|  |  |
| --- | --- |
|  | **2012-13** |
|  | Rodborough School  M Walker |

|  |
| --- |
| **Computing Schemes of work** |
| KS3 enquiry-based curriculum, following on to mandatory 1 year IT (year 10) and optional GCSE Computer Science (years 10 & 11) |

Overview

This is an ‘enquiry-based’ curriculum where students are presented with a question at the start of the course and through a combination of their own research and classroom guidance are – hopefully - in a position at the end of the topic to answer the question.

Learning Strands

Year 7

Year 8

Year 9

Key Stage 3 Level Descriptors

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Systems** | **Development** | **Programming** | | **Modelling** | | **Analysis** |
| 3 | Understand that computer systems work step-by-step and can only do what we tell them. | Be able to create a sequence of instructions and improve it if necessary. | Be able to plan a sequence of instructions for something that you want to happen. | | Be able to read a sequence of instructions and predict what the result will be. | | Be able to describe the goals of a given problem. |
| 4 | Be able to explain why we must be accurate when working with computers. | Write sequences of instructions and data in a way that a computer will understand. | Use selection and repetition correctly in your programs. | | Be able to trace instructions using variables, selection and repetition and predict what the result will be | | Understand what is meant by a computational problem. |
| 5 | Understand how data, such as numbers, sound and images are physically stored on a computer system. | Be able to plan, create, test and reflect on a solution to a problem that a computer could solve. | Correctly use variables, lists and simple procedures in your programs. | | Be able to recognise similarities between simple problems and the ways in which they can be solved. | | Be able to take a problem and divide it into its main sub-problems. |
| 6 | Understand how instructions are run inside a computer. | Be able to develop solutions for problems that are described to you by someone else. | Correctly use procedures and functions with parameters in your programs. | | Be able to take solutions to one problem and adapt them for similar problems. | | Be able to take a problem and divide it into all its sub-problems and show this as a diagram. |
|  | **Computer Systems** | **Development Process** | **Programming Skills** | | **Modelling** | | **Analysis** |
| 7 | Understand how instructions can be written efficiently and be able to describe the efficiency of your programs. | Be able to test the different modules of your programs as you are developing them, reflect on the results and then improve them. | Be able to write programs in a text-based language like Python and be able to create your own data structures. | | Be able to create a simple model for a complex problem. | | Be able to define an outline of a solution in terms of functions and global values. |
| 8 | Be able to show how elements of real life can be represented in programs and the difficulties that sometimes exist when doing this. | Make sure that the programs you develop have been written so they are unlikely to crash or cause errors. | Be able to create your own relational databases and use them in your programs and be able to find, understand and use techniques for specific tasks. | | Be able to create an accurate, detailed model for a complex problem. | | Be able to analyse real world problems and develop low-level and high-level plans for a solution. |
| EP | *Learners have a thorough conceptual understanding of computer systems and can competently and confidently use a general-purpose text-based programming language to produce efficient and robust solutions to complex problems.* | | |  | |  | |

These level descriptors are based on the proposed Computing at School curriculum (<http://academy.bcs.org/upload/pdf/curriculum-computing-schools.pdf>) with alterations and amendments.

