Computer Science

Learning Journey

Intent · Implementation · Impact

Computer Science Curriculum Learning Journey

Curriculum Intent

Computers are firmly embedded in our modern-day society. The vast number of objects that we use and interact with in our homes, places or work or study, and the places where we socialise will, typically, in some way, contain a form of computer. Gone are the days where we needed a keyboard, mouse and monitor to interact with computers; we now have computers in our pockets and bags in the form of smartphones, on our wrists in the form of smartwatches, and on our tables and shelves in the form of smart-speakers. Computers are here to stay – our interaction with them should only increase.

Computing is multifaceted – we could consider it to have three main components in Hardware, Software and Data, with each of the components having further subcomponents that relate to the user, the administrator, and the creator - the Computer Science curriculum at Parkside has been designed to allow students to get a well-rounded understanding these different aspects and components of computing. It is important for students to understand that computers aren't just boxes that operate by some unknown power.

Parkside Computer Science curriculum ensures that school students are initially introduced to the fundementals of computing, investigating and learning about basic office tools, email, and the positives and negatives of the World Wide Web. The students will then go onto taking their first steps into programming with languages such as Scratch and Python – an area of computing that will allow the development of logical thinking to create things such as algorithms that handle everyday problems and will be the basis of the developed computer code so important to the Programming discipline.

As the curriculum develops, aspects of computing that relate to things such as computer graphics and HTML development are introduced. These aspects of the Computer Science curriculum allow the students to understand the user interface and its need to be engaging for the user.

Data is introduced into the curriculum initially in terms of how it can be manipulated and presented in spreadsheets but then in terms of how it can be stored within databases and how it can be transported within data networks.

Hardware and software are learnt about in topics within the curriculum that relates to components of a computers system.

There is a cross curricular aspect to the Parkside Computer Science curriculum. For example, an understanding of switches introduced in Technology is key to understanding how a computer processes and stores information, understanding of number systems in Mathematics helps computing students appreciate how computers utilise binary and hexadecimal, and a good grasp of English grammar and the ability to learn new languages in MFL assist students when learning computing programming.

The school is committed to providing its students an appreciation of all of the various aspects of computer science so that they can work towards being expert participants in this ever-growing part of our lives and world.

A Beal – Teacher of Computer Science

Computer Science Curriculum Learning Journey

Curriculum Implementation

The Computer Science learning journey is currently three years, consisting of learning in Y7, 8 &9. We are planning as a school to enhance the curriculum and broaden to learning over 5 years as we develop this curriculum area in our school.

At the initial stages of the learning journey students will develop their understanding of Computing fundamentals and will acquire skills to be able to develop their computer science literacy. There are three distinct units of Computer Science, understanding software, developing knowledge of hardware and architecture and understanding data. As the curriculum progresses students will have the opportunity to see how these distinct units interleave and support holistic understanding of the use of computers. In 2021/22, year 7 and 8 will both follow the initial stages of the learning journey to support missed learning time due to COVID restrictions and ensure that they are equipped an elementary understanding of the subject area before progressing onto more complex content.

The planned curriculum within year 9 is structured to support transition into GCSE learning of Computer Science. The learning journey in year 9 targets key components that will be essential in years 10 and 11; students will develop their understanding of coding (in different computer programming languages), computer architecture and data (its structure, how it is stored, and how it is transported).

Disciplinary literacy and the knowledge of specialised vocabulary which are used in Computer Science is essential to supporting students' academic and theoretical knowledge of the subject. Computer Science literacy is important to all aspects of the subject, for example when students are learning to use computer programming, they must follow grammatical rules for script and syntax to ensure that the software accurately understands the intended program request. The development of subject specific literacy will also support pupils' ability to express their knowledge of Computer Science and the theoretical understanding of the subject. We know that without competent and confident literacy skills, our pupils cannot flourish in the world; it is therefore our duty to prioritise the vital acquisition of high-quality communication skills within all areas of the curriculum.

Embedded within the computer science journey, is teaching student about diversity within the subject area. It is essential to us that students understand the breadths of progression pathways with computing and the wide variety of people within the industry. To develop the students' understanding of diversity within computing, we build upon their cultural capital by teaching them about the emerging democracy of the computing industry. We promote inclusivity in our curriculum through our visual displays in the classroom and through additional content which supports students' broad and balanced knowledge of computer science.

The development of computing means that the use of technology is central to the majority of job roles regardless of whether they are seen as traditional computing role or an alternative vocation where students would be expected to understand computing and ICT. The computer science learning journey is structured to ensure that students are equipped as confident and competent digital users.

Curriculum Impact

The impact of our Computer Science Curriculum Learning Journey is defined through the accessibility pupils have to developing knowledge and the application of skills. This is determined through a number of measures:

- ✓ **Formative Reporting of Pupil Progress** will take place through assessments each lesson, and include pupil questioning, discussion, the completion of differentiated questions for each learning objective and the completion of homework.

- ✓ **Summative Reporting of Pupil Progress** will take place through topic assessments and assessment of student's practical outcomes on the computer at the end of each topic. Parents/Carers receive a report following each mid and end point assessment to understand their child's current depth of knowledge and the support they need to further develop this knowledge both inside and outside of school.

- ✓ **In-lesson learning, participation and belonging** is measured by continually monitoring pupil punctuality to lessons, rewards and sanctions, behaviour referrals, pupil voice and work-scrutiny. Our hope is that by continued positivity in these areas, our Computer Science Curriculum Learning Journey is accessible and will, therefore, positively impact knowledge growth and skill application.

- ✓ **Knowledge and understanding of the key concepts and skills**. Pupils will be able to use a computer with confidence. Pupils will develop their understanding of how computers can be used to present information and how to use the internet to find information. Over the course of the learning journey, pupils will develop their skills in data manipulation and using computer science for coding. All pupils will understand the vocational and academic opportunities available to them in the field of computer Science

Computer science is currently delivered Y7-9. Parkside is working with pupils in these year groups to develop their understanding of Computer Science to enable this subject to be delivered as a KS4 option in the future.

