Computing 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.

We have split computing into two main strands:

1. Computer Science – which focuses on core computing principles, algorithms and programming.
2. Information Technology – Which focuses on the knowledge of how IT systems work and the safe and efficient use of systems and software.

The strands recognise that whilst some students may look to careers developing IT systems and software others will need to be safe and efficient users of the IT systems and software.

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 planned and sequenced to build on knowledge and concepts.
- Key terminology is reinforced throughout teaching
- Students experience a wide range of different programming languages and software, covering a range of computing skills
- Lessons support further study at KS4 of either Computer Science or iMedia
- Lessons support use of all school IT systems to support learning across all subjects

Year	What will students learn?	Rationale	How will students be assessed?	Disciplinary knowledge Skills and links to the real world
	KS3 Computing			
7	1. Using Computers, Safely, Effectively and Responsibly – eSafety and effective 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 text based programming skills	The first unit of work is to get students used to our school systems, standard ways of working and being safe and responsible users of IT Control systems shows students how to start breaking systems/processes down and produce flow charts considering inputs/processes/output which is required for most computational thinking. Understanding that digital data is essentially binary (1/0) is fundamental to understanding how computers process and store data. Understanding what hardware and software are required in computer systems. Starting to program using scratch block code to introducing basic programming constructs before moving on to a more text-based environment (Small Basic) where students learn more about syntax and structure.	Students will complete computer based assessments covering all of the material taught. Unit 1 will have an end of unit assessment in the second half term. There will be a formal assessment1 in at the beginning of the second term which will assess unit 1 and 2. In term 3 there will be a second formal assessment assessing aspects of all units taught in Y7.	• Understand principles of protecting/ saving/storing • Staying safe online • Utilise logic to help solve a problem • Represent systems in a diagrammatic way • Formulate algorithms to solve problems • How computers work • Data representation (Binary) • Basic programming constructs: -Sequence -Selection -Iteration
8	1. Introduction to Python - Learn to program with confidence 2. Graphics & Animation – Understanding image theory and how to	Intro to python goes through programming constructs sequence, selection & iteration, as well as some of the important functions in python._ closely linked to GCSE computer	Unit 1 will have an end of unit assessment in the second half term. There will be a formal assessment1 in at the	• Recognising security threats and how to address them • Know how to secure data • Legislation • Health & Safety

	effectively edit using photoshop. Understand the basic types and principles of animation. Be able to create own animations using pivot animator. 3. HTML website development - Design and code and publish to the web 4. Mobile App development using java script/blocks 5. Data Handling – Intro to database & spreadsheet functions 6. Physical computing .	science and taught before options choices made. Graphics an animation strongly supports the iMedia course and is timed to be just before options choices made. Helps students understand graphic design, image file formats and also how to edit images. Animation (moving images) introduces computer animation and tweening. Programming is revisited using HTML web development. Students learn the difference between the Internet and the world-wide-web, how HTML and CSS are used to create webpages, and create their own website on a chosen theme. Data handling helping students understand how data can be organized, stored & searched effectively, important for any system. User interface design and coding is expanded by mobile app development utilizing app developer and block Java script coding. Physical computing utilized the raspberry Pi micro computers to develop input, process & output in a physical environment.	beginning of the second term which will assess unit 1 and 2. In term 3 there will be a second formal assessment assessing aspects of all units taught in Y8.	• Understanding Job roles in cybersecurity. • Coding – Algorithms, syntax, errors. • 3 constructs: -Sequence -Selection -Iteration • Data types • Table design • Form design • Data manipulation • Logic operators and formula design

