

Parkside House ICT and Computing Year 1 Core Elements

Term 1. Introduction to Networked Computing

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.

Programming: Using a Visual Language 1 and Computing Basics

Introducing program flow, using commands to create instruction sets. Algorithm basics and offline problem solving using graph paper programming. Using 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 combining with earlier learning to create more complex problem solving algorithms in Frozen.

Components of Computer System

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.

Term 2. Programming: Using a Visual Language 2 and Solving Problems like a Programmer

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.

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 examining maximum server load in “Tin Foil Boats” practical and open-ended problems with a “Marshmallows and Matchsticks” structure building task.

Term 3. Digital Citizenship 1, Creative Programming, and Basic Binary

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 and how to create strong and memorable passwords to protect their school work and online privacy. To consider 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.

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.

Introduction to using binary beyond a series of zeros and ones to represent images, 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.