IT/CS Curriculum Plan

Intent: The Computing department at Notre Dame High School aims to equip students with the skills to participate in a rapidly-changing world through challenging and engaging topics. Students will develop an understanding and application in the fundamental principles of computer science by having the opportunity to write programs, design webpages and produce professional digital products.
Computing skills are a major factor in enabling children to be confident, creative and independent learners and it is our intention that children have every opportunity available to allow them to achieve this.

The national curriculum for Computing aims to ensure that all pupils:

- can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
- can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
- can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
- are responsible, competent, confident and creative users of information and communication technology.

In Computer Science we are dedicated to ensuring our students leave with the skills to fully embrace a future of rapidly advancing computer technology.

The underlying attributes at the heart of our curriculum and lessons
Lessons and units are knowledge and vocabulary rich so that pupils build on what they already know to develop powerful knowledge. Knowledge is sequenced and mapped in a coherent format so that pupils make meaningful connections. Our curriculum enables us to tailor the content of the lessons to that of the national curriculum and with a local context. Our curriculum is evidence informed through rigorous application of best practice and the science of learning. We prioritise creating a diverse curriculum by committing to diversity in teaching and teachers, and the language, texts and media we use, so all pupils feel positively represented. Creating an accessible curriculum that addresses the needs of all pupils is achieved to accessibility guidelines and requirements.

Year	What will students learn?	Rationale	How will students be assessed?
7	1. Using Computers, Safely, Effectively and Responsibly – eSafety and savvy computer use 2. Control systems with Flowol - Solving problems with flowcharts 3. Understanding computers - Discover how computers work 4. Games programming in Scratch - Design and program a live web game 5. First steps in Small Basic - Introductory programming skills	In year 7 we explore the following cornerstones of the computer science curriculum: 1. Design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems 2. 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 3. 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 4. 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 5. Understand simple Boolean Logic (for example, AND, OR and NOT) and some of its uses in circuits and programming; understand now 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) 6. Understand the hardware and software components that make up computer systems, and now they communicate with one another and with other systems 7. Understand how instructions are stored and executed within a computer system 8. Understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits	Assessment is by a range of means and includes multiple choice tests, the use of an Assessment Portfolio which includes questions on what pupils have learned, space for pupils to provide evidence of their solutions and a self-assessment levelling grid.

		9. Understand a range of ways to use technology safety, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns	
8	1. Computer crime and cyber security - Hacking, data protecting and the law 2. Database development - Build and interrogate your own software system 3. Introduction to Python - Learn to program with confidence 4. Spreadsheet modelling - Develop and test a new business model 5. Modelling in Small Basic - Model sales for a lemonade stand	In year 8 we develop knowledge, skills and understanding of these cornerstones of the computer science curriculum: 1. Design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems 2. 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 3. 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 4. Understand the hardware and software components that make up computer systems, and now they communicate with one another and with other systems 5. Understand a range of ways to use technology safety, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns	Assessment is by a range of means and includes multiple choice tests, the use of an Assessment Portfolio which includes questions on what pupils have learned, space for pupils to provide evidence of their solutions and a self-assessment levelling grid.
9	1. Graphics - Discover bitmap and vector graphics 2. Creating a video - Script, film and edit an advertisement 3. HTML and website Development - Design and code a modern website	In year 9 we further develop and consolidate knowledge, skills and understanding of these cornerstones of the computer science curriculum: 1. 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	Assessment is by a range of means and includes multiple choice tests, the use of an Assessment Portfolio which includes questions on what pupils have learned, space for pupils to provide evidence of their solutions and a self-assessment levelling grid.