	GCSE Computer Science			
9	J277 2.2.1 Programming fundaments J277 2.2.3 Additional programming techniques J277 2.3.1 Defensive Design J277 2.3.2 Testing J277 2.5.1 Languages J277 2.5.2 IDEs J277 2.1.2 Designing, creating and refining algorithms J277 2.4.1 Boolean logic	Traditionally, programming and problem solving questions are found to be the most challenging for many students, and yet also the one aspect of computer science that most attracts and interests them. Therefore, the focus of this first year of the course is based on developing programming knowledge and skills (Paper 2 J277/02), based on the OCR Exam Reference Language pseudocode. (Many schools use Python, however the syntax of Python is quite different to that used by OCR in exam questions, often leading to an unnecessary level of cognitive load.) At NDHS we use an adaptive browser-based OCR ERL interpreter, so that students become familiar with the syntax of exam questions from day one. Students gain experience in programming, learning the basics programming fundamentals and progressing to additional programming techniques, with gradual exposure to	Summative formal assessment will be at the end of each term. Weekly homework is set and marked live, initially focused on keyword definitions and simple programming problems, and later based on past paper exam questions. Informal formative assessment takes place every lesson, in the form of bell tasks, live marking of weekly homework, and fun whole-class quizzes at suitable points during lessons. Students will also be tasked with completing extended programming projects to build knowledge and confidence.	OCR's GCSE (9–1) in Computer Science will encourage students to: • understand and apply the fundamental principles and concepts of Computer Science, including abstraction, decomposition, logic, algorithms, and data representation • analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs • think creatively, innovatively, analytically, logically and critically • 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
10	J277 1.1.1 Architecture of the CPU J277 1.1.2 CPU performance J277 1.1.3 Embedded systems J277 1.2.1 Primary storage J277 1.2.2 Secondary storage J277 1.2.3 Units	Having gained a good foundational knowledge and skills with programming, Y10 and Y11 topics mostly focus on Paper 1 (J277/01), with regular reviews of programming skills. Delivery of topics is adjusted slightly differently from the order listed in the OCR J277 specification to link related topics in a sensible way.	Summative formal assessment will be at various points throughout the year in line with the whole school assessment calendar. In Y11, assessment will be in the form of two mock exams, based on the previous summer's published external examination papers. Weekly	

	J277 1.2.4 Data storage J277 2.2.2 Data types J277 1.2.5 Compression J277 2.1.1 Computational thinking J277 2.1.3 Searching and sorting algorithms	The whole course content will be completed before the February half term in Y11, enabling ample time for review of learning, revision and exam paper practice.	homework is set and marked live, mostly based on past paper exam questions. Informal formative assessment takes place every lesson, in the form of bell tasks, live marking of weekly homework, and fun whole-class quizzes at suitable points during lessons.	
11	J277 1.3.1 Networks and topologies J277 1.3.2 Wired and wireless networks, protocols and layers J277 1.4.1 Threats to computer systems and networs J277 1.4.2 Identifying and preventing vulnerabilities J277 1.5.1 Operating systems J277 1.5.2 Utility software J277 1.6.1 Ethical, legal, cultural and environmental impact			
	Cambridge National iMedia			
9	**R093:** 2.1 How style, content and layout are linked to the purpose 2.2 Client requirements 2.3 Audience demographics and segmentation 3.2 Documents used to support ideas generation 3.3 Documents used to design and plan media products	In Year 9 students learn the specific elements of the examined unit R093 required to give them the knowledge and skills to complete the Graphics coursework unit R094. R093 elements are taught first, followed by the graphics skills required to complete the R094 Graphics coursework. This coursework will be due for handing in in the Easter of Y9	Students will be assessed via short quizzes to assess knowledge but the main assessment for Y10 will be of their R094 Graphics coursework, which will be graded by the teacher and moderated by the exam board.	• Layout, style and content conventions and how they relate to purpose of media products • how different demographics affect media products • Range of pre-production documents • Copyright and how it affects media production

