



Academic Year <u>Year; 7</u>	Content. Unit title and brief outline of content.	Skills taught in each unit.	Assessment – what knowledge and skills will be assessed and how?
Rationale	The modules taught below support students learning from KS2 of IT/CS or non to help built pupils skills in the subject. It also provides a taster of Computer Science GCSE. The Computer Science Pathway map is used to identify levels.		
Autumn A	Internet Safety and MS Teams	Key feature use in Teams Being able to make alerts of incidents	Knowledge on how to stay safe online will be assessed as well as how MS Teams can be used as a tool to support pupils though their education at the Ursuline.
Autumn B	Using IT to effectively Learn (Bit pattern)(History of CS)(Webquest using search engines)	Use of MS Office How to email/structure Converting binary to denary and vice verse	Pupils are assessed on Hardware/software and well as the binary numbering system. Advanced Searching skills using a browser are assessed especially to see if pupils can use keywords or element items from a search.
Spring A	Algorithms using Flowol	Decomposing Problem solving Algometric thinking	Pupils are assessed on algometric thinking skills to see how they create flowcharts based on scenarios as well how they can decompose a situation.
Spring B	Programming with Scratch	Sequencing of instructions Problem solving Decomposing Evaluating Genralisation	Pupils are assessed on a practice basis and observed to see how and what they create using the Scratch software. Pupils are assessed on the skills listed in the skills section for this module.
Summer A	Programming with Scratch and ethics	Arguing a point Evaluate	Pupils are assessed on the use of advance skills in scratch and how



		Explain and justify Debate	they can decompose a situation. Pupils are assessed on how they can argue an ethical and moral situation that technology brings through exam style questions from a GCSE paper.
Summer B	Internet of Things and Microbits	Compare Discuss Justify Suggest Problem solving Algometric thinking Decomposing Evaluating Genralisation	Pupils are assessed on key words from the module and there understanding of what 'The Internet of Things' is. Pupils are practically assessed on how they use a BBC Microbit to tackle a given scenario and observation notes are made of their creation allowing assessment to be based on what they produce and steps taken to produce the given scenario.



Academic Year <u>Year; 8</u>	Content. Unit title and brief outline of content.	Skills taught in each unit.	Assessment – what knowledge and skills will be assessed and how?
Rationale	The modules taught below help built pupils skills in the subject as a development from year 7. It also provides a taster of Computer Science GCSE allowing them to engage and take interest within the subject and explore technology used in the real world. The Computer Science Pathway map is used to identify levels. Pupils in year 8 are provided with key skills required to move into IT courses at KS4/KS5 providing a choice of course at a higher level and a taster of CS/IT.		
Autumn A	Online Safety and MS Teams	Key feature use in Teams Being able to make alerts of incidents	Knowledge on how to stay safe online will be assessed as well as how MS Teams can be used as a tool to support pupils though their education at the Ursuline.
Autumn B	Introduction to Databases and Plagiarism (Using search engines webquest)	Creating a database Creating a query SQL writing Creating a form/report	Pupils are tested on keywords relating to Database as well as how they use sources to reference work they produce.
Spring A	Programming in Python with Logic Gates	Sequencing of instructions Problem solving Decomposing Evaluating Generalisation	Pupils are assessed on materials they produce using python code as well as there understanding of logic gates.
Spring B	Programming in Python with Sorting	Sequencing of instructions Problem solving Decomposing Evaluating Generalisation	Pupils are assessed on materials they produce using python code and assessed on if they can remember the types of sorts a program can produce and the algorithm for them.
Summer A	Ethic and Cryptography and Data Representation	Arguing a point Evaluate Explain and justify	Pupils are assessed on how they encrypt and decrypt data as well as how data in an image form is



		<p>Debate Problem solving Generalisation</p>	<p>created using machine code. Pupils are assessed on how they can argue an ethical and moral situation that technology brings through exam style questions from a GCSE paper.</p>
<p>Summer B</p>	<p>Networks and Internet, Website Design & Development (HTML/CSS) and Compression</p>	<p>Problem solving Decomposing Generalisation</p>	<p>Pupils are assessed on the theory aspect of the different types of network you can produce as well as how the internet works. Pupils make a website using HTML/CSS code and are assessed on the website functionality and user interface.</p>