	4. Python Turtle – Improve confidence in programming, use graphics in a High-Level Language 5. Networks - Explore how data travels the world	2. 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 3. Understand the hardware and software components that make up computer systems, and now they communicate with one another and with other systems 4. Understand how data of various types (including text, sounds and pictures) can be represented and manipulated digitally, in the form of binary digits We explore some further cornerstones to prepare students for an information technology/computer science GCSE option by the creation of digital artefacts to meet these objectives: • 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 • Create, re-use, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability	
10	1. Python: Next steps - Advanced programming in Python 2. J277 Unit 6: Algorithms - Specification J277 from 2020 - Paper 2 3. J277 Unit 7: Programming - Specification J277 from 2020 - Paper 2 4. J277 Unit 8: Logic and languages - Specification J277 from 2020 - Paper 2 5. 20 hours practical programming 6. Exam Practice	Our GCSE (9–1) in Computer Science will encourage students to: 1. Understand and apply the fundamental principles and concepts of Computer Science, including abstraction, decomposition, logic, algorithms, and data representation 2. Analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs 3. Think creatively, innovatively, analytically, logically and critically 4. Understand the components that make up digital systems, and how they communicate with one another and with other systems 5. Understand the impacts of digital technology to the individual and to wider society 6. Apply mathematical skills relevant to Computer Science.	A low stake quiz is provided via Kahoot for each lesson in each of the J277 Units – this is used to look for a correct any common misconceptions A written assessment test is provided at the end of each J277 Unit with questions to assess understanding that unit. These are offset in order to gain recall and recovery skills

	7. J277 Unit 1: Systems architecture - Specification J277 from 2020 - Paper 1		Mock exam assessments are carried out in accordance with school directives Paper 1 is assessed by a written paper of 1 hour and 30 minutes duration. It is worth 50% of total GCSE, 80 marks This is a non-calculator paper. All questions are mandatory. Paper 2 is assessed by a written paper of 1 hour and 30 minutes duration. It is worth 50% of total GCSE, 80 marks This is a non-calculator paper. This paper has two sections: Section A and Section B. Students must answer both sections. In Section B, questions assessing students' ability to write or refine algorithms must be answered using either the OCR Exam Reference Language or the high-level programming language they are familiar with.
11	8. J277 Unit 2: Data representation - Specification J277 from 2020 - Paper 1 9. J277 Unit 3: Networks, connections and protocols - Specification J277 - Paper 1 10. J277 Unit 4: Network security and systems software - Specification J277 - Paper 1 11. J277 Unit 5: Impacts of digital technology - Specification J277 from 2020 - Paper 1 12. Exam preparation and revision		
12	1. Unit 10: Computational thinking - Component 02 2. Unit 11: Programming techniques - Component 02 3. Unit 12: Algorithms - Component 02 4. Practical Programming Project – Component 03	Our A Level in Computer Science is a practical subject where students can apply the academic principles learned in the classroom to real-world systems. It's an intensely creative subject that combines invention and excitement, and can look at the natural world through a digital prism. The aims of this qualification are to enable learners to develop: 1. An understanding of and ability to apply the fundamental principles and concepts of computer science including;	A written assessment test is provided at the end of each Unit with questions to assess understanding that unit. These are offset in order to gain recall and recovery skills

	5. OOP using LUA in the Defold environment 6. Unit 1: Components of a computer and their uses - Component 01 7. Unit 2: Systems software and applications generation - Component 01 8. Unit 3: Software development - Component 01 9. Unit 4: Exchanging data - Component 01	abstraction, decomposition, logic, algorithms and data representation 2. The ability to analyse problems in computational terms through practical experience of solving such problems including writing programs to do so 3. The capacity for thinking creatively, innovatively, analytically, logically and critically 4. The capacity to see relationships between different aspects of computer science 5. Mathematical skills 6. The ability to articulate the individual (moral), social (ethical), legal and cultural opportunities and risks of digital technology.	Regular homework is offset with the delivery to aid recall and revision Mock exam assessments are carried out in accordance with school directives Computer systems (01) is assessed by a written paper of 2 hours and 30 minutes duration. It is worth 40% of the total A Level, 140 marks. This is a non-calculator paper. All questions are mandatory. Algorithms and programming (02) are assessed by a written paper of 2 hours and 30 minutes duration. It is worth 40% of the total A Level, 140 marks. This is a non-calculator paper. All questions are mandatory. Programming Project (03) is assessed by a non-examined assessment. It is worth 20% of the total A Level, 70 marks.
13	10. Unit 5: Networks and web technologies - Component 01 11. Unit 6: Data types - Component 01 12. Unit 7: Data structures - Component 01 13. Unit 8: Boolean algebra - Component 01 14. Unit 9: Legal, moral, ethical and cultural issues - Component 01 15. Exam preparation and revision		