Mapping to other KS3 Curriculum

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Topic | Overview Content | CAS Curriculum (2012) | NaaCE Curriculum (2012) | NC ICT PoS (2007) |
| 7.1 Why is Facebook successful? | * Use of school system including VLE and email * Generic Office application use * Online Safety |  | * Safe and Responsible Use – Online identities, Personal e-safety * Digital Literacy – Online Identities * Technology in the World – Common Productivity Software and Applications | * 1.1b * 1.2a * 1.4b |
| 7.2 Why are video games fun? | * Programming with sensor boards and Scratch * Reactive systems | * Algorithms 1 * Programs 1, 2, 3, 4, 6, 7 * Computers 1 | * Digital Literacy – Gaming * Skills – Producing and editing all types of media * Skills – Modelling, Control Data Logging and Programming, Problem Solving * Technology in the World – Design and Specifications, Creative Industries * Technical Understanding – Programming and Control | * 1.1a, 1.1b, 1.1c * 1.3a, 1.3b * 2.2b, 2.2c, 2.2d, 2.2e |
| 7.3 How can I make an unbreakable code? | * Using spreadsheets to encrypt, decrypt and crack data. * Use of binary number system and characters sets | * Algorithms 1, 3 * Programs 1, 3, 4, 5, 6, 7 * Data 1, 2.1 * Computers 1 | * Safe and Responsible Use – Legal issues, Legislation concerning ICT * Skills – Modelling, Control Data Logging and Programming, Problem Solving * Technology in the World – Common Productivity Software and Applications, Design and Specifications * Technical Understanding – Programming and Control | * 1.1a, 1.1c * 1.3a, 1.3b, 1.3c * 2.2b, 2.2c, 2.2d, 2.2e |
| 7.4 Can I teach a machine to think? | * Investigation into basic AI, machine learning, game strategies and Turing test | * Algorithms 1, 3, 4 * Programs 1, 6, 7 * Data 1, 2.1 * Computers 1, 2 | * Digital Literacy – Impact of ICT on society * Skills - Control Data Logging and Programming, Problem Solving * Technical Understanding – Programming and Control | * 1.1a, 1.1b * 1.3a, 1.3b * 2.2e |
| 8.1 Why is programming simple? | * Designing and programming with functions using Scratch BYOB, sensor boards and robotic arms | * Algorithms 1, 2, 3, 4 * Programs 1, 2, 3, 4, 5, 6, 7 * Computers 1, 2 | * Skills – Producing and editing all types of media, Modelling, Control Data Logging and Programming, Problem Solving * Technology in the World - Design and Specifications, Creative Industries * Technical Understanding – Programming and Control, ICT Systems Lifecycle, Organisation of Data and Data Standards | * 1.1a, 1.1b, 1.1c * 1.3a * 2.2b, 2.2c, 2.2d, 2.2e |
| 8.2 Why do computers need electricity? | * Investigating what makes a computer system and how the components work together | * Data 1, 2.1, 2.2, 3, 4 * Computers 1, 2, 3, 4, 5 | * Skills - Problem Solving * Technology in the World – Common Productivity Software and Applications, Progress and Future Applications * Technical Understanding – How Computing Devices Work, Embedded Systems, Data Storage |  |
| 8.3 How can I make a pig fly? | * Using graphics software to create and manipulate digital images | * Data 2.4 | * Digital Literacy – Creating and sharing * Skills – Producing and editing all types of media * Technology in the World – Creative Industries | * 1.1a, 1.1b * 1.3a |
| 8.4 How does Google work? | * Searching for data in the web and creating webpages | * Algorithms 1, 3, 4 * Programs 1, 4, 6, 7 * Communication and the Internet 1, 2.1, 2.2, 2.3, 2.4, 2.5, 4 | * Safe and Responsible Use – Computer e-safety, Ethical issues * Digital Literacy – Finding retrieving and validating information, Critical thinking and evaluation * Skills – Modelling, Control Data Logging and Programming, Problem Solving * Technology in the World – Common Productivity Software and Applications, Web Design, e-commerce * Technical Understanding – Networks * Technical Understanding – Programming and Control, Organisation of Data and Data Standards | * 1.1a, 1.1b * 1.2a * 1.3a, 1.3c * 1.5a * 2.1b * 2.2b, 2.2c, 2.2d, 2.2e * 3b |
| 9.1 How is software really written? | * Using Python to develop solutions to problems | * Algorithms 1, 2, 3, 4 * Programs 1, 2, 3, 4, 5, 6, 7 * Computers 1, 2 | * Skills – Producing and editing all types of media, Modelling, Control Data Logging and Programming, Problem Solving * Technology in the World – Common Productivity Software and Applications * Technology in the World - Design and Specifications * Technical Understanding – Programming and Control, ICT Systems Lifecycle, Organisation of Data and Data Standards | * 1.1a, 1.1c * 1.3a, 1.3b, 1.3c * 2.2b, 2.2c, 2.2d, 2.2e * 3b |
| 9.2 How can I make a mobile app? | * Using MIT AppInventor to create Android mobile apps | * Algorithms 1, 2, 3, 4 * Programs 1, 2, 3, 4, 5, 6, 7 * Computers 1 | * Skills – Producing and editing all types of media, Modelling, Control Data Logging and Programming, Problem Solving * Technology in the World - Design and Specifications, Creative Industries * Technical Understanding – Networks * Technical Understanding – Programming and Control, ICT Systems Lifecycle, Organisation of Data and Data Standards | * 1.1a, 1.1b, 1.1c * 1.2a * 1.3a, 1.3c * 2.2b, 2.2c, 2.2d, 2.2e * 3b |
| EMBEDDED THROUGHOUT |  |  | * Safe and Responsible Use – Offline Safety * Digital Literacy – Functional Use, Social Networking, Learning, Impact of ICT on Society * Skills – Digital Communication Including Online Environments * Technology in the World – Collaboration and Communication Tools and Use | * 1.4b * 1.5b * 2.1a [although not necessarily in the way it was intended] * 2.2a, 2.2f * 2.3a, 2.3b, 2.3c * 2.4a, 2.4b, 2.4c * 3c, 3d * 4a, 4b, 4c, 4d |
| AREAS NOT COVERED |  | * Data 5, 6 * Computers 6 * Communication and the Internet 5 (5.1, 5.2) | * Safe and Responsible Use – Environmental Issues * Skills – ICT Skills for Career Paths and Working Life * Technology in the World – Use of ICT in Practical Contexts [as described in the curriculum], Working with ICT * Technical Understanding – Industry Standards | * 1.4a * 2.1c * 2.1d * 3a, 3e * 4e, 4f [dependent on other areas of learning] |