Computer Science Curriculum Learning Journey

Learning Journey overview

Note: Year 8's curriculum mirrors Year 7's curriculum during the 2021/22 academic year. This is due to the fact that the current Year 8 cohort had minimal, if any, introduction to Computer Science teaching during their first year at Parkside Community School due disruption caused by 2020/21 COVID "lockdown" restrictions. It would be unfair and unwise to deliver Year 8/Year 9 Computer Science topics to the current Year 8 students without them first being taught the subject fundamental topics.

A 3-year plan has been created to handle the above issue. Appropriate variation in the Computer Science curriculum will be introduced going forward from next academic year to firstly, ensure that the Current Year 8 cohort are at a point to choose Computer Science as an option in Years 10 and 11 and secondly, that each year cohort are delivered distinctly different but appropriate schemes of learning by academic year 2023/24.

Year 7: Topic summary

How we interleave topics to enable pupils to build and recall knowledge.

KS3 Year 7 Computing - Interleaving Approach

Year 7

Unit 1	Unit 2	Unit 3
	Unit 1	Unit 1 / Unit 2
Unit 4	Unit 5	Unit 6
Unit 1 / Unit 2	Unit 1 / Unit 2	Unit 1 / Unit 2
Unit 3	Unit 3 / Unit 4	Unit 3 / Unit 4
		Unit 5

Year 7

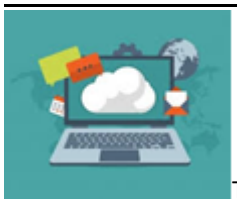
Unit 1	Unit 1 Introduction to Computing							
	Unit 1 Memory Recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4	L1 L2 L3 L4 L5 L6
	Lesson Content	L1. Folder & file management	L2. Email	L3. Office 360	L4. E-Safety	L5. IDEA Award	L6. Searching the Web	L7. Assessment
Unit 2	Unit 2 Scratch Super Mario							
	Unit 2 Memory Recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5	L2 L3 / L1 L2 L3 L4 L5 L6 L8 / Unti 1
	Lesson Content	L1 Adding Sprites	L2. Cretaing algorithyms	L3. Mushrooms	L4. Health	L5. Princess Peach	L6. Welcome Screen	L7. Adding Sound / L8. Assessment
Unit 3	Unit 3 Computer Graphics							
	Unit 3 Memory Recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5	L1 L2 L3 L4 L5 L6 / L8 / Unit 1 & 2
	Lesson Content	L1. Magic wand & rubber stamp	L2. Lasso & filters	L3. Layers	L4. Vector & bitmap graphics	L5. Vector logo	L6. Movie poster	L7. Text / L8. Assessment
Unit 4	Unit 4 Spreadsheet Modelling							
	Unit 4 Memory Recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5	L1 L2 L3 L4 L5 L6 / L1 L2 L3 L4 L5 L6 L7
	Lesson Content	L1. Computer Modelling	L2. Financial model	L3. What If	L4. Conditional formatting	L5. Charts 7 macros	L6. Practical	L7. IDEA Award / L8. Assessment
Unit 5	Unit 5 Introduction to Python							
	Unit 5 Memory recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5	L1 L2 L3 L4 L5 L6 / L1 L2 L3 L4 L5 L6 L7
	Lesson Content	L1. Strings & variables	L2. Number & arithmetic	L3. Selection	L4. Writing algorithms	L5. While loops	L6. Searching	L7. IDEA Award / L8. Assessment
Unit 6	Unit 6 HTML & Web Development							
	Unit 6 Memory recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5	L1 L2 L3 L4 L5 L6 / L1 L2 L3 L4 L5 L6 L7
	Lesson Content	L1. HTML	L2. CSS	L3. Design	L4. IDEA Award	L5. Development	L6. Web forms	L7. Testing / L8. Assessment

Computer Science Curriculum Learning Journey

Computing Topic Descriptors

Year 7

Unit 1 Introduction to Computing



This unit is an introduction to Computing at Parkside Community School. Students will be introduced to different components to equip them with the basic knowledge and skills when working with Computers. Students begin by looking at file management and security. Students will know how to use the school email system and Office 360 which all students have access to at home. Students will look at online dangers and understand how to protect themselves online and how to effectivity search the Internet and understand that not all information on the Web is accurate or truthful.

Unit 2 Scratch Super Mario

This unit introduces students to computer programming using the software Scratch which is a block-based visual programming language. This unit enables students to create their own "Super Mario" game using the software Scratch. Each lesson builds on the previous lessons work to create a fully playable and interactive game. Students arrive at Parkside with very little previous experience of using Scratch to program.

Unit 3 Computer Graphics

This unit introduces students to the topic Computer Graphics using the software Adobe Fireworks. This will be the first time students have had access to this specialist software. The unit consists of skills builder lessons which introduces students to some of the main graphic editing tools and techniques such as; magic wand tool, rubber stamp tool, lasso and filters, using layers and text. Students will use the skills established to create a logo and a movie poster to showcase their skills.

Computer Science Curriculum Learning Journey

Unit 4 Spreadsheet Modelling

This is a practical, skills-based unit covering the principles of creating and formatting basic spreadsheets to produce and use simple computer models. It is suitable for pupils who have a basic knowledge of spreadsheets including cell references, simple formulae and formatting, although these topics are revised in the first lesson, making it also suitable for pupils new to spreadsheets due to the inconsistencies of ability from KS2.
The unit is centred around creating a financial model for a TV show. Students start by looking at different types of model and then use basic spreadsheet techniques to create and format a simple financial model to calculate the expected income from viewers' voting. The model is then extended to include sales from merchandising, with the introduction of "what if" scenarios. Finally, the students create a seating plan, book seats and calculate income from seat sales. Spreadsheet features covered include SUM, MAX, IF and COUNTIF functions, cell naming for absolute referencing, conditional formatting, validation, charting and simple macros. Microsoft Excel software is used in this unit.