Academic Year <u>Year; 9</u>	Content. Unit title and brief outline of content.	Skills taught in each unit.	Assessment – what knowledge and skills will be assessed and how?
Rationale	The modules taught below help built pupils skills in the subject as a development from year 8. It is also a chance to start Computer Science GCSE modules allowing pupils to be ready to move forward with GCSE if they pick the subject as a choice. The Computer Science Pathway map and GCSE levels is used to identify levels. (OCR J277)		
Autumn A	Algorithms (GCSE 2.1) – Abstraction/Decomposition/Algorithmic thinking	Abstraction Decomposition Algorithmic thinking	Assessment on the understanding of key words and its use
Autumn B	Algorithms (GCSE 2.1) – Designing, creating and refining algorithms. Identifying inputs/processes/ and output for a problem Structure diagram	Algorithmic Thinking Generalisation Problem Solving	Assessment on a structure diagram created and identification of input/process/output
Spring A	Algorithms (GCSE 2.1) – Designing, creating and refining algorithms. Create, interpret, correct, complete, and refine algorithms using: ○ Pseudocode ○ Flowcharts ○ Reference language/high-level programming language	Abstraction Decomposition Algorithmic thinking Evaluate Coding	Creating and or working with pseudocode and flowchart on a given scenario and implementing this into code form
Spring B	Algorithms (GCSE 2.1) – Designing, creating and refining algorithms. Identify common errors Trace tables	Problem solving Evaluate	Being able to find common errors and solve problems. Create a trace table
Summer A	Algorithms (GCSE 2.1) – Searching and sorting algorithms	Algorithmic thinking Decomposition	Knowing which search and sort should be used



	<p>Standard searching algorithms:</p> <ul style="list-style-type: none"> ○ Binary search ○ Linear search <p>□ Standard sorting algorithms:</p> <ul style="list-style-type: none"> ○ Bubble sort ○ Merge sort ○ Insertion sort 	<p>Problem Solving</p>	<p>Knowing the algorithm of the sort and search. Use a search to find an item.</p>
<p>Summer B</p>	<p>Practice programming through project work with Python</p>	<p>Coding Decomposing Algorithmic thinking Evaluate Abstraction Generalisation</p>	<p>Producing a program from a given scenario.</p>



Academic Year <u>Year; 10</u>	Content. Unit title and brief outline of content.	Skills taught in each unit.	Assessment – what knowledge and skills will be assessed and how?
Rationale	GCSE Paper 2 is taught. Pupils start GCSE modules in year 9 to understand how programming is contrasted. Exam Paper 2 is taught first as this involves the logistics of programming where pupils are lead into the start of year 11, where pupils create a programming project as part of NEA. Exam paper 2 is also the paper that is the toughest for pupils to tackle therefore it allows the content to be covered and consolidated. (OCR J277)		
Autumn A	2.2 Programming fundamentals – <input type="checkbox"/> The use of variables, constants, operators, inputs, outputs and assignments <input type="checkbox"/> The use of the three basic programming constructs used to control the flow of a program: <ul style="list-style-type: none"> ○ Sequence ○ Selection ○ Iteration (count- and condition-controlled loops) <input type="checkbox"/> The common arithmetic operators <input type="checkbox"/> The common Boolean operators AND, OR and NOT	Using programming constructs Coding Use of Boolean operators Problem Solving	End of module test Observation of class tasks to see how Boolean operators are used and constructs are used in a python program.
Autumn B	2.2 Programming fundamentals – Data Types The use of data types: <ul style="list-style-type: none"> ○ Integer ○ Real ○ Boolean ○ Character and string ○ Casting 	Coding Problem Solving Algorithmic Thinking	Test on how data types are used and changed in python program. Theory test on keywords



