

# Computing

## CURRICULUM OVERVIEW

### Pearson Edexcel iPrimary is not just a curriculum, but a complete teaching toolkit.

The English, Mathematics, Science and Computing curricula have been developed specifically for the international student, and a complete suite of teaching support is included as part of the whole-school package. Exemplar schemes of work, units of work and lesson plans are included for all year groups in addition to fully integrated internal (Progress Tests) and external (Year Six Achievement Tests) assessment, all created through Pearson Edexcel's World Class Qualifications framework.

In addition, a comprehensive programme of Professional Development support is available - which includes face-to-face training, webinar training, and being part of our community forum. Plus the whole teaching and learning solution is supported by Pearson's world-renowned teaching resources such as Bug Club, Science Bug and Abacus.

The iPrimary curriculum is organised into six year groups, each building on the knowledge and skills of the last. Each provides comprehensive learning objectives and lesson plans to ensure clear targets and progression for students.

For iPrimary Computing, the curriculum contains four main strands, with each split into topic areas. The curriculum promotes engagement and enjoyment while ensuring students are well placed to achieve highly in later examinations.



### The strands are:

**Problem Solving – Algorithms**

**Create Algorithms**

**Computers - Hardware and Software**

**Communication and Networks - Social Web and Online Safety**

**Information Technology**

**Programming and Development**

Students are encouraged to take an engaged and investigative approach to their learning. The iPrimary Computing curriculum gives an excellent platform for later learning and ensures students are prepared for the challenges ahead of them.

On the following page are examples of objectives from the iPrimary Computing curriculum topics for Years 2, 4 and 6.

## YEAR 1

### Problem Solving - Algorithms

- > Understand what an algorithm is
- > Understand what algorithms are used for
- > Understand how (algorithms) are implemented as programs on digital devices
- > Be able to interpret algorithms

### Create algorithms

- > Use logical reasoning to predict the behaviour of simple algorithms
- > Be able to create a simple program
- > Understand how to identify and correct errors in programs (bug and debug)

## YEAR 3

### Computers - Hardware and Software

- > Be able to select appropriate digital devices for a given task or project
- > Know different types of input and output peripheral
- > Be able to select appropriate software or online service that meets the specified needs of a project
- > Understand the key features of a word processor and presentation software

### Communication and Networks - Social Web and Online Safety

- > Know safe, responsible and respectful practice when using digital technologies
- > Be able to recognise unacceptable behaviour and report concerns about content, contact and conduct to an appropriate individual or organisation

## YEAR 5

### Information Technology

- > Select, use and combine appropriate software applications (word processing, spreadsheet, presentation [multimedia] software, graphics, and desktop publishing) to design and create a range of content that accomplishes given goals, including collecting, analysing, evaluating and presenting data and information

### Programming and Development

- > Be able to locate and fix errors (bugs) in a program
- > Understand the structural components of a program (sequence, repetition, selection, inputs and outputs)
- > Be able to use sequencing, repetition/iteration and selection constructs in their programs
- > Understand how to write code that accepts and responds appropriately to user input
- > Be able to formulate a realistic idea that could be turned into a program
- > Be able to plan a program including objects, initialisation and algorithm

