A level/ BTEC – Subject- Exam board

1a) Intent: In ICT and Computing we aim to support our students to become active members of a rapidly changing digital world. We develop students' computational thinking and programming skills to enable them to have a positive approach to a real world problems. Our broad curriculum allows students to develop transferable skills in both ICT and computing, including the ability to program in various languages, and use of a wide range of hardware and software and devices. We encourage our students to challenge themselves to create high quality digital products and take pride in demonstrating their skills and creativity. Our teaching encourages students to consider their digital footprint, the impact of their digital presence and the robustness and accuracy of information found online. We encourage a culture of resilience where learning from failure is the key to success.

Student version in classroom – Our mission is to empower students with computational skills, to ignite your creativity and instil digital responsibility.

1b) Careers and further study: In Computer Science we prepare our students for careers in cyber security, robotics, data science and analysis as well as general programming whether this be via university or degree apprenticeship.

2)Implementation: What do we do in lessons?

Implementation – Pedagogical approaches including Rosenshine principles of instruction

Daily Review	New Material in Small Steps	Ask Questions	Provide Models	Guide Student Practice	Check Student Understanding	Obtain High Success Rate	Scaffolds for Difficult Tasks	Independent Practice	Weekly and Monthly Review
Daily review is an important component of instruction. It helps strengthen the connections of the material learned. Automatic recall frees working memory for problem solving and creativity.	Our working memory is small, only handling a few bits of information at once. Avoid its overload—present new material in small steps and proceed only when first steps are mastered.	The most successful teachers spend more than half the class time lecturing, demonstrating and asking questions. Questions allow the teacher to determine how well the material is learned.	Students need cognitive support to help them learn how to solve problems. Modelling, worked examples and teacher thinking out loud, help to clarify the specific steps involved.	Students need additional time to rephrase, elaborate and summarise new material in order to store it in their long-term memory. More successful teachers build in more time for this.	Less successful teachers merely ask "Are there any questions?" no questions are taken to mean no problems. False. By contrast, more successful teachers check on all students.	A success rate of around 80% has been found to be optimal, showing students are learning and also being challenged. Better teachers taught in small steps followed by practice.	Scaffolds are temporary supports to assist learning. They can include modelling, teacher thinking aloud, cue cards and checklists. Scaffolds are part of cognitive apprenticeship.	Independent practice produces 'overlearning' - a necessary process for new material to be recalled automatically. This ensures no overloading of students' working memory.	The effort involved in recalling recently -learned material embeds it in long-term memory. And the more this happens, the easier it is to connect new material to such prior knowledge.
• Every unit of work has a series of quiz questions to help students recall key knowledge. These are used in lessons and for prep work.	• Teachers define and chunk the steps for students to follow when learning new material. These steps are agreed across the department.	• Teachers use cold calling, pair share and stretch it TLAC strategies to check for mastery. Questions are pre-planned.	• The visualiser is used across the department. Teachers will 'live' model to demonstrate how to construct analytical and creative texts.	• Tasks and activities have been designed so that automaticity can be achieved. Repetition and revision is built into tasks.	• Specific mastery checks are embedded into SOLS so that teachers can check for mastery.	• We use I do, We do, You do to build students retention of key procedural knowledge and support automaticity.	• Scaffolds are pre-planned so that there is consistency across the department. Testing includes memorisation of scaffolds.	• Students repeat activities and tasks at spaced intervals to support learning of key procedural knowledge as well as knowledge.	• We map our quiz questions so that we can test core learning throughout the year. All SOLS have defined 'retention' knowledge.

KS5	Term 1	Term 2		Term 3	Term 4	Term 5	Term 6	
Year 12 applied	SLR 1 Programming basics SLR 2 Programming next steps	SLR 3 Programming paradigms SLR 4 Data Structures SLR 6 Abstraction and automation SLR7 Regular and context-free languages And some dedicated programming lessons		SLR 10 Number system and bases SLR 11 Binary SLR 12 Coding text and graphics And some dedicated programming lessons	SLR 13 Coding sound and music SLR 14 Hardware and software SLR 15 Programming languages and translators SLR 16 Logic gates and Boolean algebra SLR 17 Internal computer architecture	SLR 18 Input and output devices SLR 19 Moral, social, legal, cultural issues SLR 20 Communication SLR 21 Network and the internet SLR 27 Aspects of software development And some revision lessons	**Yr12 PPES**	Preparation for year 13 course after exams
Year 12 Technical								
Year 13 applied	SLR 3 Programming paradigms SLR 4 Data Structures SLR 5 Algorithms	SLR 2 Programming next steps	**Year 13 PPES**	SLR 17 Internal computer architecture SLR 21 Network and the internet	SLR 22 TCP IP and protocols SLR 23 Databases	SLR 24 Big data SLR 25 Functional programming paradigms		

	SLR7 Regular and context-free languages And some dedicated project lessons	SLR7 Regular and context-free languages SLR 8 Classification of algorithms SLR 9 A model of computation SLR 11 Binary SLR 12 Coding text and graphics And some dedicated project lessons		And some dedicated project lessons	And some dedicated project lessons And some dedicated project lessons	SLR 26 Writing functional programs SLR 27 Aspects of software development	
Year 13 technical							

3)Impact: Assessment and outcomes in key stage

	KS5				
SUBJECT					
	Nat. Av %	*Cohort %*	*Non-PP %*	*PP %*	*Gap %*
A-B*	43.4	50	60	0	+6.6
A-C*	62.9	83	80	100	+20.1
A-E*	95.2	100	100	100	+4.8
VA score					

Destinations