

KS4 Curriculum Overview: Computer Science

Rationale:

The curriculum intent for Y10 is that students will build upon knowledge and skills they have learnt during year 9 and KS3, all pupils will enhance their computational thinking skills. The year 10 content will develop problem solving abilities which will prove useful in all subjects including Computer Science. Students are given an opportunity to complete an programming project in Year 10 that combine a lot of programming skills from year 9. Although they complete in the Autumn term there is scope for this to run to the next term

Computer science year 10 curriculum is about formulating, tackling and solving problems in a unique way. Problems are to be analysed and solutions created which are explained unambiguously in order for a computer to process.

Areas that will be covered during year 10 will enable pupils to understand threats to computers and how to prevent vulnerabilities. The role of translators and facilities. Boolean logic and logic gates to enable understanding of machine code and how computers function. System software, operating systems and Utility software the purpose of each and how they enable safe computer use. To gain a deeper understanding of programming using IDE and the service they provide. To understand the ethical legal and environmental impact computers are having on society in the present and in the future.

Term / Length of Unit	Outline	Assessment	Home Learning	Resources	Knowledge/Skills End Points	Vocabulary/ reading strategy
Y10 Autumn 1	Students finish off the 2.2 programming fundamentals and then start the 20 hour project of the NEA	Students have NEA project which is marked in sections	Homework tasks to be set in accordance to the school marking policy.	All resources/ PowerPoints/ worksheets/ homework tasks which are relevant to topic being delivered to be based on: CSUK Resources Craig and Dave resources PG Online Resources Teach ICT Resources Zig Zag Resources Course Text book and cgp revision Guides	The use of basic string manipulation The use of basic file handling operations: o Open o Read o Write o Close	Arrays Structured Query Language (SQL) Procedures Selection Iteration Guided code commenting Code challenges Debugging and problem-solving Problem-solving exercises Pre teaching embedded in resources of keywords.
Y10 Autumn 2					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 All pupils will attempt to produce a solution for Music quiz extended programming challenge	

	<p>Unit 1.4 Threats to computer systems Networks</p> <p>2.3 Producing Robust Programs</p> <p>Extended Programming challenge</p> <p>It is a theoretical unit covering network threats.</p> <p>1.4.1 The unit begins by explaining the various forms of attack:</p> <ul style="list-style-type: none"> • Malware • Social engineering, e.g. phishing, people as the 'weak point' • Brute-force attacks • Denial of service attacks • Data interception and theft • The concept of SQL injection <p>1.4.2 Identifying and preventing vulnerabilities. This unit begins by explaining common prevention methods such as:</p> <ul style="list-style-type: none"> • Penetration testing • Anti-malware software • Firewalls • User access levels • Passwords • Encryption • Physical security 	<p>End of assessment for each unit</p> <p>AFL questioning throughout topic.</p> <p>Assessment via Exam style questions.</p>	<p>3 home learning tasks will be set based on Threats to computer system networks.</p> <p>Identifying and preventing vulnerabilities</p> <p>Homework tasks will be based on reinforcing content delivered in current and prior lessons to interleave and embed pupil knowledge.</p>	<p>PG online resources power point and worksheets based on:</p> <p>-unit 1.4</p> <p>- Threats to computer system networks.</p> <p>-Identifying and preventing vulnerabilities.</p> <p>- Producing Robust Programs</p> <p>-Test plans</p> <p>-Logic/syntax errors</p>	<p>1.4 All students should be able to:</p> <ul style="list-style-type: none"> • Explain threats posed to devices/systems. • Demonstrate knowledge/principles of each form of attack including: <ul style="list-style-type: none"> - How the attack is used - The purpose of the attack • Demonstrate understanding of how to limit the threats posed in 1.4.1 • Understanding of methods to remove vulnerabilities • Knowledge/principles of each prevention method: <ul style="list-style-type: none"> -What each prevention method may limit/prevent -How it limits the attack <p>2.3 Producing Robust Programs</p> <ul style="list-style-type: none"> • Understanding of the issues a programmer should consider to ensure that a program caters for all likely input values • Understanding of how to deal with invalid data in a program 	<p>Adware</p> <p>Bot</p> <p>Ransomware</p> <p>Rootkit</p> <p>Spyware</p> <p>Pre teaching embedded in resources of keywords</p> <p>Classroom discussions</p> <p>Read articles on real life situation on technology security breach.</p>
--	---	---	---	--	--	--

