CURRIULUM SUMMARY - Computing

YEAR 9 – Computer Science GCSE AQA 8525

TERM 1	TERM 2	TERM 3
CONTENT • Computational thinking • Programming literacy • Data types • Variables and Constants • Selection • Sequencing • Iteration • Subroutines • Error trapping	**CONTENT** • Algorithms • Flowcharts • Pseudocode • Programming literacy • ROM and RAM	**CONTENT** • Boolean and Logic gates • Data representation • Data compression • Computational thinking • Algorithms • Flowcharts • Pseudocode • Programming literacy • Logic circuits • Binary • Hexadecimal
ASSESSMENTS • Online tests • Functionality checks of programmes (does it work and do what it was supposed to do). • Peer assessment	**ASSESSMENTS** • Online tests • Functionality checks of programmes (does it work and do what it was supposed to do). • Peer assessment	**ASSESSMENTS** • Online tests • Functionality checks of programmes (does it work and do what it was supposed to do). • Peer assessment
HOW PARENTS CAN SUPPORT LEARNING • Check homework is completed • Support pupils with written work • Ensure pupils have access to a computer with Python installed and web access, so online resources can be used. • Give pupils encouragement to get past the inevitable struggles and failures inherent in programming.	**HOW PARENTS CAN SUPPORT LEARNING** • Check homework is completed • Support pupils with written work • Ensure pupils have access to a computer with Python installed and web access, so online resources can be used. • Give pupils encouragement to get past the inevitable struggles and failures inherent in programming.	**HOW PARENTS CAN SUPPORT LEARNING** • Check homework is completed • Support pupils with written work • Ensure pupils have access to a computer with Python installed and web access, so online resources can be used. • Give pupils encouragement to get past the inevitable struggles and failures inherent in programming.

YEAR 10 – Computer Science GCSE AQA 8525

TERM 1	TERM 2	TERM 3
CONTENT • Computational thinking • Algorithms • Flowcharts • Pseudocode • Programming literacy • Logic circuits • ROM and RAM • Different programming languages • Binary • Hexadecimal	**CONTENT** • Data structures • Trees and Huffman coding • Understanding search and sort algorithms • Algorithm efficiency • Testing code • Computer system's architecture • Von Neumann • Networks • Embedded systems • Memory ROM & RAM • Secondary storage • Flowcharts • Pseudocode • Programming literacy	**CONTENT** • Fetch-execute cycle • Encryption • System security • Social engineering and cyber security • Ethics, the law and environment • Software and its development • Algorithms • Flowcharts • Pseudocode • Programming literacy • Databases
ASSESSMENTS • Online tests • Functionality checks of programmes (does it work and do what it was supposed to do). • Peer assessment	**ASSESSMENTS** • Online tests • Functionality checks of programmes (does it work and do what it was supposed to do). • Peer assessment	**ASSESSMENTS** • Online tests • Functionality checks of programmes (does it work and do what it was supposed to do). • Peer assessment
HOW PARENTS CAN SUPPORT LEARNING • Check homework is completed • Support pupils with written work • Ensure pupils have access to a computer with Python installed and web access, so online resources can be used. • Give pupils encouragement to get past the inevitable struggles and failures inherent in programming.	**HOW PARENTS CAN SUPPORT LEARNING** • Check homework is completed • Support pupils with written work • Ensure pupils have access to a computer with Python installed and web access, so online resources can be used. • Give pupils encouragement to get past the inevitable struggles and failures inherent in programming.	**HOW PARENTS CAN SUPPORT LEARNING** • Check homework is completed • Support pupils with written work • Ensure pupils have access to a computer with Python installed and web access, so online resources can be used. • Give pupils encouragement to get past the inevitable struggles and failures inherent in programming.

YEAR 11 – Computer Science GCSE AQA 8525

TERM 1	TERM 2	TERM 3
CONTENT • **Algorithms** • **Flowcharts** • **Pseudocode** • **Programming literacy** • **Data structures** • **Trees and Huffman coding** • **Understanding search and sort algorithms** • **Algorithm efficiency** • **Testing code** • **Computer system's architecture** • **Von Neumann** • **Networks** • **Embedded systems** • **Memory ROM & RAM** • **Secondary storage**	**CONTENT** • **Revision and preparation for final exam.** • **Consolidation of programming knowledge.** • Fetch-execute cycle • Encryption • System security • Social engineering and cyber security • Ethics, the law and environment • Software and its development • Algorithms • Flowcharts • Pseudocode • Programming literacy • Databases	**CONTENT** • **Preparation for final exam**
ASSESSMENTS • **Online tests** • **Functionality checks of programmes (does it work and do what it was supposed to do).** • **Peer assessment**	**ASSESSMENTS** • **Online tests** • **Functionality checks of programmes (does it work and do what it was supposed to do).** • **Peer assessment**	**ASSESSMENTS** • **Online tests** • **Functionality checks of programmes (does it work and do what it was supposed to do).** • **Peer assessment**
HOW PARENTS CAN SUPPORT LEARNING • **Check homework is completed** • **Support pupils with written work** • **Ensure pupils have access to a computer with Python installed and web access, so online resources can be used.** • **Give pupils encouragement to get past the inevitable struggles and failures inherent in programming.**	**HOW PARENTS CAN SUPPORT LEARNING** • **Check homework is completed** • **Support pupils with written work** • **Ensure pupils have access to a computer with Python installed and web access, so online resources can be used.** • **Give pupils encouragement to get past the inevitable struggles and failures inherent in programming.**	**HOW PARENTS CAN SUPPORT LEARNING** • **Check homework is completed** • **Support pupils with written work** • **Ensure pupils have access to a computer with Python installed and web access, so online resources can be used.** • **Give pupils encouragement to get past the inevitable struggles and failures inherent in programming.**