Spring A	2.2 Programming fundamentals – Additional programming techniques <input type="checkbox"/> The use of basic string manipulation <input type="checkbox"/> The use of basic file handling operations: <ul style="list-style-type: none"> ○ Open ○ Read ○ Write ○ Close 	Being able to file handle in python Problem solving	Observation of string manipulation and file handing is used in python program.
Spring B	2.2 Programming fundamentals – Additional programming techniques <input type="checkbox"/> The use of records to store data <input type="checkbox"/> The use of SQL to search for data <input type="checkbox"/> The use of arrays (or equivalent) when solving problems, including both one-dimensional and two-dimensional arrays <input type="checkbox"/> How to use sub programs (functions and procedures) to produce structured code <input type="checkbox"/> Random number generation	Query skills in database Creating Lists in Python Problem Solving	Observing the use of SQL and arrays to store records. End of module test
Summer A	2.3 Producing robust programs Defensive design <input type="checkbox"/> Defensive design considerations: <ul style="list-style-type: none"> ○ Anticipating misuse ○ Authentication <input type="checkbox"/> Input validation <input type="checkbox"/> Maintainability: <ul style="list-style-type: none"> ○ Use of sub programs ○ Naming conventions ○ Indentation ○ Commenting 	Validation skills Coding	End of module test. Python program created to show skills are used.



<p>Summer B</p>	<p>2.3 Producing robust programs</p> <p>Testing</p> <ul style="list-style-type: none"> <input type="checkbox"/> The purpose of testing <input type="checkbox"/> Types of testing: <ul style="list-style-type: none"> ○ Iterative ○ Final/terminal <input type="checkbox"/> Identify syntax and logic errors <input type="checkbox"/> Selecting and using suitable test data: <ul style="list-style-type: none"> ○ Normal ○ Boundary ○ Invalid ○ Erroneous <input type="checkbox"/> Refining algorithms <p>2.4 Boolean Logic</p> <ul style="list-style-type: none"> <input type="checkbox"/> Simple logic diagrams using the operators AND, OR and NOT <input type="checkbox"/> Truth tables <input type="checkbox"/> Combining Boolean operators using AND, OR and NOT <input type="checkbox"/> Applying logical operators in truth tables to solve problems 	<p>Testing</p> <p>Use of Boolean Logic</p> <p>Searching and Sorting</p> <p>Coding</p> <p>Problem Solving</p>	<p>Testing on how errors can be solved. Use of how Boolean is used.</p> <p>End of module test.</p>
------------------------	--	--	--



Academic Year <u>Year 11</u>	Content and rationale. Unit title and brief outline of content.	Skills taught in each unit.	Assessment – how will the knowledge and skills be assessed?
Rationale	Pupils then work on their programming NEA project (20 hours). This is to allow the programming element as well as the NEA, which does not count towards the final mark to be complete. In year 11 pupils start to cover Exam Paper 2. This exam paper helps pupils to be prepared for BTEC Level 3 IT course.(OCR J276)		
Autumn A	Programming project NEA (20 Hours to complete)	Abstraction Decomposition Algorithmic thinking Evaluate Coding	Completed task – NEA Skills in Python seen used
Autumn B	1.1 System Architecture 1.2 Memory 1.3 Storage	Understanding how Hardware and Software work Evaluation Problem Solving	End of module tests using exam questions
Spring A	1.4 Wired and wireless networks 1.5 Network topologies, Protocols and Layers 1.6 1.6 System Security	Understanding how Hardware and Software work Evaluation Problem Solving	End of module tests using exam questions
Spring B	1.7 System Software 1.8 Ethical, Legal, Environmental concerns Revision for final exams	Understanding how Hardware and Software work Evaluation Problem Solving	End of module tests using exam questions
Summer term: End of KS readiness for the 6th form	Key Knowledge studied at KS4 that will be useful for the 6th form	Summary of the main core skills taught at KS4 that can be reactivated at KS5	



<p>Pupils are provided with a transition booklet which details information about the BTEC IT Level 3 course as well provide homework activates to allow pupils to research topics they will cover in KS5.</p>	<p>IT skills developed from KS4 will allow pupils to be well prepared and move forward into BTEC IT. Modules covered at KS5 will include spreadsheet/Database and how Social Media is used in Business. Students who study Computer Science at KS4 will find Unit 1 within the course easier to tackle as elements from the KS4 curriculum reappear, but with a step up of a level 3 standard.</p>	<p>Being able to problem solve is a key skill, which will be required for KS5. Being organised and planning ahead will support the coursework element of the course. If pupils have studied Business Studies at KS4 skills can be transferred into the BTEC IT course as the first coursework will focus on the use of IT with Business's.</p>	
--	---	---	--