Years 12 & 13 Curriculum
A Level: Computer Science

Year 12	Term 1 (Autumn)		Term 2 (Spring)			Term 3 (Summer)
	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Topic	**Abstraction & Automation** **Theory of Computation** **Fundamentals of Data Representation** **Fundamentals of Computer Systems** **Fundamentals of Programming**	**Theory of Computation** **Fundamentals of Data Representation** **Fundamentals of Computer systems** **Consequences of Uses of Computing**	**Fundamentals of Communication & Networking** **Classification of Programming Languages & Translation** **Exam Code Release & Exam Preparation**	**Revision** **Mock Examinations** **Exam Code Preparation and Java Practice**		**Fundamentals of Programming**
Key Concepts	• Abstraction, Information hiding, Procedural abstraction, Functional abstraction, Data abstraction, Problem abstraction/reduction, Decomposition, Composition, and Automation • Finite state machines • Number types • Number Bases • Units of information • External Hardware devices including secondary storage devices	• Finite state machines • Number types • Number Bases • Units of information • Information Coding Systems • Representing images, sound and other data • Data Compression and Encryption • Logic gates • Boolean Algebra • Types of program translator • Internal hardware components of a computer • The stored program concept • Structure and role of the processor and its components • The Fetch-Execute cycle and the role of registers within it • The processor instruction set • Addressing modes • Machine-code/assembly language operations • Factors affecting processor performance • Individual (moral), social (ethical), legal and cultural issues and opportunities	• Communication methods • The processor instruction set • Addressing modes • Machine-code/assembly language operations	• Working through past exam code and completing practice tests.		• Object orientated programming - Class - Object - Instantiation - Encapsulation - Inheritance - Aggregation - Composition - Polymorphism - Overriding. • Java FX • Abstract Data Structures • Introduction to the NEA

Years 12 & 13 Curriculum *(continued)*
A Level: Computer Science

Year 13	Term 1 (Autumn)		Term 2 (Spring)		Term 3 (Summer)	
	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Topic	**Fundamentals of Data Structures** **Fundamentals of Algorithms** **NEA and Exam Code Preparation**	**Fundamentals of Data Representation** **Theory of Computation** **Searching & Sorting Algorithms** **Fundamentals of Algorithms** **NEA & Exam Code Preparation**	**Theory of Computation** **Fundamentals of Data Representation** **Fundamentals of Computer Organisation & Architecture** **Logic Gates** **Fundamentals of Databases** **The Internet** **NEA & Exam Code Preparation**	**Fundamentals of Communication & Networking** **Big Data** **NEA & Exam Code Preparation**	**Fundamentals of Functional Programming** **Exam Code Preparation**	
Key Concepts	• Data structures and abstract data types • Trees – Binary Trees • Hash Tables • Vectors • Graph Traversal	• Real Numbers - Numbers with a fractional part • Reverse Polish – infix transformations • Finite state machines (FSMs) with and without output • Regular expressions and maths for regular expressions • Backus-Naur Form (BNF)/syntax diagrams • Linear and Binary Search • Bubble and Merger Sort • Comparing algorithms • Maths for understanding Big-0 notation • Order of complexity • Limits of computation • Classification of algorithmic problems • Computable and non-computable problems • Halting problem	• Turing Machines • Vector graphics • Vector graphics versus bitmapped graphics • Interrupts • Logic gates • Conceptual data models and entity relationship modelling • Relational databases • Database design and normalisation techniques • Structured Query Language (SQL) • Client server databases • The Internet and how it works • Internet security • TCP/IP • Standard application layer protocols	• IP address structure • Subnet masking • IP standards • Public and private IP addresses • Public and private IP addresses • Network Address Translation (NAT) • Port forwarding • Client server model • Thin- versus thick-client computing • Big Data – volume, velocity and variety • Be familiar with the: - Fact-based model for representing data - Graph schema for capturing the structure of the dataset - Nodes, edges and properties in graph schema	• Function type • First-class object • Function application • Partial function application • Composition of functions • Writing functional programs • Functional language programs • List processing	

