Computing Year 11 Curriculum Map - Computer Science

Notre Dame Catholic College

YEAR 11	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Curriculum Content	**Component 02 Computational Thinking, Algorithms and Programming link to KS4 Programme of Study.** **Composite = Understand and apply the fundamental principles and concepts of abstraction, decomposition and algorithms:** This topic will focus on developing an understanding of computational thinking such as abstraction, decomposition and algorithmic thinking and apply this theory to a range of problems. Pupil's will also develop a understanding of different types of algorithms such as constructs, searching and sorting and apply this theory by creating a variety of flowcharts, pseudocode and referencing language algorithms. Pupil's will also develop an understanding of identifying common errors using trace tables in a variety of algorithms too. **Component 2.1.4** Developing algorithms using flowcharts **Component 2.1.5** Developing algorithms using pseudocode **Component 2.1.6** Interpret, correct and complete algorithms	**Component 02 Computational Thinking, Algorithms and Programming link to KS4 Programme of Study.** **Composite = Analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs:** This topic will focus on developing your understanding of programming fundamentals such as the use of variables, constants, operators, inputs, outputs and assignments in coding. Pupil's will also develop a understanding of the use of constructs such as sequence, selection and iteration to control the flow of a program. Pupil's will also develop a understanding of the use of common arithmetic operators and common Boolean operators AND, OR and NOT. Pupil's will also develop an understanding of additional programming techniques such as the use of basic string manipulation, basic file handling operations and records to store data, the use of SQL to search for data and the use of one-dimensional and two-dimensional arrays. Pupil's will also learn to use subprograms (functions and procedures) to produce structured code and create random number generation. **Component 2.2.1** Programming fundamentals **Component 2.1.2** Data types **Component 2.1.3** Additional programming techniques	**Component 02 Computational Thinking, Algorithms and Programming link to KS4 Programme of Study.** **Composite = Analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs:** This topic will focus on developing your understanding of how to produce robust programs this will include: Defensive design considerations i.e. anticipating misuse, authentication, input validation and maintainability (naming conventions, indentation and commenting). Pupil's will also develop an understanding of testing such as the purpose and types of testing, identifying syntax and logic errors, being able to select and use suitable test data and develop techniques to refine algorithms too. **Component 2.3.1** Defensive design **Component 2.3.2** Testing **Composite = Understand and apply the fundamental principles and concepts of logic:** This topic will focus on developing your understanding of Boolean logic by learning the theory of logic gates and their truth tables. Pupil's will also develop an understanding of simple logic diagrams using the operators AND, OR and NOT. You will also develop an understanding of combining Boolean operators using AND, OR and NOT logic gates and applying logical operators in truth tables to solve problems. **Component 2.4.1** Boolean logic	**Component 02 Computational Thinking, Algorithms and Programming link to KS4 Programme of Study.** **Composite = Understand and apply the fundamental principles and concepts of: programming languages and IDEs:** This topic will focus on developing your understanding of the different types of language characteristics and also the purpose of different levels of programming language too. Pupil's will also develop an understanding of the purpose of translators such as the characteristics of compilers and an interpreters. Pupil's will also develop an understanding of the integrated development environment (IDE) and the common tools and facilities available in an IDE. **Component 2.5.1** Languages **Component 2.5.2** The integrated development environment (IDE)	**Component 01 Computer Systems link to KS4 Programme of Study.** **Composite = Understand the components that make up digital systems, and how they communicate with one another and with other systems. Understand the impacts of digital technology to the individual and to wider society. Apply mathematical skills relevant to Computer Science:** Pupils will revise component 01 topics in preparation for the summer examination. **Component 1.1** Systems Architecture **Component 1.2** Memory and storage **Component 1.3** Computer networks, connections and protocols **Component 1.4** Network security **Component 1.5** System software **Component 1.6** Ethical, legal, cultural and environmental impacts of digital technology **Component 02 Computational Thinking, Algorithms and Programming link to KS4 Programme of Study.** **Composite =** Pupils will revise component 02 topics in preparation for the summer examination. **Component 2.1** Algorithms **Component 2.2** Programming fundamentals **Component 2.3** Producing robust programs **Component 2.4** Boolean Logic **Component 2.5** Programming languages and Integrated Development Environments	Qualification Completed
Prior knowledge and skills (from previous year / key stage)	Pupils will have some knowledge of algorithm develop i.e. flowcharts. Pupils will have some knowledge of constructs such as sequence, selection and iteration. Pupils will have no knowledge of computational thinking concepts such as abstraction and decomposition. Pupils will have no knowledge of search and sort algorithms or the use of trace tables.	Pupils may recognise some simple programming constructs such as sequencing, selection and iteration. Pupils will have knowledge of arithmetic operators. Pupil's will have no knowledge of Boolean operators, basic string manipulation, file handling operators, SQL code, one & two dimensional arrays and subprograms (functions and procedures).	Pupils may recognise maintainability techniques such as indentation and commenting on code. Pupils will have no knowledge of defensive design considerations such as authentication and input validation. Pupils will have no knowledge of testing (purpose and types). Pupils may recognise syntax error and logic errors.	Pupils will have some knowledge programming languages python (high level language) and machine code (binary). Pupils will have some knowledge of Integrated Development Environment (IDE's) software. Pupil's will have no knowledge of assembler language and all the facilities available on a IDE.	Pupils will have some knowledge of component 01 and component 02 topics. Pupils may have some misconceptions to address for each component.	N/A

