



St Antony's
Roman Catholic School
ASPIRE + BELIEVE + ACHIEVE

Subject: Computer Science

Year group	Topics and resources
Year 7	<p data-bbox="383 499 573 531">Spring Term 1</p> <p data-bbox="383 539 880 571">Unit 7.3 - Gaining support for a cause</p> <p data-bbox="383 579 1989 675">During this unit, learners develop their understanding of information technology and digital literacy skills. They will use the skills learnt across the unit to create a blog post about a real-world cause that they would like to gain support for. Learners will develop software formatting skills and explore concerns surrounding the use of other people's work, including licensing and legal issues.</p> <p data-bbox="383 683 801 715">Link to lessons on Oak Academy</p> <p data-bbox="383 722 1402 754">https://classroom.thenational.academy/units/gaining-support-for-a-cause-33b8.html</p> <p data-bbox="383 834 1155 866"><u>National Curriculum Statement Link – Students will learn to</u></p> <ul data-bbox="432 874 2022 1066" style="list-style-type: none"><li data-bbox="432 874 2022 986">• undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users<li data-bbox="432 994 2022 1066">• create, re-use, revise and re-purpose digital artefacts for a given audience, with attention to trustworthiness, design and usability. <p data-bbox="383 1114 573 1145">Spring Term 2</p> <p data-bbox="383 1153 976 1185">Unit 7.4 – Programming Essentials in Scratch</p> <p data-bbox="383 1233 2022 1329">This unit is the first programming unit of KS3. The aim of this unit and the following unit ('programming 2') is to build learners' confidence and knowledge of the key programming constructs. Importantly, this unit does not assume any previous programming experience, but it does offer learners the opportunity to expand on their knowledge throughout the unit.</p>

The main programming concepts covered in this unit are sequencing, variables, selection, and count-controlled iteration. All of the examples and activities for this unit use Scratch 3. data transmission. Learners will develop an understanding of the terms ‘internet’ and ‘World Wide Web’, and of the key services and protocols used. Practical exercises are included throughout to help strengthen understanding.

Link to lessons on Oak Academy

<https://classroom.thenational.academy/units/programming-essentials-in-scratch-part-i-b4aa.html>

National Curriculum Statement Link – Students will learn to

- understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem
- use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions
- understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]
- create, re-use, revise and re-purpose digital artefacts for a given audience, with attention to trustworthiness, design and usability

Home work

This is based around the badges on idea.org.uk. If a student is unable to access the curriculum due to the specialist software used, then they can complete more badges related to this qualification.

Year 8

Spring Term 1

Unit 8.3 – Mobile App Development

In a world where there’s an app for every possible need, this unit aims to take the learners from designer to project manager to developer in order to create their own mobile app. Using App Lab from code.org, learners will familiarise themselves with the coding

environment and have an opportunity to build on the programming concepts they used in previous units before undertaking their project. Learners will work in pairs to consider the needs of the user; decompose the project into smaller, more manageable parts; use the pair programming approach to develop their app together; and finish off by evaluating the success of the project against the needs of the user.

Link to lessons in Oak Academy:

<https://classroom.thenational.academy/units/mobile-app-development-4fbe.html>

[National Curriculum Statement Link – Students will learn to](#)

- Design, use, and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
- Use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables, or arrays]; design and develop modular programs that use procedures or functions
- Understand several key algorithms that reflect computational thinking; use logical reasoning to compare the utility of alternative algorithms for the same problem
- Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability

Spring Term 2

Unit 8.2 – Development for the Web

In this unit, learners will explore the technologies that make up the internet and World Wide Web. Starting with an exploration of the building blocks of the World Wide Web, HTML, and CSS, learners will investigate how websites are catalogued and organised for effective retrieval using search engines. By the end of the unit, learners will have a functioning website.