	<p>2.3 Producing Robust Programs</p> <p>2.3.1 Defensive design. Defensive design considerations:</p> <ul style="list-style-type: none"> -Anticipating misuse -Authentication • Input validation • Maintainability: <ul style="list-style-type: none"> - Use of sub programs - Naming conventions - Indentation - Commenting <p>2.3.2 Testing</p> <ul style="list-style-type: none"> • The purpose of testing • Types of testing: <ul style="list-style-type: none"> -Iterative -Final/terminal <ul style="list-style-type: none"> • Identify syntax and logic errors • Selecting and using suitable test data: - Normal -Boundary - Invalid - Erroneous <ul style="list-style-type: none"> • Refining algorithms 				<ul style="list-style-type: none"> • Practical experience of designing input validation and simple authentication e.g. password • Understand why commenting 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 <ul style="list-style-type: none"> • 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 • Boundary test data as data of the correct type • Invalid test data as data of the correct type but outside accepted 	
--	---	--	--	--	---	--

					<ul style="list-style-type: none"> Erroneous test data as data of the incorrect type which should be rejected by a computer system Ability to identify suitable test data for a given scenario and complete test plan 	
Y10 Spring	<p>Unit 1.5 System Software 2.4 Boolean Logic</p> <p>It is a theoretical unit covering the purpose and functionality of Operating systems and Utility software.</p> <p>1.5.1 Operating Systems The unit explains the purpose and functionality of operating systems:</p> <ul style="list-style-type: none"> User interface Memory management and multitasking Peripheral management and drivers User management File management. <p>1.5.2 Utility Software The purpose and functionality of utility software:</p> <ul style="list-style-type: none"> Encryption software 	<p>End of assessment for each unit AFL questioning throughout topic. Assessment via Exam style questions.</p>	<p>Homework tasks to be set in accordance to the school marking policy.</p> <p>3 home learning tasks will be set based on System Software Boolean logic and logic languages</p> <p>Homework tasks will be based on reinforcing content delivered in current and prior lessons to interleave and embed pupil knowledge.</p>	<p>All resources/ PowerPoints/ worksheets/ homework tasks which are relevant to topic being delivered to be based on:</p> <p>CSUK Resources Craig and Dave resources PG Online Resources Teach ICT Resources Zig Zag Resources Course Text book and cgp revision Guides. PG online resources power point and worksheets based on: Unit 1.5 -Operating system software -Utility Software. Unit 2.4 PG online resources power point and worksheets based on: -logic diagrams - truth tables</p>	<p>1.5 All Pupils should be able to :</p> <ul style="list-style-type: none"> list the functions of an operating system: user interface, memory management, multi-tasking, peripheral management, user and file management explain briefly what is meant by memory management and multi-tasking describe briefly the purpose of encryption, defragmentation and data compression software <p>Explain what each function of an operating system does</p> <ul style="list-style-type: none"> Features of a user interface Memory management, e.g. the transfer of data between memory, and how this allows for multitasking. <p>Understand that:</p> <ul style="list-style-type: none"> Data is transferred between devices and the processor This process needs to be managed and what this entails (e.g. the use of 	<ul style="list-style-type: none"> Operating systems Utility software Boolean logic Encryption software Defragmentation Data compression <p>Problem solving exercises Pre teaching embedded in resources</p>