			Pupils will have knowledge of basic logic gates and truth tables. Some pupils will have knowledge of combined Boolean operators.			
Vocabulary / Key Subject Terminology	Algorithms, Pseudo code, Flow charts, Sequence, Selection, Iteration, Loops, Files, Procedures, Functions, Python, Programming, Techniques, Problem Solve, Variable, Constant, Program and Code, Search, Binary, Linear, Sort, Bubble, Insertion, Merge, Trace Table, Errors.	Programming, Fundamentals, Variables, Constants, Operators, Inputs, Outputs Assignments, Coding, Constructs, Sequence, Selection, Iteration, Control, Flow, Program, Arithmetic, Operators, Boolean AND, OR, NOT. Additional, Techniques, Basic, String Manipulation, File Handling, Operations, Records, Store, Data, SQL, Search, One-dimensional, Two-dimensional, Arrays, Subprograms, Functions, Procedures) Produce, Structured, Random, Generation.	Defensive, Design, Considerations, Anticipating, Misuse, Authentication, Input Validation, Maintainability, Naming Conventions, Indentation Commenting, Purpose, Types, Testing, Identifying, Syntax, Logic, Errors, Select, Use, Suitable, Test, Data Develop, Techniques, Refine, Algorithms. Fundamentals, Boolean, Logic, Logic Gates, Truth Tables, Simple, Diagrams Operators, AND, OR, NOT, Combining Logical, Solve, Problems.	Types, Language, Characteristics, Purpose, Levels, Assembly, Machine Code, High Level, Python, Programming Translators, Compilers, Interpreters, Integrated Development Environment (IDE), Tools, Facilities.	Refer to Year 10 curriculum map and Year curriculum map section Autumn 1 – Spring 2 Vocabulary / Key Subject Terminology section.	N/A
Assessment 1	AO3 – End of Topic Test – Algorithms (Part 2) Flowcharts, Pseudo Code & Referencing Language	AO3 – End of Topic Test – Programming Fundamentals	AO3 – End of Topic Test – Producing Robust Programs	AO1 & AO2 – End of Topic Test – Programme Languages & IDEs	AO1, AO2 & AO3 Computer Science Mock Examinations Paper 1 Computer Systems Paper 2 Computational Thinking, Algorithms and Programming	N/A
Assessment 2	AO3 – End of Topic Test – Algorithms (Part 3) Error Checking – Trace Tables	AP1 Computer Science Assessment focusing on AUT1/AUT2 theory	AO1/AO2 & AO3 – Component 02 Pass Paper 2 Computational Thinking, Algorithms and Programming Assessment	AP2 Computer Science Mock Examinations Paper 1 Computer Systems Paper 2 Computational Thinking, Algorithms and Programming	AO1, AO2 & AO3 Computer Science Mock Examinations Paper 1 Computer Systems Paper 2 Computational Thinking, Algorithms and Programming	N/A
Extra-Curricular Offer	After school weekly sessions. Additional resources promoted – Seneca, BBC Bitesize KS4 Computer Science and Oak Academy for revision materials and information.	After school weekly sessions. Additional resources promoted – Seneca, BBC Bitesize KS4 Computer Science and Oak Academy for revision materials and information.	After school weekly sessions. Additional resources promoted – Seneca, BBC Bitesize KS4 Computer Science and Oak Academy for revision materials and information.	After school weekly sessions. Additional resources promoted – Seneca, BBC Bitesize KS4 Computer Science and Oak Academy for revision materials and information.	After school weekly sessions. Additional resources promoted – Seneca, BBC Bitesize KS4 Computer Science and Oak Academy for revision materials and information.	N/A
Time Allocation	Autumn 1, 8 weeks, 2.5 lessons per week	Autumn 2, 7 weeks, 2.5 lessons per week	Spring 1, 6 weeks, 2.5 lessons per week	Spring 1 & Spring 2, 6 weeks, 2.5 lessons per week	Summer 1, 5 weeks, 2.5 lessons per week	N/A