Curriculum Overview: **Year 11 Computer Science**

A learner in Year 11 will know: what robust programming is, translators and facilities of languages computational logic and data representation and how to apply the content through exam techniques. The will know about the knowledge and understanding gained in Component 01; This knowledge and understanding will be applied to computational thinking. Students will know about algorithms, how to design them, correct them and identify errors in algorithms. They will know about programming techniques, how to produce robust programs, computational logic, translators and facilities of computing languages and data representation. They will become familiar with computing related mathematics.	**A learner in Year 11 will be able to:** demonstrate the relevant and comprehensive knowledge and understanding of fundamental concepts in programming techniques, computational thinking, algorithms and different types of algorithms. Added to this the knowledge they have gained in this unit along with that in component 1, students will become well equipped to perform well in the two exam papers.

A: Topic/Theme 2.3 Producing robust programs	B: Topic/Theme 2.4 Computational logic	C: Topic/Theme 2.5 Translators and facilities of languages	D: Topic/Theme 2.6 Data representation	E: Topic/Theme Revision	F: Topic/Theme Revision

Term 1	1:1: Topics/Themes **2.3 Producing Robust programs**	1:2: Topics/Themes **2.4 Computational logic**	Autumn % Assessment
	Knowledge: - Defensive design considerations - Maintainability of programs - Testing, purpose and types of testing - Errors and types of errors **Skills:** - Understanding of what to take into consideration when designing and creating a computer program - Understanding of how to maintain program code and the importance of doing so - Understanding of what is testing, its purpose and the different types of testing - Understanding of the different types of errors **Formative Assessment:** Pit stop: quiz on Data Robust programs **End point**: Students should be able to explain what defence programming is, what are the steps to be taken to maintain program and evaluate the importance of doing so. They should be able to identify and explain the different types of testing and errors in programming.	**Knowledge:** - Data represented in computer systems in binary form - Simple logic diagrams using the operations AND, OR and NOT - Truth tables - Logical operators in appropriate truth tables to solve problems applying computing-related mathematics **Skills:** - Understanding of why data is represented on computers only as binary - Understanding of logic gates and how to construct trust tables - Understanding of logical operators and how they used in programming **Formative Assessment:** Pit stop: quiz on computational Logic **End point:** Students should be able to explain why computers only process and use binary number system to represent data. They should be able to identify the logic gate symbols and evaluate their use in programming; identifying when they can be used to solve a problem. They should be able to construct simple truth tables as well as combine Boolean operators to two levels. They can apply computing related mathematics using the correct symbols/operators	Knowledge coverage: - Boolean Operator - Programming Logic **Skills tested**: - A01: Demonstrate knowledge and understanding of the key concepts and principles of Computer Science. - AO2: Apply knowledge and understanding of key concepts and principles of Computer Science. - AO3: Analyse problems in computational terms: ○ to make reasoned judgements ○ to design, program, evaluate and refine solutions **Assessment style/questions:** Exam style questions, combination of short written answers requiring students to state, explain, describe analyse and compare different aspects of the content covered in these units
Term 2	2:1 Topics/Themes **2.5 Translators and facilities of languages**	2:2: Topics/Themes **2.6 Data Representation**	Spring % Assessment

	Knowledge: • Different levels of programming languages • Compilers • Interpreters • Translators • Assembler • integrated development environment **Skills:** • Understanding of the different levels programming languages, their purpose and characteristic • Understanding of the purpose of assembler, compiler and interpreters • Understanding of common tools and facilities within IDE such as editors, error diagnostic, runtime environment and translator **Formative Assessment:** Pit stop: quiz on Translators and facilities of languages **End point:** Students should be able to identify the different levels of programming languages, know thee characteristic and purposes. Be able to evaluate their suitability in different scenarios. He should be identify and explain the different features within the integrated development environment (IDE)	Knowledge: • Data measure - b bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte • Number systems – denary, binary and hexadecimal • Character set – ASCII, Unicode • How are images represented as series if pixels presented in binary • The effect of colour depth and resolution on the size of an image file • How is sound sampled and stored in digital form • Data compression methods • Data encryption methods **Skills:** • An understanding of why and how data needs to be converted into a binary format to be processed by a computer • Understanding of how data is measured on the computer • Be able to convert positive denary whole numbers (0–255) into 8 bit binary numbers and vice versa • Be able to convert positive denary whole numbers (0–255) into 2 digit hexadecimal numbers and vice versa • Understanding of how to convert from binary to hexadecimal equivalents and vice versa • Be able to explain what/why is compression needed **Formative Assessment:** Pit stop assessment **End point:** Students should be able to confidently explain why computers store and process data in binary format. They should be able convert decimal/denary → binary → hexadecimal. They should be to explain how data is compressed and analyse of what methods are effective based on a given scenario. The can explain encryption and the purpose of encrypting data.	Knowledge coverage: • How does the computer represent text, image and sound • Data compression methods • Programming languages • Features within the IDLE • Encryption **Skills tested:** • A01: Demonstrate knowledge and understanding of the key concepts and principles of Computer Science. • AO2: Apply knowledge and understanding of key concepts and principles of Computer Science. • AO3: Analyse problems in computational terms: ○ to make reasoned judgements ○ to design, program, evaluate and refine solutions **Assessment style/questions:** Exam style questions, combination of short written answers requiring students to state, explain, describe analyse and compare different aspects of the content covered in these units
Term 3	3:1: Topics/Themes **Revision- EXAM PREP**	3:2: Topics/Themes **Revision – EXAM PREP**	Summer % Assessment
			Skills tested: • A01: Demonstrate knowledge and understanding of the key concepts and principles of Computer Science. • AO2: Apply knowledge and understanding of key concepts and principles of Computer Science. • AO3: Analyse problems in computational terms: ○ to make reasoned judgements

			○ to design, program, evaluate and refine solutions