Cardinal Newman Catholic School
Department of computer science & ICT

INTENT: Curriculum Overview: **Year 10 Computer Science**

A learner in Year 10 will know: *understand the components that make up computer systems and how they communicate with one another. They will learn about the different kinds of memory used within a computer system, software and the different types of software, the roll of the OS, computer networks and the security issue associated with the use of computer networks. They will also learn about the computational thinking processes that is essential to create programs for a computer.*			**A learner in Year 10 will be able to**: *the understand and apply the fundamental principles and concepts of Computer Science, including algorithms, and data representation, analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs and understand the impacts of digital technology to the individual and to wider society.*		
A: Topic/Theme **1.1Computer System Architecture**	B: Topic/Theme **1.2Memory**	C: Topic/Theme **1.3 System Software**	D: Topic/Theme **1.4 Computer Networks** **1.6 system security**	**E: Topic/Theme** **Algorithms**	**D: Topic/Theme** **Programing Techniques**
2.1 &2.2 Computational thinking / programing			2.1 &2.2 Computational thinking / programing		

Term 1 **15 Wks**	**Topics/Themes** 1.1 Systems architecture 2.1 &2.2 Computational thinking / programing	**Topics/Themes** 1.2 Memory 1.3 System Software 2.1 &2.2 Computational thinking / programing	Autumn % Assessment *(ensure differentiated assessment is planned too)*
	Knowledge: This unit will introduce students to the components of the motherboard, identifying heir features and functions within the computer system. **Skills:** ▪ Examine the major components of the PC architecture; the CPU, Cache and Registers, their features/characteristics and functions within the computer system. ▪ Understand the purpose of embedded systems and be able to identify them ▪ Understand how design an algorithm for chosen problem ▪ Understand how to interpret algorithms ▪ Understand sorting algorithm (insertion, merge &bubble sort) and searching algorithm (linear and binary) **Formative Assessment:** Pit stops: Components of the motherboard **End point:** ▪ Students can identify the features of CPU and their functions ▪ They can explain the role and impact of Cache on the performance of the computer ▪ They can explain and identify embedded computer systems	**Knowledge:** This Unit looks at types of memory, their features, characteristic and their role within the computer system; virtual memory and its impact the performance of a computer system. Unit also covers data capacity and calculation of data capacity requirements and common types of storage technology used within the industry. **Skills:** ▪ Understand the functions of RAM, ROM and virtual memory ▪ Understand the impact on performance of RAM ad virtual memory ▪ Identify common types of storage technologies and storage media ▪ Understand how design an algorithm for chosen problem ▪ Understand how to interpret algorithms **Formative Assessment:** it stop assessment **End point:** ▪ They can explain the difference between RAM & ROM ▪ Explain the function of cache and its impact on the speed of a computer ▪ Understand the impact of virtual memory on the performance of a computer ▪ Identify suitable storage media for different applications, their pros and cons	Knowledge coverage: **Memory and System Software** **Systems architecture** Skills tested: ▪ Components of computer architecture, their features and functions ▪ Types of memory, their features and functions ▪ Types of storage and examples of different storage media and technology **Assessment style/questions:** Exam style questions, combination of short written answers requiring students to state, explain, describe analyse and compare

Cardinal Newman Catholic School
Department of computer science & ICT

Term 2 **12 Wks**	2:1 Topics/Themes **1.4 Computer Networks** 2.1 &2.2 Computational thinking / programing	2:2: Topics/Themes **1.6 system security** 2.1 &2.2 Computational thinking / programing	Spring % Assessment *(ensure differentiated assessment is planned too)*
	Knowledge: Unit will introduce students to the concept of networking computers, network types and topologies, system security and system software. **Skills:** ▪ Understand what computer networks are, ▪ identify their pros and cons, ▪ Identify types of networks and the different network topologies and their features ▪ Understand how design an algorithm for chosen problem ▪ Understand how to interpret algorithms **Formative Assessment:** Pit stop: quiz on network topologies End point: ▪ Know why companies setup network and identify the pros and cons ▪ Identify and explain the different network topologies ▪ Identify the factors that affect the performance of networks ▪ Identify the hardware needed to connect stand-alone computers into a Local Area Network	Knowledge: Unit covers the purpose and functionality of systems software. Data representation and different number system in computing **Skills:** Be able to identify and place software in these two groups based on the functions they perform within the computer system ▪ Understand how design an algorithm for chosen problem ▪ Understand how to interpret algorithms **Formative Assessment**: Pit stop: Questions on system software **End point:** ▪ Be able to explain what is system software ▪ Identify and explain the difference between the 2 categories ▪ be able to convert data from denary →binary → hexadecimal and vice versa	Knowledge coverage: Computer networks System software **Skills tested:** ▪ Understanding of what system software is and the 2 categories of software – operating and utility software. ▪ Understanding of what computer networks are, identify their pros and cons, types of networks and the different network topologies and their features **Assessment style/questions:** exam style questions, combination of short written answers requiring students to state, explain, describe analyse and compare
Term 3 12 wks	3:1: Topics/Themes **Data Representation and Algorithms** 2.1 &2.2 Computational thinking / programing	3:2: Topics/Themes **1.7 System Software** 2.1 &2.2 Computational thinking / programing	Summer % Assessment *(ensure differentiated assessment is planned too)*
	Knowledge: Introducing student to the concept of problem solving, they will look the role of algorithms when design algorithms and how algorithm can be represented using flow charts and pseudocode. They will look at sorting and searching algorithms **Skills:** Understand how design an algorithm for chosen problem Understand how to interpret algorithms Understand sorting algorithm (insertion, merge &bubble sort) and searching algorithm (linear and binary) Data compression Data encryption **Formative Assessment:** Pit stop: quiz on computational thinking	Knowledge: Students will be introduced to programming, learning about programming techniques and how to produce robust programs,. Students will become familiar with computing related mathematics. **Skills:** Understand what and how to use a variable in programming, identify the different programming constructs, and identify the datatypes in programming and how to read/write to external files. Understand how design an algorithm for chosen problem Understand how to interpret algorithms **Formative Assessment**: Programming task **End point:** ▪ Students can explain the difference between a variable and constant	**Knowledge coverage:** ▪ Algorithms – representation, interpretation searching and sorting algorithms ▪ Data Representation - number systems, text, sound, images, data compression and data encryption **Skills tested**: ▪ Identify and explain key terminologies used in programming ▪ Ability of convert data from denary →binary → hexadecimal and vice versa ▪ Understanding of data compression methods ▪ Understanding of data encryptions for different file formats and types

	End point: ▪ Can explain what an algorithm is ▪ Able to explain key terminology – abstraction, decomposition and algorithmic thinking ▪ Able to design, interpret and correct algorithm ▪ Able to use pseudocode to represent an algorithm	▪ They can identify why programming concept to use when writing a program	**Assessment style/questions**: Exam style questions, combination of short written answers requiring students to state, explain, describe analyse and compare