Parkside House School Key Stage 3 ICT and Computing Scheme of Work 2020 Year 1

Module	Theme	Learning Objectives	Coverage
Term 1			
Introduction to Networked Computing	Pupils learn how to use the school computer network to store and organise their work safely.	• How to log on to the school network with individual User accounts • Passwords, Security and Usernames. • How to use shared files, folders and Home and Classroom drives. • How to organise work by creating folders and how to save/load and manage files on their home area. • The advantages of using a school network to store and manage their work.	1. Hardware, software components, computer systems and networks 2. Use technology safely
Programming: Using a Visual Language 1 and Computing Basics	This module introduces programming via a visual based online programming course at code.org using BLOCKLY. The course can be differentiated to allow pupils of different abilities to access programming. BLOCKLY is similar to SCRATCH which some pupils may have used at Key Stage 2. It is based on JAVA text based programming language. This individual nature of the course encourages the pupils become owners of their own learning.	• What is program flow, using commands to create instruction sets known as *procedures*. • Algorithm basics and offline problem solving using graph paper programming. • Use the JAVA based "Blockly" visual language commands to solve basic puzzles introducing sequencing, debugging and the RUN, STEP and STOP functions in single LOOP code. • Controlling Events using *non-linear* visual programming linking user input via arrow keys to on-screen movement and adding sounds to events i.e. collisions and score counts. • How to use nested loops to improve code efficiency and computational thinking by creating pattern sequences in Artist and combing with earlier learning to create more complex problem solving algorithms in Frozen. • Pupils produce an ongoing working document of their progress	1. Hardware, software components, computer systems and networks (Cloud Storage and Online working) 2. Design, use and evaluate computational abstractions 3. Use 2 or more programming languages (BLOCKLY/JAVA) 4. Develop application skills in MS Office and/or a similar package.

		in programming using Power Point. This combines screenshots of their work on Code.org with a brief description of the task. • They practice screenshots, importing/editing images and formatting text. This does not count as a creative exercise.	
Components of Computer System	Pupils learn that a computer is an interconnected device combing soft (virtual) elements and hard (physical) components usually operating in a network to carry out their functions.	• Identify core components of a system i.e. CPU, RAM, Hard Drive and Motherboard, how the function and communicate with each other using software. • The characteristics of volatile and non-volatile storage e.g. RAM, Magnetic, Optical and Solid State storage. • Interactions between Operating Systems and Application software, operating systems ecosystems Mac OS, Windows, Android and introducing cross-platform working in Browsers/Cloud.	1. Hardware, software components, computer systems and networks 2. Understand how instructions are stored and executed within a computer system
Module	**Theme**	**Learning Objectives**	**Coverage**
Term 2			
Programming: Using a Visual Language 2	This element builds on last term's Code.org course; it develops the use of loops, procedures, and conditions. Pupils are faced with tasks that have more than one algorithmic solution and challenged by given coding limits that can be solved using the new commands such as *nested loops*.	• Correcting and improving code through debugging by fixing puzzles, simple algorithms loops and nested loops. Using repeat loops to create conditional code solutions using WHILE. • How to use IF/ELSE statements to create branching choices within code. Creating code that reacts to specific conditions it encounters in Bee. • Combining LOOPS and CONDITIONALS in Maze to create more complex and flexible solutions. Introducing UNTIL command to algorithms using problem solving in Harvester. • Pupils will add additional screenshots and descriptions to their Power Point programming log to aid in evidencing their	1. Design, use and evaluate computational abstractions 2. Use 2 or more programming languages 3. Develop application skills in MS Office and/or a similar package.