NEA = Non-examined assessment

Years 12 & 13 Curriculum
A Level: Computer Science

Year 12	Term 1 (Autumn)		Term 2 (Spring)			Term 3 (Summer)
	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Topic	**Abstraction & Automation** **Theory of Computation** **Fundamentals of Data Representation** **Fundamentals of Computer Systems** **Fundamentals of Programming**	**Theory of Computation** **Fundamentals of Data Representation** **Fundamentals of Computer systems** **Consequences of Uses of Computing**	**Fundamentals of Communication & Networking** **Classification of Programming Languages & Translation** **Exam Code Release & Exam Preparation**	**Revision** **Mock Examinations** **Exam Code Preparation and Java Practice**		**Fundamentals of Programming**
Key Concepts	• Abstraction, Information hiding, Procedural abstraction, Functional abstraction, Data abstraction, Problem abstraction/reduction, Decomposition, Composition, and Automation • Finite state machines • Number types • Number Bases • Units of information • External Hardware devices including secondary storage devices	• Finite state machines • Number types • Number Bases • Units of information • Information Coding Systems • Representing images, sound and other data • Data Compression and Encryption • Logic gates • Boolean Algebra • Types of program translator • Internal hardware components of a computer • The stored program concept • Structure and role of the processor and its components • The Fetch-Execute cycle and the role of registers within it • The processor instruction set • Addressing modes • Machine-code/assembly language operations • Factors affecting processor performance • Individual (moral), social (ethical), legal and cultural issues and opportunities	• Communication methods • The processor instruction set • Addressing modes • Machine-code/assembly language operations	• Working through past exam code and completing practice tests.		• Object orientated programming - Class - Object - Instantiation - Encapsulation - Inheritance - Aggregation - Composition - Polymorphism - Overriding. • Java FX • Abstract Data Structures • Introduction to the NEA

Years 12 & 13 Curriculum *(continued)*
A Level: Computer Science

Year 13	Term 1 (Autumn)		Term 2 (Spring)		Term 3 (Summer)	
	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Topic	**Fundamentals of Data Structures** **Fundamentals of Algorithms** **NEA and Exam Code Preparation**	**Fundamentals of Data Representation** **Theory of Computation** **Searching & Sorting Algorithms** **Fundamentals of Algorithms** **NEA & Exam Code Preparation**	**Theory of Computation** **Fundamentals of Data Representation** **Fundamentals of Computer Organisation & Architecture** **Logic Gates** **Fundamentals of Databases** **The Internet** **NEA & Exam Code Preparation**	**Fundamentals of Communication & Networking** **Big Data** **NEA & Exam Code Preparation**	**Fundamentals of Functional Programming** **Exam Code Preparation**	
Key Concepts	• Data structures and abstract data types • Trees – Binary Trees • Hash Tables • Vectors • Graph Traversal	• Real Numbers - Numbers with a fractional part • Reverse Polish – infix transformations • Finite state machines (FSMs) with and without output • Regular expressions and maths for regular expressions • Backus-Naur Form (BNF)/syntax diagrams • Linear and Binary Search • Bubble and Merger Sort • Comparing algorithms • Maths for understanding Big-0 notation • Order of complexity • Limits of computation • Classification of algorithmic problems • Computable and non-computable problems • Halting problem	• Turing Machines • Vector graphics • Vector graphics versus bitmapped graphics • Interrupts • Logic gates • Conceptual data models and entity relationship modelling • Relational databases • Database design and normalisation techniques • Structured Query Language (SQL) • Client server databases • The Internet and how it works • Internet security • TCP/IP • Standard application layer protocols	• IP address structure • Subnet masking • IP standards • Public and private IP addresses • Public and private IP addresses • Network Address Translation (NAT) • Port forwarding • Client server model • Thin- versus thick-client computing • Big Data – volume, velocity and variety • Be familiar with the: - Fact-based model for representing data - Graph schema for capturing the structure of the dataset - Nodes, edges and properties in graph schema	• Function type • First-class object • Function application • Partial function application • Composition of functions • Writing functional programs • Functional language programs • List processing	

NEA = Non-examined assessment