Unit 5 Introduction to Python

This unit of work is an introduction to Python, a powerful but easy-to-use high-level programming language. Although Python is an object-oriented language, at this level the object-oriented features of the language are barely in evidence and do not need to be discussed. The focus is on getting pupils to understand the process of developing programs, the importance of writing correct syntax, being able to formulate algorithms for simple programs and debugging their programs.

Unit 6 HTML & Website Development

Students will learn the basics of HTML and CSS. They will learn how to create text styles and add content, including text and graphics, in a specified position on a page, as well as navigation links to other p ages on their website and to external websites. The basics of good design are covered and students will develop their own templates using Notepad as a text editor. Students will decide on a topic for their websites, document their designs and collect suitable text and images. They will then use their HTML templates to create their websites, including a web form.

Computer Science Curriculum Learning Journey

Year 8: Topic summary

How we interleave topics to enable pupils to build and recall knowledge.

KS3 Year 8 Computing - Interleaving Approach

Year 8

Unit 1	Unit 2 Unit 1	Unit 3 Unit 1 / Unit 2
Unit 4 Unit 1 / Unit 2 Unit 3	Unit 5 Unit 1 / Unit 2 Unit 3 / Unit 4	Unit 6 Unit 1 / Unit 2 Unit 3 / Unit 4 Unit 5

Year 8

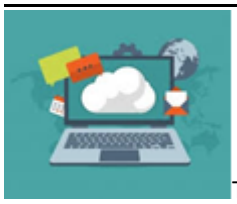
Unit 1	Unit 1 Introduction to Computing							
	Unit 1 Memory Recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4	L1 L2 L3 L4 L5 L6
	Lesson Content	L1. Folder & file management	L2. Email	L3. Office 360	L4. E-Safety	L5. IDEA Award	L6. Searching the Web	L7. Assessment
Unit 2	Unit 2 Scratch Super Mario							
	Unit 2 Memory Recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5	L2 L3 / L1 L2 L3 L4 L5 L6 L8 Unti 1
	Lesson Content	L1 Adding Sprites	L2. Cretaing algorithyms	L3. Mushrooms	L4. Health	L5. Princess Peach	L6. Welcome Screen	L7. Adding Sound / L8. Assessment
Unit 3	Unit 3 Computer Graphics							
	Unit 3 Memory Recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5	L1 L2 L3 L4 L5 L6 / L1 L2 L3 L4 L5 L6 L7 L8 Unit 1 & 2
	Lesson Content	L1. Magic wand & rubber stamp	L2. Lasso & filters	L3. Layers	L4. Vector & bitmap graphics	L5. Vector logo	L6. Movie poster	L7. Text / L8. Assessment
Unit 4	Unit 4 Spreadsheet Modelling							
	Unit 4 Memory Recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5	L1 L2 L3 L4 L5 L6 / L1 L2 L3 L4 L5 L6 L7
	Lesson Content	L1. Computer Modelling	L2. Financial model	L3. What If	L4. Conditional formatting	L5. Charts 7 macros	L6. Practical	L7. IDEA Award / L8. Assessment
Unit 5	Unit 5 Introduction to Python							
	Unit 5 Memory recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5	L1 L2 L3 L4 L5 L6 / L1 L2 L3 L4 L5 L6 L7
	Lesson Content	L1. Strings & variables	L2. Number & arithmetic	L3. Selection	L4. Writing algorithms	L5. While loops	L6. Searching	L7. IDEA Award / L8. Assessment
Unit 6	Unit 6 HTML & Web Development							
	Unit 6 Memory recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5	L1 L2 L3 L4 L5 L6 / L1 L2 L3 L4 L5 L6 L7
	Lesson Content	L1. HTML	L2. CSS	L3. Design	L4. IDEA Award	L5. Development	L6. Web forms	L7. Testing / L8. Assessment

Computer Science Curriculum Learning Journey

Computing Topic Descriptors

Year 8

Unit 1 Introduction to Computing



This unit is an introduction to Computing at Parkside Community School. Students will be introduced to different components to equip them with the basic knowledge and skills when working with Computers. Students begin by looking at file management and security. Students will know how to use the school email system and Office 360 which all students have access to at home. Students will look at online dangers and understand how to protect themselves online and how to effectivity search the Internet and understand that not all information on the Web is accurate or truthful.

Unit 2 Scratch Super Mario

This unit introduces students to computer programming using the software Scratch which is a block-based visual programming language. This unit enables students to create their own "Super Mario" game using the software Scratch. Each lesson builds on the previous lessons work to create a fully playable and interactive game. Students arrive at Parkside with very little previous experience of using Scratch to program.

Unit 3 Computer Graphics

This unit introduces students to the topic Computer Graphics using the software Adobe Fireworks. This will be the first time students have had access to this specialist software. The unit consists of skills builder lessons which introduces students to some of the main graphic editing tools and techniques such as; magic wand tool, rubber stamp tool, lasso and filters, using layers and text. Students will use the skills established to create a logo and a movie poster to showcase their skills.

Computer Science Curriculum Learning Journey

Unit 4 Spreadsheet Modelling

This is a practical, skills-based unit covering the principles of creating and formatting basic spreadsheets to produce and use simple computer models. It is suitable for pupils who have a basic knowledge of spreadsheets including cell references, simple formulae and formatting, although these topics are revised in the first lesson, making it also suitable for pupils new to spreadsheets due to the inconsistencies of ability from KS2.
The unit is centred around creating a financial model for a TV show. Students start by looking at different types of model and then use basic spreadsheet techniques to create and format a simple financial model to calculate the expected income from viewers' voting. The model is then extended to include sales from merchandising, with the introduction of "what if" scenarios. Finally, the students create a seating plan, book seats and calculate income from seat sales. Spreadsheet features covered include SUM, MAX, IF and COUNTIF functions, cell naming for absolute referencing, conditional formatting, validation, charting and simple macros. Microsoft Excel software is used in this unit.

Unit 5 Introduction to Python

This unit of work is an introduction to Python, a powerful but easy-to-use high-level programming language. Although Python is an object-oriented language, at this level the object-oriented features of the language are barely in evidence and do not need to be discussed. The focus is on getting pupils to understand the process of developing programs, the importance of writing correct syntax, being able to formulate algorithms for simple programs and debugging their programs.

