

# Computing

## Preparing students for tomorrow, bit by bit

The Computing department will help to create, share, and apply knowledge in all branches of Computer Science and ICT. We will educate students to be successful, ethical, and effective problem-solvers with a passion to innovate and create, rather than just passive consumers and users of technology. We will develop an understanding and appreciation of all aspects of digital products, from how they work to how they look. We will foster curiosity and encourage exploration to create students who can contribute positively to the well-being of our society and who are prepared to tackle the complex 21st Century challenges facing the world.

Summary focus areas:

- Innovate, create, develop
- Solving 21st Century problems
- Active developers not passive consumers

Autumn	Spring	Summer
Networking technologies and methods  The Internet	Conventional programming (Java): Basic constructs, error handling, object-orientism, graphical interfaces	

Homework for Computing is set weekly to support and extend the students' studies from their lessons. Work may be a mixture of practical, computer-based tasks and paper-based written work or design tasks. Activities set as homework may be:

- preparatory work or research ahead of a new topic or concept being discussed in lessons
- extension work that allows the student to explore a topic in more depth or in other contexts
- application work that allows students to practise skills or demonstrate abilities

Students are expected to spend around an hour on a homework activity each week and work is marked promptly to help students to identify and understand their weaknesses to make incremental improvements over the course of the year.

Unit	Duration (lessons)	Learning Objectives/Outcomes
Networking and the Internet	12	<ul style="list-style-type: none"> <li>• understand what a computer network is</li> <li>• be able to discuss the advantages and disadvantages of using a computer network</li> <li>• be able to describe and explain the bus, ring and star networking topologies</li> <li>• be able to explain the different hardware needed as part of a network system and the role each piece of hardware plays</li> <li>• be able to discuss the advantages and disadvantages of each of these topologies</li> <li>• understand the client-server model</li> <li>• be able to explain, in simple terms, the handshake process used in most modern networking protocols</li> <li>• be able to explain how coding for a client-server model is different from coding for a stand-alone application</li> </ul>
Web programming	34	<ul style="list-style-type: none"> <li>• understand the concept of coding at the server and client end</li> <li>• know what can be coded at the server end</li> <li>• know what can be coded at the client end</li> <li>• have experience of coding solutions to simple web application problems</li> <li>• know about and be able to describe common built in functions in their chosen language(s)</li> <li>• use common built-in functions in their chosen language(s) when coding solutions to problems</li> <li>• understand what a parameter is when working with procedures and functions</li> <li>• know how to use parameters when creating efficient solutions to problems</li> <li>• understand the concepts of parameters and return values when working with procedures and functions</li> <li>• know what is meant by the scope of a variable, constant, function or procedure</li> <li>• be able to identify what value a particular variable will hold at a given point in the code</li> <li>• be able to discuss and identify the different types of errors that can occur within code (i.e. syntax, run-time and logical)</li> <li>• understand that some errors can be detected and corrected during the coding stage</li> <li>• understand that some errors will occur during the execution of the code</li> <li>• know how to detect errors at execution time and</li> </ul>

		<p>how to handle those errors to prevent the program from crashing where desirable</p> <ul style="list-style-type: none"> <li>• be able to use trace tables to check their code for errors</li> <li>• understand that computer programs can be developed with tools to help the programmer detect and deal with errors (e.g. Watch, Breakpoint, Step)</li> <li>• know how to use an external text file to read and write data in a way that is appropriate for the programming language(s) used and the problem being solved</li> <li>• understand the basic concepts of a relational database as a data store</li> <li>• be able to explain the terms record, field, table, query, primary key, relationship, index and search criteria</li> <li>• be able to create simple SQL statements to extract, add and edit data stored in databases</li> <li>• have experience of using these SQL statements from within their own coded systems</li> <li>• be able to use databases from within their own web based applications</li> <li>• know how to read and write data from an external database in a way that is appropriate for the programming language(s) used and the problem being solved</li> </ul>
Controlled assessment 1 (Web programming)	26	<ul style="list-style-type: none"> <li>• understand the requirements of the controlled assessment brief</li> <li>• be able to produce a fully working and documented solution to the given problem</li> </ul>