		progress and embed understanding.	
Solving Problems like a Programmer	This module gives pupils a structured approach known as *computational thinking*; the logical procedure used by programmers to create algorithms to solve real world problems. The module also introduces the idea that a problem may not have a solution i.e. it is practically impossible to create a 2m tower out of matchsticks and marshmallows. This is a progression from investigations in earlier years where there is usually an answer to every academic problem. Pupils learn there are limits to solutions and about how small scale problems can be used to *model* real world problems.	• How to break down problems using computational thinking i.e. *decomposition, pattern recognition, abstraction* and using pseudocode to plan *an algorithm* that solves the problem. Practical programming and problem solving exercises including: 1. Examining maximum server load in "Tin Foil Boats" practical. o What is the maximum load a boat can take? o What are the best designs to take the most loads? o Which designs are more robust than others? o How does this exercise relate to a real world computing problem? i.e. beta testing a game server for maximum number of players or stress testing an online shopping website. o How could hackers use this maliciously: DDOS attacks, suppressors etc. o How could a company protect themselves i.e. Cloudflare technology or limit numbers of players per server. 2. Open-ended problems with a "Marshmallows and Matchsticks" structure building task. Pupils are asked to build several structures of different heights to explore the following questions: o Which 3D shapes create a strong structure? o Which 3D shapes create a weak structure? o How the requirements change as structure height increases?	1. Design, use and evaluate computational abstractions (Basic introduction of key concepts of *computational thinking* and *pseudocode*) 2. Understand several key algorithms (num count and conditional routing) 3. Hardware, software components, computer systems and networks 4. Develop application skills in MS Office and/or a similar package.

		○ How can designs be modified to accommodate increased stresses? ○ Is there a maximum height that these materials can sustain? ○ How could we modify the materials to increase the maximum height? ○ How does this exercise relate to a real world computing problem? For example, what do we do when a system is no longer suitable for the purpose? i.e. a company moving from a peer-to-peer to server based network, using cloud storage to increase capacity, buying new hardware when systems become slow and outdated or when and why to buy a new phone/tablet/console. • Pupils analyse and reflect on how they approached the tasks and how effective their and their peers' strategies were in solving the tasks. • Pupils will use digital images or video, and text to record their solutions and strategies. These may combined on either MS Word, Power Point or similar. • How could we use *pseudocode* to describe these exercises and the real world examples? i.e. setting an max value for a game and checking if it has been reached if a new player tries to join? How could we set up a counting algorithm using a variables i.e. *playernumcount=X, maxplayers=50, if playernumcount=50 then*	
Module	**Theme**	**Learning Objectives**	**Coverage**
Term 3			

Parkside House School Key Stage 3 ICT and Computing Scheme of Work 2020 Year 1

Digital Citizenship 1	This unit builds on the work done in the first modules and in the annual E-Safety Day., with the intention of deepening pupils' knowledge in this area and introducing key terms and concepts in relation to digital literacy including *digital footprints* and being a good digital citizen.	• Students will understand how computer viruses are transmitted, how to recognise them and how they can reduce their risks of downloading them. • The importance of secure passwords in an online and interconnected world and the risks of insecure or shared passwords. • How to create strong and memorable passwords to protect their school work and online privacy. • To consider their *Digital* Footprint i.e. what they post online and how they are perceived through the digital footprints. • Compare their responsibilities to their online and offline communities, the risks of identity theft, phishing and scams. • To reflect on the characteristics of being a safe and civil online citizen.	1. Create, reuse, revise and repurpose digital artefacts 2. Use technology safely 3. Develop application skills in MS Office and/or a similar package.
Creative Programming	This is a challenge module designed to model practices in the working world giving pupils an opportunity to work collaboratively, allocating roles and tasks as well as re-apply their programming knowledge in a new context.	• Pupils will be given the opportunity to practise the skills they have developed in previous terms in an online game application *Play Lab* to create their own animated game. • This also allows for team working, group planning and debugging each other's code like a real game development team. • Students will create a manual or report on their game combining images and text.	1. Design, use and evaluate computational abstractions 2. Use 2 or more programming languages 3. Creative projects that involve selecting, using and combining multiple applications 4. Develop application skills in MS Office and/or a similar package
An Introduction to	Pupils will learn about how data	• How digital technologies use binary to represent images,	1. Hardware, software

Representing Data in Binary	is stored in a common medium: binary, and how this and other conventions allow different kinds of information to be stored , shared and used by all digital devices due to their expression in binary code.	colours and text i.e. ASCII and how this related back to the Hardware unit in term 1 where ASCII code links peripherals (keyboard) via the Operating System to Applications (MS Word) in a cross-platform standard . • How elements i.e. images are translated into a binary format that a computer can then process e.g. JPEGs and PNGs. • The course concludes with another practical exercise in Artist that demonstrates how the theory learnt earlier can translate images into binary on and off (black and white) pixels.	components, computer systems and networks 2. Understand how instructions are stored and executed within a computer system
End of Year Assessment		• The students will be given an end of year assessment based on the topics covered this year. • An assessment of progress and performance in class work and projects will made for each student. • These assessments will be combined to provide an overall assessment for the year and be reported to parents/carers via each student's annual report.	