Unit 6 HTML & Website Development

Students will learn the basics of HTML and CSS. They will learn how to create text styles and add content, including text and graphics, in a specified position on a page, as well as navigation links to other p ages on their website and to external websites. The basics of good design are covered and students will develop their own templates using Notepad as a text editor. Students will decide on a topic for their websites, document their designs and collect suitable text and images. They will then use their HTML templates to create their websites, including a web form.

Computer Science Curriculum Learning Journey

Year 9: Topic summary

How we interleave topics to enable pupils to build and recall knowledge.

KS3 Year 9 Computing - Interleaving Approach

Year 9

Unit 1	Unit 2 Unit 1	Unit 3 Unit 1 / Unit 2
Unit 4 Unit 1 / Unit 2 Unit 3	Unit 5 Unit 1 / Unit 2 Unit 3 / Unit 4	Unit 6 Unit 1 / Unit 2 Unit 3 / Unit 4 Unit 5

Year 9

Unit 1	Unit 1 Introduction to Python								
	Unit 1 Memory Recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4	L1 L2 L3 L4 L5 L6	L1 L2 L3 L4 L5 L7
	Lesson Content	L1. Strings & variables	L2. Number & arithmetic	L3. Selection	L4. Writing algorithms	L5. While loops	L6. Searching	L7. IDEA Award	L8. Assessment

Unit 2	Unit 2 Understanding Computers							
	Unit 2 Memory Recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5	L1 L2 L3 L4 L5 L6
	Lesson Content	L1. Elements of a Computer	L2. The CPU	L3. Understanding Binary	L4. Binary Addition	L5. Storgae Devices	L6. New Technology	L7. Assessment

Unit 3	Unit 3 First steps in Small Basic							
	Unit 3 Memory Recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5	L1 L2 L3 L4 L5 L6
	Lesson Content	L1. Introducing the Turtle	L2. Using a For loop	L3. The text window	L4. Using variables	L5. Conditions and branching	L6. Using random numbers	L7. Assessment

Unit 4	Unit 4 Python Next Steps						
	Unit 4 Memory Recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5
	Lesson Content	L1. The basics	L2. Loops	L3. Lists	L4. Procedures	L5. Functions	L6. Assessment

Unit 5	Unit 5 Networks						
	Unit 5 Memory recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5
	Lesson Content	L1. The Internet	L2. Connectivity	L3. Topologies	L4. Client/Servers Networks	L5. Encrypition	L6. Assessment

Unit 6	Unit 6 Modelling in Small Basic							
	Unit 6 Memory recall		L1	L1 L2	L1 L2 L3	L1 L2 L3 L4	L1 L2 L3 L4 L5	L1 L2 L3 L4 L5 L6
	Lesson Content	L1. Introduction	L2. Calculations	L3. Selection	L4. Loops	L5. Using subroutines to build a model	L6. Evaluating the model	L7Assessment

Computer Science Curriculum Learning Journey

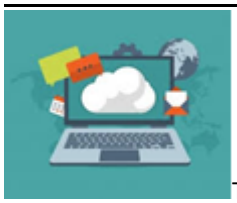
Computing Topic Descriptors

Year 9

Unit 1 Introduction to Python

This unit of work is an introduction to Python, a powerful but easy-to-use high-level programming language. Although Python is an object-oriented language, at this level the object-oriented features of the language are barely in evidence and do not need to be discussed. The focus is on getting pupils to understand the process of developing programs, the importance of writing correct syntax, being able to formulate algorithms for simple programs and debugging their programs.

Unit 2 Understanding Computers



This unit goes beyond an introduction to Computing; the unit is centred around gaining an elementary understanding of how computers work. It gives the students an understanding of what computers are made up of – Input Devices, Output Device, the CPU, and Storage i.e. the "computer architecture".

Further to the introduction to computer architecture, students will be introduced to data; the forms it takes, how it is used, and how it is stored within the computer. The students will then move onto learning how the computer carries out calculations using the binary number system.

The student learning within this unit concludes with a look into the future at new and emerging technology and a chance for the students to use their imaginations to understand and predict new and existing technologies can converge into single devices.

Unit 3 First Steps in Small Basic

Small Basic is a programming language that has been created to allow students to bridge the gap between block-based programming languages such as Scratch and text-based languages such as Python. Small Basic can be considered to be the "student suitable" version of the more powerful BASIC computer language (and its derivative computer languages ".NET" and Visual Basic) used in the computing and other industries.

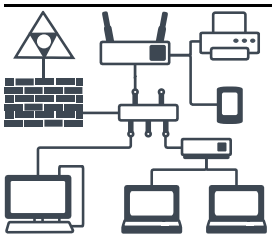
Students will initially visit some of the fundamental programming practices that were visited earlier in the term during the Python unit but will be able to get a perspective of how these fundamentals are handled in another programming language. They will then move onto using Small Basic to cover some of the other elementary programming tasks and techniques such as looping (repetition of a computer command) and Conditions and Branching (building a decision system based upon simple choices).

Unit 4 Python Next Steps

At this point in the learning journey, the students will have had experience of the fundementals of programming in at least three programming languages – Small Basic, Python and Scratch. Now is the time to extend learning in Python to carry out more sophisticated tasks by employing coding techniques and program language components that are new to students.

Within this unit, students will initially refresh their knowledge of Python Programming basics before looking at different (and in some cases more "elegant") ways to carry out tasks they covered in the previous Python and Small Basic units such as looping (repetition of a computer command). Students will then progress onto techniques that cover things such as handling lists of data and reusing code within programs without having to rewrite it.

Unit 5 Networks



Data is covered earlier in the Y9 computing curriculum in terms of how it is used and stored but how data is transported from place to place (within the same building, the same organisation, other organisations, within countries and the world!) is a subject in its own right – this subject is Networks.

