

Pupils in Computer Science in year 11 have two (1 hour) lessons each week.

Department: <i>ICT & Computing</i>							
Term	Topic/Subject	Assessment Objectives	Knowledge Acquisition	Skill building & intent	Wider reading opportunities, including numeracy & SMSC.	Assessment Task	SEND & PP
Aut 1	2.3 Producing Robust Programs	<ul style="list-style-type: none"> Know what is meant by the term “defensive design considerations” when writing programs. Understand why input validation is necessary. Know a range of validation techniques that can be used to write a robust program. Know what is meant by the term “defensive design considerations” when writing programs. Know a range of potential problems that can occur when a program is running, especially if it requires communication to servers, peripherals, data in files and arithmetic. Understand some authentication techniques a programmer may choose to use to protect their program from misuse. Know why creating easy to read code is important with large projects. Understand what programmers can do to make their 	Pupils learn key concepts of producing robust programs, including validation techniques, defensive design, refining algorithms, syntax errors, logic errors & test data.	Pupils to continue to use the Google Classroom teaching resources for this topic. Theory topics are introduced during each lesson through video based resources, supported by google slide presentations. Pupils use flip learning & cornell note taking methods to complete a series of online workbook based tasks. All resources are accessible at home, and are supported through ‘CraignDave’ Smart Revise platform of self quizzing.	<p>A combination of visual & google classroom based resources are used, and all are delivered to pupils in lesson time. Pupils can access all resources, and SmartRevise revision platform is used throughout.</p> <p>Key Questions:</p> <ul style="list-style-type: none"> What issues should a programmer consider to ensure a program caters for all likely input values? What does code maintainability mean? What are the different types of errors that can occur in a program? What are the features of good testing strategy? What makes a robust program? 	Smart Revise assessment evidence is used to track progress from start to end of topic. End of topic written test is also completed.	Additional support to be provided where required for both SEND & PP pupils to help access resources.

		<p>code more readable.</p> <ul style="list-style-type: none"> • Understand how to refine algorithms in order to make them more robust. • Know four reasons why a program should be tested. • Know what iterative testing is. • Know what final/terminal testing is. • Know what a syntax error is. • Know what a logic error is. • Understand that because a program works, it doesn't mean it works for all inputs. • Understand that suitable test data for a program needs to include: <ul style="list-style-type: none"> ○ Normal data ○ Boundary data ○ Invalid data ○ Erroneous data 					
Aut 2	Paper 2 Exam Revision unit	<ul style="list-style-type: none"> • Understand how to develop techniques to answer the J277/02 component paper. • Understand how to address the question focus areas of component 2 paper; design, write, test & refine. • Understand flow charts & data types. • Understand how to refine algorithms. • Understand how to use appropriate test data and techniques. • Understand how to write answers in 	Pupils learn key techniques of answering component 2 exam questions.	Pupils build on skills already developed, and use a combination of resources available through google classroom.	A combination of visual & google classroom based resources are used, and all are delivered to pupils in lesson time. Pupils can access all resources, and SmartRevise revision platform is used throughout.	Smart Revise assessment evidence is used to track progress from start to end of topic.	Additional support to be provided where required for both SEND & PP pupils to help access resources.

		OCR exam reference language or high-level language of choice (Python)					
Spr 1	2.4 Boolean Logic	<ul style="list-style-type: none"> Know how to make simple logic diagrams from Boolean expressions using AND, OR, NOT. Understand how to complete truth tables from one and two level logic diagrams. Understand how to create, complete or edit logic diagrams and truth tables for given scenarios. 	Pupils learn key concepts of Boolean logic, including logic diagrams, & truth tables.	Pupils to continue to use the Google Classroom teaching resources for this topic. Theory topics are introduced during each lesson through video based resources, supported by oogle slide presentations. Pupils use flip learning & cornell note taking methods to complete a series of online workbook based tasks. All resources are accessible at home, and are supported through 'CraignDave' Smart Revise platform of self quizzing.	<p>A combination of visual & google classroom based resources are used, and all are delivered to pupils in lesson time. Pupils canaccess all resources, and SmartRevise revision platform is used throughout.</p> <p>Key Questions:</p> <ul style="list-style-type: none"> What are the symbols used in logic diagrams? How do you complete a truth table? How do you create logic diagrams from truth tables? <p>A combination of tutorial video, and google slides based resources are used, together with demonstrations outlining core coding concepts.</p>	Smart Revise assessment evidence is used to track progress from start to end of topic. End of topic written test is also completed.	Additional support to be provided where required for both SEND & PP pupils to help access resources.
Spr 2	2.5 Programmin g Languages & IDEs	<ul style="list-style-type: none"> Know the characteristics of high level and low level programming languages. Understand the terms: <ul style="list-style-type: none"> Source code 	Pupils learn key concepts of programming languages & IDEs, including characteristics of high level languages, assembly code, source code, translators,	Pupils to continue to use the Google Classroom teaching resources for this topic. Theory topics are introduced during each lesson through video based resources, supported by oogle slide presentations. Pupils use flip learning & cornell note taking	A combination of tutorial video, and google slides based resources are used, together with demonstrations outlining core coding concepts.	Smart Revise assessment evidence is used to track progress from start to end of topic. End of topic written test is also completed.	Additional support to be provided where required for both SEND & PP pupils to help access resources.

		<ul style="list-style-type: none"> ○ Assembly code ○ Machine code ● Understand how to write programs in a low level language using assembly with Little Man Computer. ● Know what a translator does. ● Understand the differences between compilers and interpreters. ● Know a range of facilities provided by an integrated development environment (IDE) to assist the programmer in writing code. 	<p>difference between compilers & interpreters.</p>	<p>methods to complete a series of online workbook based tasks. All resources are accessible at home, and are supported through 'CraignDave' Smart Revise platform of self quizzing.</p>	<p>Key Questions:</p> <ul style="list-style-type: none"> ● What are the differences between high and low level languages? ● How do you write a program in assembly language? ● How does code a programmer writes become binary a computer can execute? ● Why do programmers use IDEs? 		
Sum 1	Component 1 & component 2 exam paper revision.	<ul style="list-style-type: none"> ● Understand how to develop techniques to answer the J277/02 component paper. ● Understand how to address the question focus areas of component 2 paper; design, write, test & refine. ● Understand flow charts & data types. ● Understand how to refine algorithms. ● Understand how to use appropriate test data and techniques. ● Understand how to write answers in OCR exam reference language or high-level 	<p>Pupils learn key techniques of answering component 1&2 exam questions.</p>	<p>Pupils build on skills already developed, and use a combination of resources available through google classroom. All resources are accessible at home, and are supported through 'CraignDave' Smart Revise platform of self quizzing, and advanced exam style question technique & analysis.</p>	<p>A combination of visual & google classroom based resources are used, and all are delivered to pupils in lesson time. Pupils can access all resources, and SmartRevise revision platform is used throughout.</p>	<p>Smart Revise assessment evidence throughout, combined with past paper question bank of resources.</p>	<p>Additional support to be provided where required for bothSEND & PP pupils to help access resources.</p>

		language of choice (Python)					
							9

Last updated: June 2023