Link to lessons in Oak Academy:

<https://classroom.thenational.academy/units/networks-2-d74a.html>

[National Curriculum Statement Link – Students will learn to](#)

	<ul style="list-style-type: none"> • Create, reuse, revise, and repurpose digital artefacts for a given audience, with attention to trustworthiness, design, and usability. <p><u>Home work</u></p> <p>This is based around the badges on idea.org.uk. If a student is unable to access the curriculum due to the specialist software used then they can complete more badges related to this qualification.</p>
<p>Year 9</p>	<p>Spring Term 1 Unit 9.3 – Data Science</p> <p>This unit introduces learners to how data can be represented and processed in sequences, such as lists and strings. The lessons cover a spectrum of operations on sequences of data, that range from accessing an individual element to manipulating the entire sequence. Great care has been taken so that the selection of problems used in the programming tasks are realistic and engaging: learners will process solar system planets, book texts, capital cities, leaked passwords, word dictionaries, ECG data, and more.</p> <p>A range of pedagogical tools are employed throughout the unit, with the most prominent being pair programming, live coding, and worked examples.</p> <p>Link to lessons in Oak Academy: https://classroom.thenational.academy/units/python-programming-with-sequences-of-data-7716</p> <p><u>National Curriculum Statement Link – Students will learn to</u></p> <ul style="list-style-type: none"> • design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems • understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem • use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions

- understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]
- understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits

Spring Term 2

Unit 9.4 – Representations - Going Audio Visual

In this unit, learners will focus on digital media such as images and sounds, and discover the binary digits that lie beneath these types of media.

Just like in the previous unit, where learners examined characters and numbers, the ideas that learners need to understand are not really new to them. You will draw on familiar examples of composing images out of individual elements, mixing elementary colours to produce new ones, and taking samples of analogue signals, to illustrate these ideas and bring them together in a coherent narrative.

This unit also has a significant practical aspect. Learners will use relevant software (GIMP and Audacity, in this case) to manipulate images and sounds and get an idea of how the underlying principles of digital representations are applied in real settings.

This unit builds on the material from the Year 8 unit, 'Representations: from clay to silicon'.

Link to lessons in Oak Academy:

<https://classroom.thenational.academy/units/representations-going-audiovisual-5107>

National Curriculum Statement Link – Students will learn to

- Understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits

Home work

This is based around the badges on idea.org.uk. If a student is unable to access the curriculum due to the specialist software used then they can complete more badges related to this qualification.

Year group	Topics and resources	Revision resources (Revision Guides, PPTs etc)	Exam resources (exam board links, mark schemes, model answers, past papers)
Year 10	<p>Specification J277 https://www.ocr.org.uk/Images/558027-specification-gcse-computer-science-j277.pdf</p> <p>Course YouTube Revision CraigNDave Channel</p> <p>Seneca Learning https://senecalearning.com/</p> <p>Smart Revise www.smartrevise.co.uk</p> <p>Term 2a students will study the following units alongside Programming Skills in Python:</p> <p>1.3.1 – Network and Topologies</p> <ul style="list-style-type: none"> • The characteristics of LANs and WANs including common examples of each • Understanding of different factors that can affect the performance of a network, e.g.: • Number of devices connected • Bandwidth • The tasks performed by each piece of hardware • The concept of the Internet as a network of computer networks • A Domain Name Service (DNS) is made up of multiple Domain Name Servers • A DNS’s role in the conversion of a URL to an IP address 	<p>Revision Materials can be found all on Microsoft Teams</p> <p>Revision Aids</p> <ul style="list-style-type: none"> • GCSEPOD • Seneca Learning • SmartRevise • Revision Sheets • Youtube • Quizlet.com • Quizziz.com 	<p>See Microsoft Teams</p>

	<ul style="list-style-type: none"> • Concept of servers providing services (e.g. Web server " Web pages, File server " file storage/retrieval) • Concept of clients requesting/using services from a server • The Cloud: remote service provision (e.g. storage, software, processing) • Advantages and disadvantages of the Cloud • Advantages and disadvantages of the Star and Mesh topologies • Apply understanding of networks to a given scenario <p>1.3.2 – Wired and Wireless Networks, Protocols and Layers</p> <ul style="list-style-type: none"> • Compare benefits and drawbacks of wired versus wireless connection • Recommend one or more connections for a given scenario • The principle of encryption to secure data across network connections • IP addressing and the format of an IP address (IPv4 and IPv6) • A MAC address is assigned to devices; its use within a network • The principle of a standard to provide rules for areas of computing • Standards allows hardware/software to interact across different manufacturers/producers • The principle of a (communication) protocol as a set of rules for transferring data • That different types of protocols are used for different purposes • The basic principles of each protocol i.e. its purpose and key features • How layers are used in protocols, and the benefits of using layers; e.g. the 4-layer TCP/IP model <p>Term 2B Unit 1.4 – Network Security 1.4.1 - Threats to Networks and Computer Systems</p> <ul style="list-style-type: none"> • Threats posed to devices/systems • Knowledge/principles of each form of attack including: 		
--	--	--	--