This unit covers how data is delivered within organisations and over the internet, the physical hardware we use to transfer it and the speeds that data can be transferred. Students are introduced to network topologies (network designs); the different types of topology and the pros and cons of each, before looking at the classic "Client/Server" model for connecting "everyday" computers to the bigger more powerful computers that hold large amounts of data and processing capacity. Finally, students are introduced to how data has to be encrypted to ensure that valuable information isn't seen or stolen by someone it is not intended for.

Unit 6 Modelling in Small Basic

Following on from the earlier unit introducing Small Basic, we return to this programming language to gain experience of using it in a more sophisticated way.

This unit aims to provide students with an understanding of computer modelling, different modelling variants, and how modelling is applied to predict, understand and analyse real world situations. The students will initially return to the fundementals of Small Basic to recall and refresh their prior knowledge before moving on to learn new or enhance existing skills (such as selection, iteration, subroutines and modular programming) that will enable them to create a computer model.

The unit will conclude with the students gaining experience of how to evaluate their model.

Core Knowledge Concepts

Computer Science Curriculum implementation
Knowledge concepts used to form Schemes of Learning

Computer Science Curriculum Learning Journey

Year 7 Core Knowledge Concepts

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power
Unit 1 Introduction to Computing 	• I can send and reply to emails and send attachments • I can use an email signature • I can identify a probable phishing email and deal with it appropriately • I can explain the advantages and disadvantages of email as a method of communication • I can manage a Contacts list efficiently for email • I know how to access Office 360 at home • I understand the term “cloud” computing • I understand how I can use “OneDrive” • I can create an online document and save it in my OneDrive so I can access at home/school • I can explain what Social Networking means • I can explain what cyberbullying is and how to report it • I can explain why using privacy settings on Social Networking sites is important • I can explain how to create an online profile that will keep me safe • I know what the IDEA Award is and how it can benefit me • I can complete badges to work towards the Bronze Award • I can confidently navigate the IDEA Award website • I understand how the IDEA Award can make me more employable • I can explain what a search engine is and give examples • I can use a search engine to locate information on the internet • I can explain why information on the internet is not always accurate • I can use advanced searches to locate meaningful information	• Folder • File • Email • Phishing • Cloud computing • E-safety • Search engine • Reliable

Computer Science Curriculum Learning Journey

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power

Unit 2
Scratch Super Mario
SCRATCH
	• I can navigate the Scratch environment
	• I can edit and import sprites into Scratch
	• I can add costumes to my sprites
	• I can clearly explain the use of sprites and costumes
	• I can give a sprite basic movement
	• I can use a range of sprites which I can control in different ways
	• I can write code to make a sprite jump
	• I can add comments to my code to clearly explain how it works
	• I can design and create an **algorithm** that:
	  ○ makes the mushrooms disappear when Mario collects them
	  ○ make Mario shoot mushrooms once the space key is pressed
	• I can design and create an **algorithm** that:
	  ○ uses the broadcast function so that Mario knows he has collected mushrooms
	• I can **write like a Computer Scientist** to explain how my algorithms work
	• I can design and create an **algorithm** that:
	  ○ used variables to set the lives for Bowser and Mario
	• I can design and create an **algorithm** that:
	  ○ Change the lives of Mario when Bowser touches him
	  ○ Change the lives of Bowser when a mushroom touches him
	• I can **write like a Computer Scientist** to explain how my algorithms work
	• I can design and create an **algorithm** that:
	  ○ makes a key disappear when Mario touches it
	  ○ remembers Mario has collected the key
	• I can design and create an **algorithm** that:
	  ○ makes princess tell Mario that he needs a key to free her
	  ○ frees Princess when Mario has collected a key
	• I can **write like a Computer Scientist** to explain how my algorithms work
	• I can design and create a welcome screen that:
	  ○ Includes at least two suitable images
	• I can design and create a welcome screen that:
		• Scratch
		• Stage
		• Sprite
		• Costume
		• Script
		• Coordinates
		• Variable
		• Annotate
		• Algorithm

	○ includes text which uses filters • I can write an algorithm so my welcome screen will appear straight away and disappear when the user selects the letter 's' • I can create my own sounds • I can justify my reasons for adding sounds to my game • I can import sounds • My sounds will play automatically at appropriate times throughout my game	

Computer Science Curriculum Learning Journey

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power
Unit 3 Computer Graphics	• I can use the rubber stamp tool and magic wand tool to edit images • I can explain how I can use graphic editing tools to edit different images • I understand the limitations to some of these tools • I understand how images can be manipulated to give different impressions • I can use the lasso tools and filters to edit images • I can demonstrate and explain different tools and techniques in Adobe Fireworks • I understand and can explain the limitations to some editing tools • I can use a range of tools to remove part of an image • I can use layers to add depth with my work • I can manipulate layers to create a final image • I understand how a bitmap graphic is made up of individual pixels • I can manipulate vector and bitmap images • I can compare the different characteristics, strengths and uses of vector and bitmap images • I can create a suitable logo based on a business scenario • I can combine the use of shapes and text to create an effective logo • I can justify the use of tools I have used • I can create a movie poster using Fireworks which included both text and images • I can demonstrate the use of at least three different graphic editing techniques on my movie poster • I can demonstrate the use of layers on my movie poster • My movie poster will be fit for purpose and audience • I can use care and attention when selecting a font to get a particular message across • I understand the importance of consistency in font selection • I understand that using too many different fonts dilutes the message and looks messy • I can create my own font using Fireworks	• Graphic • Edit • Bitmap • Vector • Manipulate • Layer • Pixel • Transparent • Limitations