	3.4.2 Intellectual property rights 3.4.3 Regulation, certification, and classification 4.2.1 Image Files **R094:** 1.1 Purpose, elements and design of visual identity 2.1 Graphic design and conventions 2.2 Properties of digital graphics 2.3 Techniques to plan 3.1 Tools and techniques of imaging editing 3.2 Preparing assets 3.3 Exporting graphic products			• Regulation and certification and the impacts on media production • Planning documents required • Properties of image files • Planning/editing and exporting graphic products • Evaluating media products
11	**R093:** 1.1 Media industry sectors and products 1.2 Job roles in the media industry 2.4 Research methods, sources and types of data 2.5 Media codes used 3.1 Work planning 3.4.4 Health and safety 4.1 Distribution platforms 4.2 Properties and formats of media files **R096 – Animation with Sound** 1.Features of animation/audio 1.2 Resources required	In Y10 students split the lesson time between theory lessons designed to give the remaning knowledge required for the exam in the summer of Y11 R093 and the knowledge and skills required to complete the coursework for R096 – Animation with sound.	Students will be assessed by short quizzes and will also have formal mock exams covering all the content of R093. Assessment for R096 – Animation with sound will be based on the coursework for hand in at Easter. This will be marked by the teacher and moderated by the exam board.	• Job roles and products in the media industry • Primary and secondary research methods • Gannt charts • Health & Safety considerations • Platforms to distribute media • Properties of a full range of media files • What an animation is • How to create and export animation • Test plans • Effective reviewing of media products

	1.3 Pre-production and planning 2.1 Creating and managing assets 2.2 Tools to create animation/sound 2.3 Exporting animation and sound 3 Testing/checking & Reviewing			
	A-Level Computer Science			
12	H446/01 1.1.1 Structure and function of the processor H446/01 1.1.2 Types of processor H446/01 1.1.3 Input, output and storage H446/01 1.2.1 Systems Software H446/01 1.2.2 Applications Generation H446/01 1.2.3 Software Development H446/01 1.2.4 Types of Programming Language H446/01 1.3.1 Compression, Encryption and Hashing H446/01 1.3.2 Databases H446/01 1.4.1 Data Types H446/01 1.4.2 Data Structures H446/01 1.4.3 Boolean Algebra	Building upon the prior knowledge and learning from GCSE computer science, at A level the course begins in Y12 following two main strands, broadly following the requirements for paper 1 (computing theory) and paper 2 (algorithms and programming). The bulk of the A level linear course is taught in Y12, including programming knowledge and skills. As with GCSE, we use the OCR ERL interpreter to learn programming skills, simply because students become familiar with the syntax used in exam paper questions. However, students are encouraged to develop their knowledge of a real programming language (eg C#, VB, Python, Lua, Javascript, PHP) in preparation for starting their coursework. Students are introduced to the project very early on, to give sufficient time to	Summative formal assessment will be at various points throughout the year in line with the whole school assessment calendar. In Y13, assessment will be in the form of two mock exams, based on the previous summer's published external examination papers. Weekly homework is set and marked live, mostly based on past paper exam questions. Informal formative assessment takes place every lesson, in the form of bell tasks, live marking of weekly homework, and fun whole-class quizzes at suitable points during lessons. A significant element of the A level is the completion of a programming project chosen	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: • An understanding and ability to apply the fundamental principles and concepts of computer science, including: abstraction,decomposition, logic, algorithms and data representation • The ability to analyse problems in computational terms through practical experience of solving such problems, including writing programs to do so

	H446/02 2.1 Elements of computational thinking H446/02 2.2.1 Programming techniques H446/02 2.2.2 Computational methods H446/02 2.3.1 Algorithms H446/03 3.1. Analysis of the problem H446/03 3.2 Design of the solution	decide on a suitably complex (but achievable) project proposal. By the end of Y12 it is expected that students would have completed the analysis and design stages of their programming projects; some may then choose to begin coding (implementation stage) during the summer holiday. Regardless, some lesson time in Y13 is given over to project development; this is important, as it enables the teacher to see the student working on their own project (rather than relying on A.I.) Project completion deadline is usually shortly after the Easter holiday. Submission of marking to OCR must be completed by 15 May.	by the student. This is marked by the teacher then externally moderated by OCR. Time is given throughout Y12/13 to work on this with appropriate guidance, although the majority of the project should be completed during the students' own study time.	• The capacity to think creatively, innovatively, analytically, logically and critically • The capacity to see relationships between different aspects of computer science • Mathematical skills.
13	H446/03 3.3 Developing the solution H446/03 3.4 Evaluation H446/01 1.3.3 Networks H446/01 1.3.4 Web Technologies H446/01 1.5 Legal, moral, cultural and ethical issues			