	<ul style="list-style-type: none"> • How the attack is used • The purpose of the attack <p>1.4.2 – Identifying and Preventing Vulnerabilities</p> <ul style="list-style-type: none"> • Understanding of how to limit the threats posed in 1.4.1 • Understanding of methods to remove vulnerabilities • Knowledge/principles of each prevention method: • What each prevention method may limit/prevent • How it limits the attack 																																		
<p>Year 11</p>	<p>Specification J277 https://www.ocr.org.uk/Images/558027-specification-gcse-computer-science-j277.pdf</p> <p>For lessons students can continue their studies using the CraigNDave youtube channel which is linked at the bottom.</p> <p>Programming Fundamentals 2.2.1 – Programming Fundamentals</p> <ul style="list-style-type: none"> • Practical use of the techniques in a high-level language within the classroom • Understanding of each technique • Recognise and use the following operators: <table border="1" data-bbox="365 1134 864 1378"> <thead> <tr> <th colspan="2">Comparison operators</th> <th colspan="2">Arithmetic operators</th> </tr> </thead> <tbody> <tr> <td>==</td> <td>Equal to</td> <td>+</td> <td>Addition</td> </tr> <tr> <td>!=</td> <td>Not equal to</td> <td>-</td> <td>Subtraction</td> </tr> <tr> <td><</td> <td>Less than</td> <td>*</td> <td>Multiplication</td> </tr> <tr> <td><=</td> <td>Less than or equal to</td> <td>/</td> <td>Division</td> </tr> <tr> <td>></td> <td>Greater than</td> <td>MOD</td> <td>Modulus</td> </tr> <tr> <td>>=</td> <td>Greater than or equal to</td> <td>DIV</td> <td>Quotient</td> </tr> <tr> <td></td> <td></td> <td>^</td> <td>Exponentiation (to the power)</td> </tr> </tbody> </table>	Comparison operators		Arithmetic operators		==	Equal to	+	Addition	!=	Not equal to	-	Subtraction	<	Less than	*	Multiplication	<=	Less than or equal to	/	Division	>	Greater than	MOD	Modulus	>=	Greater than or equal to	DIV	Quotient			^	Exponentiation (to the power)	<p>Revision Materials can be found all on Microsoft Teams</p> <p>Revision Aids</p> <ul style="list-style-type: none"> • GCSEPOD • Seneca Learning • SmartRevise • Revision Sheets • Youtube • Quizlet.com • Quizziz.com 	<p>See Microsoft Teams</p>
Comparison operators		Arithmetic operators																																	
==	Equal to	+	Addition																																
!=	Not equal to	-	Subtraction																																
<	Less than	*	Multiplication																																
<=	Less than or equal to	/	Division																																
>	Greater than	MOD	Modulus																																
>=	Greater than or equal to	DIV	Quotient																																
		^	Exponentiation (to the power)																																

2.2.2 – Data Types

- Practical use of the data types in a high-level language within the classroom
- Ability to choose suitable data types for data in a given scenario
- Understand that data types may be temporarily changed through casting, and where this may be useful

2.2.3 – Additional Programming Techniques

- Practical use of the additional programming techniques in a high-level language within the classroom
- Ability to manipulate strings, including:
 - Concatenation
 - Slicing
- Arrays as fixed length or static structures
- Use of 2D arrays to emulate database tables of a collection of fields, and records
- The use of functions
- The use of procedures
- Where to use functions and procedures effectively
- The use of the following within functions and procedures:
 - local variables/constants
 - global variables/constants
 - arrays (passing and returning)
- SQL commands:
 - SELECT
 - FROM
 - WHERE
- Be able to create and use random numbers in a program

Spring Term 2

Unit 2.3 - Producing Robust Programs

2.3.1 - Defensive Design

- Defensive design considerations:
 - Anticipating misuse
 - Authentication
- Input validation
- Maintainability:
 - Use of sub programs
 - Naming conventions
 - Indentation
 - Commenting

2.3.2 – Testing

- The purpose of testing
- Types of testing:
 - Iterative
 - Final/terminal
- Identify syntax and logic errors
- Selecting and using suitable test data:
 - Normal
 - Boundary
 - Invalid/Erroneous
- Refining algorithms