Computer Science Curriculum Learning Journey

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power
Unit 4 Spreadsheet Modelling Microsoft Excel	• I understand what is meant by the term "**computer model**" • I know spreadsheet basics: entering text, numbers and formulae • I can compare different types of model • I can format cells & insert a graphic • I can use and understand relative and absolute referencing • I can create a financial model to predict the profit on the sale of merchandise • I can make the model as realistic as possible based on known sales figures and prices • I can consider ways of increasing profit to meet a given target • I can use a spreadsheet to model outcomes • I can use **functions** including "**Max**", "**Min**" and "**If**" • I can name a cell • I can sort data into different sequences • I can try out different 'What if' scenarios to achieve a goal • I can create a spreadsheet for a seat booking system and display formulae within it • I can use a **Countif** function in calculations of seat sales • I can use **conditional formatting** to show which seats have been booked • I can use validation techniques to ensure that only valid data can be entered • I can add comments to my work to show an increased level of understanding of the formulas, functions and tool I have used. • I can create a **macro** to reset all seats to unbooked • I can create a pie chart to show the number of seats remaining • I can assign the macro to a button to make the spreadsheet more user friendly • I can customise a chart • I can explain what a macro is and the advantages of using Macros • I can explain the different types of Charts available and how they are used for different types of data	• Computer model • Spreadsheet • Cell • Format • Absolute referencing • Relative referencing • Function • Formulae • Macro • Chart

Computer Science Curriculum Learning Journey

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power
Unit 5 Introduction to Python python™	• I know what Python is and some of the applications it is used for • I can run a simple Python program in Interactive mode using the input and print functions • I can write, save and run a program in Script mode • I understand what a syntax error is and how to interpret an error message • I know the rules for variable names and use variables in a program • I understand the use and value of using comments • I understand the importance of using correct data types: string, integer or float • I can use the int, float and round functions • I understand how to use assignment statements correctly • I can perform arithmetic using the BIDMAS rule • I can write a program involving input, calculation and output • I can use selection statements if, else and elif in a program • I can learn how to use different comparison operators • I can use indentation correctly to define a block of code • I can learn to write algorithms in pseudocode • I can review the difference between syntax errors, run-time errors and logic errors • I can learn techniques for debugging programs • I can use a while loop in a program • I can use an if statement within a while loop • I can use a function to generate a random number • I can compare alternative algorithms for a given problem • I can use a linear search to find a number • I understand how a binary search works	• Python • Programming language • String • Variable • Data types • Selection • Pseudocode • Debug • While loops • Algorithms

Computer Science Curriculum Learning Journey

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power
Unit 6 HTML & Website Development 	• I understand that the WWW is a huge collection of websites all over the world • I know what HTML is and what it is used for • I can type basic HTML tags using a text editor to create a page that can be viewed in a browser • I can edit the HTML code and view the changes in a browser • I know how CSS is used to set the styles in web pages and websites • I can write CSS code to set styles, e.g. background colour of sections of the page; size, font, colour and alignment of text • I understand what is meant by responsive design, and create a responsive web page • I know the main principles of good website design • I can complete website designs and gather content • I can use an HTML template to create consistent web pages • I can use float to position elements on a page • I can learn how to create a consistent look and feel throughout a website • I can add well formatted content, including text and images, to each page • I can create internal and external links and make sure they all work • I can learn how to create a web form • I understand what happens to the input data once it has been submitted • I can complete a self-evaluation of my website. • I can use my self-evaluation and testing to suggest improvements for my website. • I can create and use a test plan	• HTML • Website • Browser • Tag • CSS • Alignment • Responsive design • Template • Web form • Replicate • Consistent

Computer Science Curriculum Learning Journey

Year 8 Core Knowledge Concepts

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power
Unit 1 Introduction to Computing 	• I can send and reply to emails and send attachments • I can use an email signature • I can identify a probable phishing email and deal with it appropriately • I can explain the advantages and disadvantages of email as a method of communication • I can manage a Contacts list efficiently for email • I know how to access Office 360 at home • I understand the term “cloud” computing • I understand how I can use “OneDrive” • I can create an online document and save it in my OneDrive so I can access at home/school • I can explain what Social Networking means • I can explain what cyberbullying is and how to report it • I can explain why using privacy settings on Social Networking sites is important • I can explain how to create an online profile that will keep me safe • I know what the IDEA Award is and how it can benefit me • I can complete badges to work towards the Bronze Award • I can confidently navigate the IDEA Award website • I understand how the IDEA Award can make me more employable • I can explain what a search engine is and give examples • I can use a search engine to locate information on the internet • I can explain why information on the internet is not always accurate • I can use advanced searches to locate meaningful information	• Folder • File • Email • Phishing • Cloud computing • E-safety • Search engine • Reliable

Computer Science Curriculum Learning Journey

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power

Unit 2
Scratch Super Mario
SCRATCH
	• I can navigate the Scratch environment
	• I can edit and import sprites into Scratch
	• I can add costumes to my sprites
	• I can clearly explain the use of sprites and costumes
	• I can give a sprite basic movement
	• I can use a range of sprites which I can control in different ways
	• I can write code to make a sprite jump
	• I can add comments to my code to clearly explain how it works
	• I can design and create an **algorithm** that:
	  ○ makes the mushrooms disappear when Mario collects them
	  ○ make Mario shoot mushrooms once the space key is pressed
	• I can design and create an **algorithm** that:
	  ○ uses the broadcast function so that Mario knows he has collected mushrooms
	• I can **write like a Computer Scientist** to explain how my algorithms work
	• I can design and create an **algorithm** that:
	  ○ used variables to set the lives for Bowser and Mario
	• I can design and create an **algorithm** that:
	  ○ Change the lives of Mario when Bowser touches him
	  ○ Change the lives of Bowser when a mushroom touches him
	• I can **write like a Computer Scientist** to explain how my algorithms work
	• I can design and create an **algorithm** that:
	  ○ makes a key disappear when Mario touches it
	  ○ remembers Mario has collected the key
	• I can design and create an **algorithm** that:
	  ○ makes princess tell Mario that he needs a key to free her
	  ○ frees Princess when Mario has collected a key
	• I can **write like a Computer Scientist** to explain how my algorithms work
	• I can design and create a welcome screen that:
	  ○ Includes at least two suitable images
	• I can design and create a welcome screen that:
		• Scratch
		• Stage
		• Sprite
		• Costume
		• Script
		• Coordinates
		• Variable
		• Annotate
		• Algorithm

	○ includes text which uses filters • I can write an algorithm so my welcome screen will appear straight away and disappear when the user selects the letter 's' • I can create my own sounds • I can justify my reasons for adding sounds to my game • I can import sounds • My sounds will play automatically at appropriate times throughout my game	