	<ul style="list-style-type: none"> Defragmentation Data compression <p>2.4 Boolean Logic</p> <p>Pupils will be taught Boolean logic diagrams and truth tables.</p> <p>Pupils will be taught how to create simple logic diagrams using the operators AND, OR and NOT</p> <ul style="list-style-type: none"> Truth tables Combining Boolean operators using AND, OR and NOT Applying logical operators in truth tables to solve problems 				<p>buffers when transferring data to a printer)</p> <ul style="list-style-type: none"> User management functions, e.g: Allocation of an account, Access rights, Security, etc. File management, and the key features, e.g.: Naming, Allocating to folders, Moving files, Saving, etc. <p>2.4 All Pupils should be able to :</p> <ul style="list-style-type: none"> Recognise standard symbols used to represent NOT, AND OR, NAND, NOR and XOR logic gates Draw truth tables for the above logic gates Complete a trace table to trace through a simple algorithm Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios Ability to work with more than one gate in a logic diagram 	
Y10Summer	<p>1.6 Ethical</p> <p>2.5 Programming and IDE</p> <p>1.6 Ethical</p> <p>Different computer technologies and applications and the ethical, environmental and legal considerations surrounding them are discussed.</p> <p>The purpose and the clauses of the Data Protection Act and Computer Misuse Act are explained and examples</p>		<p>Homework tasks to be set in accordance to the school marking policy.</p> <p>3 home learning tasks will be set based on Ethical, legal and cultural issues. Programming and IDE.</p>	<p>All resources/ power points/ worksheets/ homework tasks which are relevant to topic being delivered to be based on:</p> <p>CSUK Resources Craig and Dave resources PG Online Resources Teach ICT Resources Zig Zag Resources Course Text book and cgp revision Guides.</p>	<p>1.6 All Pupils should be able to :</p> <ul style="list-style-type: none"> List some ethical, legal, cultural or environmental issues in relation to a given scenario List one attribute and advantage of open source software and proprietary software Understand technology introduces ethical, legal, cultural, environmental and privacy issues 	<p>Data Protection Act</p> <p>Computer Misuse Act</p> <p>Ethical</p> <p>Legal</p> <p>Cultural</p> <p>Pre teaching using resources.</p> <p>Report writing</p> <p>Literature review</p>

	<p>of situations in which they are relevant provided.</p> <p>Pupils are taught to evaluate Impacts of digital technology on wider society including:</p> <ul style="list-style-type: none"> • Ethical issues • Legal issues • Cultural issues • Environmental issues • Privacy issues <p>Legislation relevant to Computer Science:</p> <ul style="list-style-type: none"> • The Data Protection Act 2018 • Computer Misuse Act 1990 • Copyright Designs and Patents Act 1988 • Software licences (i.e. open source and proprietary) <p>2.5 Programming and IDE</p> <p>2.5.1 Programming Languages</p> <p>Pupils will be taught the characteristics and purpose of different levels of programming language:</p> <ul style="list-style-type: none"> • High-level languages • Low-level languages • The purpose of translators 		<p>Homework tasks will be based on reinforcing content delivered in current and prior lessons to interleave and embed pupil knowledge.</p>	<p>PG online resources power point and worksheets based on:</p> <p>PG online resources power point and worksheets based on:</p> <p>Unit 1.6</p> <ul style="list-style-type: none"> -ethical and cultural issues -computers in the modern world -legislation and privacy <p>Unit 2.5</p> <ul style="list-style-type: none"> - High and low level languages -Translators and compilers -IDE 	<ul style="list-style-type: none"> • Demonstrate knowledge of a variety of examples of digital technology and how this impacts on society • An ability to discuss the impact of technology based around the issues listed • Explain the purpose of each piece of legislation and the specific actions it allows or prohibits • The need to license software and the purpose of a software licence • Features of open source (providing access to the source code and the ability to change the software) • Features of proprietary (no access to the source code, purchased commonly as off-the-shelf. • Recommend a type of licence for a given scenario including benefits and drawbacks <p>2.5 All Pupils should be able to :</p> <ul style="list-style-type: none"> • Explain the differences between high- and low-level programming languages • The need for translators • Explain the differences, benefits and drawbacks of using a compiler or an interpreter. 	
--	--	--	--	---	---	--

	<ul style="list-style-type: none"> The characteristics of a compiler and an interpreter <p>2.5.2 Integrated development Environment</p> <p>Pupils will be taught common tools and facilities available in an Integrated Development Environment (IDE):</p> <ul style="list-style-type: none"> Editors Error diagnostics Run-time environment Translators 				<ul style="list-style-type: none"> Demonstrate knowledge of the tools that an IDE provides Explain how each of the tools and facilities listed can be used to help a programmer develop a program Practical experience of using a range of these tools within at least one IDE 	
--	---	--	--	--	---	--