Saint Martin's Catholic Academy

Computer Science - Year 10 Long Term Plan

Time allocation	3 lessons per week		
OCR GCSE Computer Science – 1.1 – Systems Architecture			
Key learning in this unit: Students learn the fundamental building blocks of computer architecture (the way in which computers are designed). Based on the seminal work of John Von Neumann in the mid to late 1940's, students will learn the history of computer design and the common threads that have run through computing ever since. Concepts introduced in this unit include CPU design, components of a CPU, buses and registers, common performance factors (cores, cache, clock speed) and also the concept of an embedded system. Students are provided with a copy of the 1.1 – Systems Architecture knowledge booklet and knowledge organiser to complement their lessons and home learning tasks.			
	Key learning, tasks and skills	**Opportunities for assessment**	**Specific knowledge focus**
Lesson 1 – The Von Neumann Architecture	• Students learn about the origins of computing and a short history of computer design. • Students read about John Von Neumann and answer questions based on his work, following a short video presentation. • Students are introduced to a diagram of the Von Neuman Architecture. This is discussed and tasks completed to reproduce and annotate the diagram. • Further reading and video resources are used to embed the basic principles of a "general purpose" computer design, its advantages and disadvantages.	• Do now activity. • Written tasks. • Q&A response.	• John Von Neumann was one of the first "computer scientists" – an almost unparalleled genius who lived in the USA during the Second World War and invented modern computing as we know it today. • Von Neumann Architecture – a blueprint design for "general purpose computers." Broadly, this specifies the main components of a computer and how they are connected. • A Von Neumann machine consists of some form of input, a memory store for programs and data the machine is working on, some form of output and a CPU which contains a control unit, ALU, Program Counter and an Accumulator register.

Lesson 2 – Common CPU Components	• In this lesson, students learn about the common components inside modern CPU's. They do not learn about a specific architecture, for example X64, Intel or AMD. This is beyond the scope of GCSE. Instead, a "generic" architecture is used which covers the main components of a "Von Neumann" CPU. • Students learn about the main phases of the fetch, decode, execute cycle. This is linked in with later learning about RAM and software specifically. • Students complete tasks to learn and match the correct definition of CPU components to their corresponding labels. • Students complete exam questions based on this topic.	• Knowledge check – Von Neumann Architecture. • Do now activity. • Written tasks. • Q&A response. • Exam question answers.	• Any modern CPU is made up of a similar set of core components – an ALU, control unit, program counter, a set of special purpose registers and some cache memory. • A CPU carries out something called the "fetch, decode, execute" cycle repeatedly from power on to power off. • Data and instructions are fetched from memory, carried out in the CPU and the results stored back in memory. • The ALU is responsible for carrying out nearly all instructions in the CPU instruction set. • The Control Unit coordinates the actions of the CPU during the fetch, decode, execute cycle. It usually houses the instruction decoder unit. It sends signals to various components to indicate what needs to happen at any given stage.
Lesson 3 – Registers and Buses	• This lesson introduces students to the CPU at a lower level than last time. We now look at the individual registers (small pieces of memory inside a CPU which can hold a single instruction or piece of data) that are found in our generic CPU architecture. This is the most critical lesson in this unit and a lot of time is spent on learning the low level fetch, decode, execute cycle and how each register is used throughout. • Students learn the name, role and purpose of each register	• Knowledge check – Components of a CPU • Do now activity. • Written tasks. • Q&A response. • Weekly HSQ • Exam question answers.	• There are a set of generic registers in most CPU's. These are the PC, MAR, MDR, CIR and Accumulator (ACC). • PC – Program Counter - holds the address of the next instruction to be executed. • MAR – Memory Address Register – holds the address which is to be read from or written to next. • MDR – Memory Data Register – holds the data which has been read from RAM or is about to be written to RAM • CIR – Current Instruction Register - Holds an instruction-data pair which has been decoded.

	• Students use their knowledge booklets and video resources provided to draw, annotate and learn the FDE cycle in detail. • Students complete various tasks based on the stages of the FDE cycle • Students answer exam questions based on this lesson and last.		• ACC – holds the results of the last executed instruction. • There are three common buses in a Von Neumann machine – the control, address and data bus. These connect the CPU to memory.
Lesson 4 – CPU Performance Factors	• This lesson introduces students to the three main factors which affect the performance of a CPU. They are already familiar with the idea of the fetch, decode, execute cycle so the first focus is on the effect of increasing this "clock speed." This is followed by the idea of parallel or multi-core processing and the addition of cache memory and how this enables the CPU to predict the flow of execution and to work at as near to 100% capacity as possible. • Students are introduced to each concept one at a time, using visual/animated examples to show what is happening during a clock cycle and how increasing this may affect performance or how adding another CPU would mean processes can happen in parallel. • Using knowledge booklets and other resources, students complete comprehension tasks based on these three concepts. • Students answer exam questions on the topics covered in this and previous lessons.	• Knowledge check – Registers and Buses • Do now activity. • Written tasks. • Q&A response. • Weekly HSQ • Exam question answers.	• There are three common factors which affect CPU performance – cores, cache and clock speed. • Clock speed is measured in hz, and in modern CPU's GHZ. • 1hz = 1 fetch, decode, execute cycle per second. • GHZ = billion – 1 billion cycles per second. • More cycles = more instructions executed per second. • A core is a processing unit within a CPU. Effectively, it is a CPU in its own right. • More cores = more instructions executed simultaneously. • Cache holds frequently used instructions or those which are expected to be executed next. • Cache is faster than RAM • Cache speeds up the CPU as it does not have to wait to read data from RAM
Lesson 5 – Embedded Systems	• This lesson introduces students to the concept of computers being	• Knowledge check – CPU performance factors	• An embedded system is any device which is not necessarily a computer

	"embedded" in other devices. This covers a wide range of technology from "smart" appliances, speakers, doorbells, push button activators such as the now defunct Amazon Dash buttons and so forth. Students investigate the impact of the rise of IoT devices and the security / environmental impacts this has – this links in with their learning later in the course during 1.6 – Legal and Ethical Impacts of Technology. • Students complete a range of tasks using their knowledge booklets and other resources. • Students complete exam questions based on this topic. • Students spend time revising this section of the course for future assessment.	• Do now activity. • Written tasks. • Q&A response. • Weekly HSQ • Exam question answers.	in its own right, but contains a computer. For example, a TV set top box or a smart watch. • Embedded systems usually consist of a cut down version of a computer. Processors are slower, resources are limited. • Embedded systems will usually contain a microprocessor. • Embedded systems are usually very robust and heavily tested to ensure they are reliable. • They can only carry out one very specific task or a small set of tasks – they are not generic, multi-purpose devices.