Computer Science Curriculum Learning Journey

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power
Unit 3 Computer Graphics	• I can use the rubber stamp tool and magic wand tool to edit images • I can explain how I can use graphic editing tools to edit different images • I understand the limitations to some of these tools • I understand how images can be manipulated to give different impressions • I can use the lasso tools and filters to edit images • I can demonstrate and explain different tools and techniques in Adobe Fireworks • I understand and can explain the limitations to some editing tools • I can use a range of tools to remove part of an image • I can use layers to add depth with my work • I can manipulate layers to create a final image • I understand how a bitmap graphic is made up of individual pixels • I can manipulate vector and bitmap images • I can compare the different characteristics, strengths and uses of vector and bitmap images • I can create a suitable logo based on a business scenario • I can combine the use of shapes and text to create an effective logo • I can justify the use of tools I have used • I can create a movie poster using Fireworks which included both text and images • I can demonstrate the use of at least three different graphic editing techniques on my movie poster • I can demonstrate the use of layers on my movie poster • My movie poster will be fit for purpose and audience • I can use care and attention when selecting a font to get a particular message across • I understand the importance of consistency in font selection • I understand that using too many different fonts dilutes the message and looks messy • I can create my own font using Fireworks	• Graphic • Edit • Bitmap • Vector • Manipulate • Layer • Pixel • Transparent • Limitations

Computer Science Curriculum Learning Journey

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power
Unit 4 Spreadsheet Modelling Microsoft Excel	• I understand what is meant by the term "**computer model**" • I know spreadsheet basics: entering text, numbers and formulae • I can compare different types of model • I can format cells & insert a graphic • I can use and understand relative and absolute referencing • I can create a financial model to predict the profit on the sale of merchandise • I can make the model as realistic as possible based on known sales figures and prices • I can consider ways of increasing profit to meet a given target • I can use a spreadsheet to model outcomes • I can use **functions** including "**Max**", "**Min**" and "**If**" • I can name a cell • I can sort data into different sequences • I can try out different 'What if' scenarios to achieve a goal • I can create a spreadsheet for a seat booking system and display formulae within it • I can use a **Countif** function in calculations of seat sales • I can use **conditional formatting** to show which seats have been booked • I can use validation techniques to ensure that only valid data can be entered • I can add comments to my work to show an increased level of understanding of the formulas, functions and tool I have used. • I can create a **macro** to reset all seats to unbooked • I can create a pie chart to show the number of seats remaining • I can assign the macro to a button to make the spreadsheet more user friendly • I can customise a chart • I can explain what a macro is and the advantages of using Macros • I can explain the different types of Charts available and how they are used for different types of data	• Computer model • Spreadsheet • Cell • Format • Absolute referencing • Relative referencing • Function • Formulae • Macro • Chart

Computer Science Curriculum Learning Journey

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power
Unit 5 Introduction to Python python™	• I know what Python is and some of the applications it is used for • I can run a simple Python program in Interactive mode using the input and print functions • I can write, save and run a program in Script mode • I understand what a syntax error is and how to interpret an error message • I know the rules for variable names and use variables in a program • I understand the use and value of using comments • I understand the importance of using correct data types: string, integer or float • I can use the int, float and round functions • I understand how to use assignment statements correctly • I can perform arithmetic using the BIDMAS rule • I can write a program involving input, calculation and output • I can use selection statements if, else and elif in a program • I can learn how to use different comparison operators • I can use indentation correctly to define a block of code • I can learn to write algorithms in pseudocode • I can review the difference between syntax errors, run-time errors and logic errors • I can learn techniques for debugging programs • I can use a while loop in a program • I can use an if statement within a while loop • I can use a function to generate a random number • I can compare alternative algorithms for a given problem • I can use a linear search to find a number • I understand how a binary search works	• Python • Programming language • String • Variable • Data types • Selection • Pseudocode • Debug • While loops • Algorithms

Computer Science Curriculum Learning Journey

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power
Unit 6 HTML & Website Development 	• I understand that the WWW is a huge collection of websites all over the world • I know what HTML is and what it is used for • I can type basic HTML tags using a text editor to create a page that can be viewed in a browser • I can edit the HTML code and view the changes in a browser • I know how CSS is used to set the styles in web pages and websites • I can write CSS code to set styles, e.g. background colour of sections of the page; size, font, colour and alignment of text • I understand what is meant by responsive design, and create a responsive web page • I know the main principles of good website design • I can complete website designs and gather content • I can use an HTML template to create consistent web pages • I can use float to position elements on a page • I can learn how to create a consistent look and feel throughout a website • I can add well formatted content, including text and images, to each page • I can create internal and external links and make sure they all work • I can learn how to create a web form • I understand what happens to the input data once it has been submitted • I can complete a self-evaluation of my website. • I can use my self-evaluation and testing to suggest improvements for my website. • I can create and use a test plan	• HTML • Website • Browser • Tag • CSS • Alignment • Responsive design • Template • Web form • Replicate • Consistent

Computer Science Curriculum Learning Journey

Year 9 Core Knowledge Concepts

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power
Unit 1 Introduction to Python python™	• I know what Python is and some of the applications it is used for • I can run a simple Python program in Interactive mode using the input and print functions • I can write, save and run a program in Script mode • I understand what a syntax error is and how to interpret an error message • I know the rules for variable names and use variables in a program • I understand the use and value of using comments • I understand the importance of using correct data types: string, integer or float • I can use the int, float and round functions • I understand how to use assignment statements correctly • I can perform arithmetic using the BIDMAS rule • I can write a program involving input, calculation and output • I can use selection statements if, else and elif in a program • I can learn how to use different comparison operators • I can use indentation correctly to define a block of code • I can learn to write algorithms in pseudocode • I can review the difference between syntax errors, run-time errors and logic errors • I can learn techniques for debugging programs • I can use a while loop in a program • I can use an if statement within a while loop • I can use a function to generate a random number • I can compare alternative algorithms for a given problem • I can use a linear search to find a number • I understand how a binary search works	• Python • Programming language • String • Variable • Data types • Selection • Pseudocode • Debug • While loops • Algorithms

