

St Mary's CE High School Curriculum Map 2022-23



Subject: Computer Science

Year: 11

Providing students with the knowledge, understanding and skills necessary to prepare them for employment or higher education

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
<p>CONTENT</p> <p><i>Declarative / core / powerful Knowledge – 'Know What'</i></p>	<p>Algorithms:</p> <ul style="list-style-type: none"> - Computational Thinking <ul style="list-style-type: none"> o Principles of computational thinking - Designing, creating and refining algorithms <ul style="list-style-type: none"> o Identify the inputs, processes and outputs for a problem o Structure diagrams o Create, interpret, correct, complete and refine algorithms using: Pseudocode, Flowcharts, Reference Language o Identify common errors o Trace tables - Searching and sorting algorithms <ul style="list-style-type: none"> o Standard searching algorithms (Binary & Linear) o Standard sorting algorithms (Bubble, Merge & Insertion) 	<p>Programming Fundamentals:</p> <ul style="list-style-type: none"> - Programming fundamentals <ul style="list-style-type: none"> - The use of variables, constants, operators, inputs, outputs and assignments - The use of the three basic programming constructs used to control the flow of a program: <ul style="list-style-type: none"> o Sequence o Selection o Iteration (count- and condition-controlled loops) - The common arithmetic operators - The common Boolean operators AND, OR and NOT - The use of data types: <ul style="list-style-type: none"> o Integer o Real o Boolean o Character and string o Casting - The use of basic string manipulation - The use of basic file handling operations: <ul style="list-style-type: none"> o Open 	<p>Producing Robust Programs:</p> <ul style="list-style-type: none"> - Defensive Design <ul style="list-style-type: none"> - Defensive design considerations: <ul style="list-style-type: none"> o Anticipating misuse o Authentication - Input validation - Maintainability: <ul style="list-style-type: none"> o Use of sub programs o Naming conventions o Indentation o Commenting - Testing <ul style="list-style-type: none"> - The purpose of testing - Types of testing: <ul style="list-style-type: none"> o Iterative o Final/terminal - Identify syntax and logic errors - Selecting and using suitable test data: <ul style="list-style-type: none"> o Normal o Boundary o Invalid/Erroneous - Refining algorithms 	<p>Boolean Logic –</p> <ul style="list-style-type: none"> - Boolean Logic <ul style="list-style-type: none"> - Simple logic diagrams using the operators AND, OR and NOT - Truth tables - Combining Boolean operators using AND, OR and NOT - Applying logical operators in truth tables to solve problems 	<p>Programming Languages and Integrated Development Environments:</p> <ul style="list-style-type: none"> - Languages <ul style="list-style-type: none"> - Characteristics and purpose of different levels of programming language: <ul style="list-style-type: none"> o High-level languages o Low-level languages - The purpose of translators - The characteristics of a compiler and an interpreter - Common tools and facilities available in an Integrated Development Environment (IDE): <ul style="list-style-type: none"> o Editors o Error diagnostics o Run-time environment o Translators 	<p>Content revision and recap in preparation for final exams</p>

St Mary's CE High School Curriculum Map 2022-23



Subject: Computer Science

Year: 11

Providing students with the knowledge, understanding and skills necessary to prepare them for employment or higher education

		<ul style="list-style-type: none"> ○ Read ○ Write ○ Close - The use of records to store data - The use of SQL to search for data - The use of arrays (or equivalent) when solving problems, including both one-dimensional (1D) and two-dimensional arrays (2D) - How to use sub programs (functions and procedures) to produce structured code - Random number generation 				
<p>Skills</p> <p><i>Procedural Knowledge – ‘Know How’</i></p>	<p>Understanding of these principles and how they are used to define and refine problems</p> <p>Produce simple diagrams to show:</p> <ul style="list-style-type: none"> -The structure of a problem -Subsections and their links to other subsections <p>Complete, write or refine an algorithm using the techniques listed</p> <p>Identify syntax/logic errors in code and suggest fixes</p>	<p>Practical use of the techniques in a high-level language within the classroom</p> <p>Understanding of each technique</p> <p>Recognise and use the following operators:</p> <ul style="list-style-type: none"> - == equal to - != not equal to - < less than - <= less than or equal to - > greater than 	<p>Understanding of the issues a programmer should consider to ensure that a program caters for all likely input values</p> <p>Understanding of how to deal with invalid data in a program</p> <p>Authentication to confirm the identity of a user</p> <p>Practical experience of designing input validation and simple authentication (e.g. username and password)</p> <p>Understand why commenting</p>	<p>Knowledge of the truth tables for each logic gate</p> <p>Recognition of each gate symbol</p> <p>Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios</p> <p>Ability to work with more than one gate in a logic diagram</p>	<p>The differences between high- and low-level programming languages</p> <p>The need for translators</p> <p>The differences, benefits and drawbacks of using a compiler or an interpreter</p> <p>Knowledge of the tools that an IDE provides</p> <p>How each of the tools and facilities listed can be used to help a programmer develop a program</p>	



Subject: Computer Science
Year: 11

Providing students with the knowledge, understanding and skills necessary to prepare them for employment or higher education

