KS2 Prior Computer Knowledge

- Design, write and debug programs that accomplish specific goals.
- Use sequence, selection, and repetition in programs.
- Understand computer networks including the internet and use search technologies effectively.
- Select, use and combine a variety of software on a range of digital devices to design and create a range of programs.
- Use technology safely, respectfully and responsibly.

Autumn Term 1

1

Usernames and Passwords Digital Citizenship
Students learn of organisation within Computing as well as basic proficiencies within Microsoft. Safety of being online features heavily

Autumn Term 2

2

Digital Citizenship continued then moving onto - Using Media
Students learn basic software skills and legislation, trustworthiness of sources–for example: word processing, spreadsheet, desk top publishing, copyright, researching and blogging

Spring Term 1

3

Using Media continued then moving onto - Desk Top Publishing
Students learn to create documents used by businesses

Spring Term 2

4

Desk Top Publishing continued

Summer Term 1

5

Introduction to Data Representation
Students explore the concept of Data Representation Binary, ASCII, binary addition, Hexadecimal Number Systems and Conversions

Summer Term 2

6

Introduction to Control Technologies
Students learn the basic skills they need to understand algorithms, the construction of flow charts and the use and sequencing of events via Flowol

SKILLS TAUGHT ACROSS COMPUTERS
Throughout all topics students will develop skills in Algorithms and Flowol programme. Business documents. Make/predict the outcome of a simple sequence that includes variables

Skills Development, Key terms:

- E-safety (passwords, cyberbullying, stranger danger topics)
- Presenting to an audience, how to give peer feedback and responding to feedback
- Understand different software features and their use
- Copyright designs and patents act
- Credibility of sources
- Researching, acknowledging sources and creating a blog
- Recognise that computers follow the control flow of input/process/output
- Create conditions that use logic operators (and/or/not)
- Define iteration as a group of instructions that are repeatedly executed
- Implement count-controlled iteration in a program
- Detect and correct errors in a program (debugging)