Computer Science Curriculum Learning Journey

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power
Unit 2 Understanding Computers 	• I can distinguish between hardware and software • I can give examples of computer hardware and software • I can understand that a computer system cannot truly operate without data • I can explain how data flows in and out of a computer • I can understand that a computer processes data • I can explain the primary function of the CPU, input, output and storage devices • I can draw a block diagram showing CPU, input, output and storage devices • I can name different types of permanent storage device • I can suggest appropriate input and output devices for a simple scenario • I can explain what RAM and ROM are used for • I can show how numbers and text can be represented in binary • I can explain the impact of future technologies • I can perform simple binary arithmetic • I can state strengths and weaknesses of different storage devices • I can describe briefly how data is stored on a CD • I can explain how characters are encoded using the ASCII system • I can use an ASCII reference chart to convert a character into binary and its decimal equivalent	• Data • Input • Output • Device • CPU • Input Device • Output Device • Storage Device • RAM • ROM • Binary

Computer Science Curriculum Learning Journey

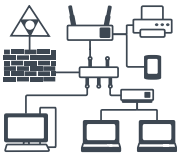
Unit	Student Learning Outcomes	Disciplinary Literacy Word Power
Unit 3 First steps in Small Basic	• I have explored and understood the parts of the Small Basic window • I have written and run a Small Basic program using turtle graphics • I have used Intellisense to type programs faster • I know how to save a program • I can use a "For..End" For loop to repeat a series of instructions several times • I have drawn different shapes using a "For..End" For loop • I have entered text in the Text window that is displayed • I understand what variables are and have used them in a program • I have used variables within a Small Basic program to create a shape using turtle graphics • I have used the selection statements "If…Then" and "If…Then…Else" to choose which statements within a program script will be performed • I understand how to use and have used the RandomNumber function to generate random coordinates • I understand how to use and have used the RandomColor function to get a random colour • I understand how to use and have used a "While…End" While loop	• Small Basic • Programming language • String • Variable • Data types • Selection • "If…Then" • "If…Then…Else" • While loops • "For…End" • Random

Computer Science Curriculum Learning Journey

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power

Unit 4
Python Next Steps
python™
	• I can recall the fundamental Python concepts that I learnt earlier in the curriculum, for example:
	  ○ I can correctly read and understand an existing Python program
	  ○ I can recall different data types
	  ○ I can use the int(), float() and str() functions to convert data types
	  ○ I can write an if-else statement
	• I can use a “while” loop to repeat a section of code
	• I can use a “for” loop to repeat a section of code
	• I can make a choice about which loop to use and explain that choice
	• I understand how and am able to store and update values in a list
	• I understand how and am able to append data to a list
	• I understand how and am able to use a “for” loop to step through a list
	• I understand why using a list can be more efficient than using single variables
	• I understand what a procedure is
	• I understand how and am able to define and call a procedure
	• I understand why procedures are useful
	• I understand how and am able to use parameters in a procedure
	• I understand what a function is
	• I understand how and am able to define a function
	• I understand how and am able to call a function and capture the return value
		• Python
		• “while” loop
		• “for” loop
		• List
		• Values
		• Append
		• Procedure
		• Function

Computer Science Curriculum Learning Journey

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power
Unit 5 Networks 	• I know what the Internet and World Wide Web are and how they are different • I understand how web addresses are constructed • I know what a protocol is and why one is needed for data communication • I understand how packet switching works • I know what a Domain Name Server (DNS) is and understand what it does • I know the meaning and understand significance of bandwidth • I can compare different types of network cable and the relative speeds of data transmission for each cable type • I know how to calculate time taken to download files of different sizes at different bandwidths • I understand what is meant by buffering and why it is used • I understand the difference between LANs and WANs • I am able to give an example of each type of network • I know what extra hardware is needed for a LAN to operate • I can identify three different network topologies – bus, ring and star • I understand what constitutes a client-server network • I can contrast a client-server network with a peer-to-peer network • I am able to list advantages and disadvantages of each network model • I understand what is meant by cloud computing • I can list the main advantages of cloud computing • I am able to identify some of the types of data that need to be kept secure • I know some of the ways in which data is kept secure • I can appreciate that unauthorised people can break ciphers and read encrypted data • I know some classical encryption techniques	• Data • Network • Domain • Domain Name Server (DNS) • Topology • LAN • WAN • Cable • Cloud • Encryption

Computer Science Curriculum Learning Journey

Unit	Student Learning Outcomes	Disciplinary Literacy Word Power

Unit 6
Modelling in Small Basic
	• I can recall the fundamental Small Basic concepts that I learnt earlier in the curriculum, for example:
	  ○ I have explored and understood the parts of the Small Basic window
	  ○ I have written and run a Small Basic program using turtle graphics
	  ○ I have used Intellisense to type programs faster
	  ○ I know how to save a program
	  ○ I can recall different data types
	  ○ I can use the int(), float() and str() functions to convert data types
	  ○ I can write an if-else statement
	  ○ I have written and run a Small Basic program using inputs, outputs and variables
	• I can understand what is meant by computer modelling
	• I can perform calculations in Simple Basic
	• I can use mathematical functions in a program
	• I know how to generate random numbers within a given range
	• I understand how to use Comparison operators “=”, “>”, “<”, “>=”, “<=”, “<>”
	• I understand how to use Logical operators “And” and “Or”
	• I understand how to use selection statements to make decisions
	• I understand the limitations of a model
	• I can use a While loop to repeat a section of code until a condition becomes false
	• I can use a For loop to repeat a section of code a given number of times
	• I have created a model of a dice-rolling experiment
	• I understand the concept of a subroutine
	• I have created a model of a Lemonade Stand
	• I can determine what subtasks are required in the model
	• I can write a program incorporating the subtasks as subroutines
	• I can suggest how the reliability of the model could be evaluated
	• I have suggested ways of improving the model
		• Model
		• Modelling
		• “while” loop
		• “for” loop
		• Procedure
		• Function
		• Subroutine