

Unit 4

| | |
|--|--|
| Topic | Algorithms |
| Topic overview Pupils will learn... | Understand the fundamentals of computational thinking (abstraction, decomposition, pattern recognition and algorithms) to solve a range of problems through flow diagrams and text-based programming |
| Components | <ul style="list-style-type: none"> • Understand the concepts of abstraction and decomposition in order to be able to focus on the most important aspects of a problem and break down large problems into manageable chunks • Design algorithms to solve a range of computational problems • Analyse the effectiveness of different approaches to solving problems in order to identify and apply the most efficient techniques when creating algorithms • Combine the principles of abstraction, decomposition and algorithm design with pattern recognition to solve a range of problems in order to replicate the user of algorithms to solve real world problems • Understand the benefits of a modular approach to programming to be able to develop efficient solutions for specific aspects of a larger problem • Recognise the standard symbols used in flow diagrams be able to read flow diagrams in order to visualise a solution to a problem and be able to design algorithms in the form of flow diagrams |
| What pupils should already know (Prior learning components) | <p>Within the KS3 Computing, students should have been taught to –</p> <ul style="list-style-type: none"> • create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability • understand basic key algorithms that reflect computational thinking and use logical reasoning to compare alternative algorithms for the same problem • used a block-based programming language to solve a variety of computational problems • examine a set of basic flow diagrams in order to determine the sequence of an algorithm and predict outcomes (As well as recognise the basic purpose of each shape) |
| Transferrable knowledge (skills) | <ul style="list-style-type: none"> • Being able to access computer systems, navigate to specific files and organise work in a logical structure. • Being able to use multiple pieces of software (such as a web browser, presentation software, word-processing software and a cloud computing system) in quick succession to create and refine design projects • Use of inference and articulation to obtain key knowledge from a topic and apply understanding when presenting findings |

| | |
|---|---|
| | <ul style="list-style-type: none"> Being able to break down a large problem into smaller aspects and remove unnecessary details to identify key aspects |
| Key vocabulary pupil will know and learn | computational thinking, decomposition, abstraction, algorithm design, algorithm, pattern recognition, input, output, flow, process, decision |
| Assessment activities | <ul style="list-style-type: none"> Regular low stakes testing at the end of each lesson to check knowledge. Practical lesson activities with digital activities assessed by teachers Do Now tasks which test previous learning and build recall on key terms and applying them to specific contexts |
| Resources available | <p>KS3 NC information National Curriculum - Computing key stages 3 and 4 (publishing.service.gov.uk) BBC Bitesize reference to computational thinking Computational thinking - KS3 Computer Science - BBC Bitesize BBC Bitesize reference to flowcharts Flowcharts - Designing an algorithm - KS3 Computer Science Revision - BBC Bitesize</p> |
| Notes Why this topic is important... | <p>The term 'algorithm' has come to refer to any set of rules that precisely define a sequence of operations, such as making a cup of tea or cleaning your teeth. In the world of computing, an algorithm is a set of instructions that can be implemented as code to program a computer. Computational thinking is a logical, strategic approach to problem solving involving four cornerstones: decomposition, abstraction, pattern recognition and algorithm design to formulate an efficient and effective algorithm. These principles are key to being able to explore a problem and design an effective algorithm to solve it.</p> <p>Following on from an introduction to algorithms in the Year 8 roundelay, this unit introduces students to the use of computational thinking to solve problems. They will learn about three of the four cornerstones of the computational-thinking approach to problem solving: decomposition, abstraction and algorithm design. They will then apply what they have learnt to create a number of different algorithms, which will support them to develop logical thinking skills and support them in understanding programming and algorithm design in GCSE Computer Science.</p> <p>The unit also builds upon the use flow diagrams, first introduced in the Year 7 unit 5, and students will learn how to use the functions of some standard flow diagram symbols and how to link them together to represent an algorithm. They will apply this learning to practical activities that will enable them to practise flow-diagram design and consolidate their comprehension of flow diagrams and how they relate to algorithms.</p> <p>Programming forms one of two major components within GCSE Computer Science and as such is a core element of KS3 Computing. This unit will enable students to program in a text-based language, and build upon smaller programming activities from using a block-based language within Year 7 Computing.</p> |