problem-solving

that usually (though not always) finds expression in software. It is an academic discipline in itself and has plenty of

cross-curricular “oomph” especially with maths and sciences. In Google’s words, it “involves a set of problem solving skills and techniques that software engineers use to write programs that underlie the computer

applications you use”.

There are essentially four such skills: abstraction (generalising about the qualities of a problem), decomposition

(breaking down complex problems into smaller ones), algorithms (sequencing instructions) and pattern-spotting.

Rather than teach these theoretically, each student becomes familiar through a range of different programming

projects, be it robotics or writing a game for an Xbox. To make computational thinking as tangible as possible,

every student will have repeated opportunities to design, write, run, and debug, executable programs.

Want to start your own company? Maybe you want to make your own film? Or perhaps you want to gather some

data to help you cure malaria? Not yet? Too young? Maybe you just have a history essay due in and you want to

make your teacher spit out their digestive biscuit in surprise at the amazing facts you’ve found? Chances are a

computer can help. This is where we show you how.

The Computing curriculum has recently been changed. The changes have moved towards computer studies,

programming and coding.

We do, though, need students to understand how to use these tools, both for their own projects, for other

subjects and for later life. As per the computer science, we aim to do that by making use of real-world projects

and data wherever possible. To that end, students learn how:

to use spreadsheets by using them to analyse census data to see how their neighbourhoods compare to

others

to use search engines by using them to discover and evaluate different sources for use in their own research

to use presentation software by using them for Kids Dig, an event where the children take over the teaching of

lessons for half a day.

There is a common assumption that children are “digital natives”. In effect, so the thinking goes, because children

have been born into a connected world they understand the rules for working in it better than their adults. It is a

dangerous assumption. While, yes, adults still might be working out how best to operate in a connected world and

making very visible mistakes, digital citizenship needs both new world technology and old world values.

This strand of the curriculum helps children learn two things. First, we look at how to navigate life online safely and

responsibly through looking at topics such as cyber safety and identity theft. Second we look at how to become a

proactive “force for the good” online, through looking at the Wisdom of Crowds, privacy issues and how to amplify

positive behavioural norms in online forums or social networks.

Autumn Spring Summer

Fundamentals of algorithms

Programming

Aspects of software

development

Fundamentals of data

representation

Computer systems

Fundamentals of computer

networks

Fundamentals of cyber security

Ethical, legal and environmental

impacts of digital technology on

wider society, issues of privacy