

Computing

CURRICULUM OVERVIEW

Pearson Edexcel iLowerSecondary 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 and mapping enable you to access Pearson's world-renowned teaching resources. In addition fully integrated internal (Progress Tests) and external (Year Nine Achievement Tests) assessments are available, 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 iLowerSecondary curriculum is organised into three 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 iLowerSecondary Computing, the curriculum contains five 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 and Abstraction**
- Programming and Development**
- Data and Representation**
- Information Technology - Application of IT**
- Software Skills - Web Authoring**



Students are encouraged to take an engaged and investigative approach to their learning. The iLowerSecondary 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 7, 8 and 9.

YEAR 7

Problem Solving: Algorithms and Abstraction

- > Be able to code an algorithm in both a visual and textual language
- > Be able to compare the utility of alternative algorithms for the same problem
- > Use computational abstractions that model the state and behaviour of real-world problems and physical systems.

Programming and Development – Errors

- > Be able to locate and fix syntax errors in a program
- > Be able to interpret an error report from a textual language IDE.

YEAR 8

Programming and Development - Programming Constructs

- > Understand and use variables and type declarations.
- > Understand and use sequence, selection and iteration

Data and Representation – Binary

- > Understand how to perform simple binary arithmetic (addition).
- > Understand how bitmap images are represented in binary.
- > Understand and be able to create a binary word to represent a simple bitmap image.
- > Understand and be able to create a simple bitmap image that is represented by a binary word.
- > Understand and use the terms describing capacity of storage
- > Be able to convert between binary and denary multiples of values for storage
- > Understand that file storage is measured in bytes and be able to calculate storage requirements

YEAR 9

Information Technology - Application of IT

- > Be able to select and use multiple applications
- > Be able to collect data
- > Be able to meet the needs of known users
- > Be able to reuse digital artefacts for a given audience
- > Be able to bring together and organise different types of information to achieve a purpose
- > Be able to work accurately and proofread, using software facilities where appropriate for the task

Software Skills - Web Authoring

- > Use HTML, including hyperlinks, bookmarks and anchors, inserting images/animations/sound, font enhancements, text (including title, body, paragraph), lists (including simple, bulleted), head (including meta data), comments to create web pages