	<p>Create and use trace tables to follow an algorithm Understand the main steps of each algorithm Understand any pre-requisites of an algorithm Apply the algorithm to a data set Identify an algorithm if given the code or pseudocode for it</p>	<ul style="list-style-type: none"> - >= greater than or equal to - + addition - - subtraction - * multiplication - / division - MOD modulus - DIV quotient - ^ exponentiation (to the power) <p>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 Practical use of the additional programming techniques in a high-level language within the classroom Ability to manipulate strings, including:</p> <ul style="list-style-type: none"> - Concatenation - Slicing <p>Arrays as fixed length or static structures Use of 2D arrays to emulate database tables of a</p>	<p>is useful and apply this appropriately The difference between testing modules of a program during development and testing the program at the end of production Syntax errors as errors which break the grammatical rules of the programming language and stop it from being run/translated Logic errors as errors which produce unexpected output Normal test data as data which should be accepted by a program without causing errors Boundary test data as data of the correct type which is on the very edge of being valid Invalid test data as data of the correct data type which should be rejected by a computer system Erroneous test data as data of the incorrect data type which should be rejected by a computer system Ability to identify suitable test data for a given scenario Ability to create/complete a test plan</p>	<table border="1"> <tr> <th>Boolean Operators</th> <th>Logic Gate Symbol</th> </tr> <tr> <td>AND (Conjunction)</td> <td></td> </tr> <tr> <td>OR (Disjunction)</td> <td></td> </tr> <tr> <td>NOT (Negation)</td> <td></td> </tr> </table> <p>Truth Tables:</p> <table border="1"> <thead> <tr> <th colspan="3">AND</th> </tr> <tr> <th>A</th> <th>B</th> <th>A AND B</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="3">OR</th> </tr> <tr> <th>A</th> <th>B</th> <th>A OR B</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2">NOT</th> </tr> <tr> <th>A</th> <th>NOT A</th> </tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	Boolean Operators	Logic Gate Symbol	AND (Conjunction)		OR (Disjunction)		NOT (Negation)		AND			A	B	A AND B	0	0	0	0	1	0	1	0	0	1	1	1	OR			A	B	A OR B	0	0	0	0	1	1	1	0	1	1	1	1	NOT		A	NOT A	0	1	1	0	<p>Practical experience of using a range of these tools within at least one IDE</p>	
Boolean Operators	Logic Gate Symbol																																																									
AND (Conjunction)																																																										
OR (Disjunction)																																																										
NOT (Negation)																																																										
AND																																																										
A	B	A AND B																																																								
0	0	0																																																								
0	1	0																																																								
1	0	0																																																								
1	1	1																																																								
OR																																																										
A	B	A OR B																																																								
0	0	0																																																								
0	1	1																																																								
1	0	1																																																								
1	1	1																																																								
NOT																																																										
A	NOT A																																																									
0	1																																																									
1	0																																																									

St Mary's CE High School Curriculum Map 2022-23



Subject: Computer Science

Year: 11

Providing students with the knowledge, understanding and skills necessary to prepare them for employment or higher education

		<p>collection of fields, and records</p> <p>The use of functions The use of procedures Where to use functions and procedures effectively</p> <p>The use of the following within functions and procedures:</p> <ul style="list-style-type: none"> - local variables/constants - global variables/constants - arrays (passing and returning) <p>SQL commands:</p> <ul style="list-style-type: none"> - SELECT - FROM - WHERE <p>Be able to create and use random numbers in a program</p>				
Key Questions	<p>What are the different types of computational thinking and what do they do?</p> <p>What do the different symbols of a flow diagram mean?</p> <p>What is the difference between a Binary and a Linear search?</p> <p>What do the following searches do: Bubble, Merge & Insertion?</p>	<p>What is the different fundamentals used within programming?</p> <p>What are the three basic programming constructs used to control the flow of a program?</p> <p>What are the common arithmetic operators used in programming?</p> <p>What are the differences between the different Boolean operators?</p>	<p>What are the defensive design considerations that need to be considered?</p> <p>What is input validation and why is it used?</p> <p>How do we maintain programs?</p> <p>What is the purpose of testing?</p> <p>What are the different types of testing?</p>	<p>What is the difference between the different Boolean logic operators?</p> <p>How do we use truth tables?</p> <p>How do we calculate the number of columns and rows needed for a truth table?</p> <p>How do we combine Boolean operators?</p> <p>How can we apply logical operators to truth tables?</p>	<p>What are the different characteristics between high- and low- level programming languages?</p> <p>What is the purpose of translators?</p> <p>What are the characteristics of a compiler?</p> <p>What are the characteristics of an interpreter?</p> <p>What are the common tools and facilities available within</p>	

St Mary's CE High School Curriculum Map 2022-23



Subject: Computer Science

Year: 11

Providing students with the knowledge, understanding and skills necessary to prepare them for employment or higher education

		<p>What are the different data types and what do they do? How can we manipulate data? What are the basic file handling operations? How can we use records to store data? What does SQL stand for? How can we use arrays to solve problems?</p>	<p>What is the difference between syntax and logic errors? How do we refine algorithms?</p>		<p>an Integrated Development Environment?</p>	
Assessment	<p>End of Algorithms assessment including previous modules</p>	<p>End of Programming Fundamentals assessment including previous modules</p>	<p>End of Producing Robust Programs assessment including previous modules</p>	<p>End of Boolean assessment including previous modules</p>	<p>End of Programming languages and Integrated Development Environments assessment including previous modules</p>	