Computer Science

Year 11 Curriculum Map

Year 11 – Autumn Term

Prior Learning	The key stage 3 curriculum provides pupils at key stage 4 with the fundamentals of programming concepts such as algorithms, abstraction, decomposition and computational thinking. Kodu Game Lab, Scratch, Micro: bit and Python practical programming provides the skills, knowledge and understanding for pupils to hit the ground running when opting for Computer Science at key stage 4.
What will I learn?	**Algorithms** This unit begins by looking at computational thinking, including abstraction and decomposition. Two lessons are given to interpreting and comparing relevant searching and sorting algorithms including the merge and insertion sorts. These are written in the new OCR Reference Language. Practical experience of writing, tracing and modelling algorithms using pseudocode and flowcharts is then provided. Students are also be given ample practical experience of correcting and completing algorithms (including debugging and testing) in worksheets and homework tasks. **Programming Fundamentals** This programming unit covers the theoretical aspects of Section 2.2 of the latest J277 OCR GCSE Computer Science specification, covering all the knowledge and skills that students will need to tackle exam questions in Paper 2. The basic programming constructs are covered as well as string manipulation and file handling. Iteration and arrays are subsequently covered, before examining the use of procedures and functions to structure code. Finally, records and the use of SQL to search for data are covered. The unit is independent of any particular programming language but basic knowledge and practical experience of Python programming in a language is assumed.
How will I be assessed?	**Formal Assessments** - Each unit of work is assessed with a written examination on completion. **Formative Assessment** - will take place as part of recall and retrieval practice, low stakes testing, self and peer assessment to identify gaps in knowledge, monitoring and tracking progress throughout.
Next Steps	Computational thinking explores essential programming skills which naturally progresses to programming fundamentals so laying the foundation of more technical aspects required. The work covered next term will require pupils to think, learn and create like a programmer.
Opportunities for Independent Learning	https://student.craigndave.org/videos/slr2-1-algorithms https://student.craigndave.org/videos/slr2-2-programming-fundementals https://student.craigndave.org/videos/slr2-3-producing-robust-programs https://student.craigndave.org/videos/slr2-4-boolean-logic https://student.craigndave.org/videos/slr2-5-programming-languages-and-ides Specimen Paper 1 Walkthrough - https://www.youtube.com/watch?v=YuUt37FoQig Specimen Paper 2 Walkthrough - https://www.youtube.com/watch?v=Rkc50S-tj4A https://www.gcsepod.com/ www.educake.co.uk https://www.bbc.co.uk/bitesize/examspecs/zmtchbk Essay Style Questions - https://www.bbc.co.uk/bitesize/guides/zmm7xfr/revision/1
Personal Development and CEIAG	As pupils develop their skills in a range of software, they are challenged to work in groups to find solutions whilst developing respect for the ideas and opinions of others in their team. This is particularly prevalent in the design phase of tasks given. In addition, pupils are encouraged to develop their team working skills through collaborative work and research. The pupils also explore the concept of teams and the roles that individuals have

	to play. Computing can also help all pupils to express themselves clearly and to communicate. Jobs directly related to your degree include: • Application analyst • Applications developer • Cyber security analyst • Data analyst • Forensic computer analyst • Game designer • Games developer • Machine learning engineer • Penetration tester • Software engineer • Systems analyst • UX designer • Web designer • Web developer Jobs where your degree would be useful include: • Business analyst • IT sales professional • IT trainer • Nanotechnologist • Network engineer • Telecommunications researcher Further Career Research https://icould.com/explore/categories/subject/computing/
Enrichment Opportunities (Cultural Capital)	The Computer Science department have put together a list of enrichment opportunities available: **Computer Science Club** – Lunchtime activities to build coding skills. **Arcade Gaming Centre, Bury** – https://www.arcadeclub.co.uk/bury/ **Belong Gaming Arenas, Nationwide** - https://www.belong.gg/ • The Imitation Game (Problem Solving) • Source Code (Loops and Iteration) • Minority Report (Artificial Intelligence)

	Year 11 – Spring Term
Prior Learning	For the topic area of 2.3, defensive design and testing programs, pupils will recall and retrieve good practice from key stage 3. These learning experiences from key stage 3, provide pupils with the fundamentals of testing, having completed projects based in Kodu, Scratch, Python and Micro: bit. Unit 2.5 builds on the skills, knowledge and understanding developed in the KS3 program of study based around Python programming language. Pupils explore integrated development environment to write, run, check and test programs designed and created.
What will I learn?	**Producing Robust Programs – Defensive Design** Understanding the challenges, a programmer should consider ensuring that a program caters for all likely input values and how to deal with invalid data in a program.

	Authentication to confirm the identity of a user and the practical experience of designing input validation and simple authentication (e.g., username and password). Pupils will understand why commenting is useful and apply this appropriately. **Producing Robust Programs – Testing** The difference between testing modules of a program during development and testing the program at the end of production. Syntax errors as errors which break the grammatical rules of the programming language and stop it from being run and translated. Logic errors as errors which produce unexpected output. Normal test data as data which should be accepted by a program without causing errors. Boundary test data as data of the correct type which is on the very edge of being valid. Invalid test data as data of the correct data type which should be rejected by a computer system. Erroneous test data as data of the incorrect data type which should be rejected by a computer system. Ability to identify suitable test data for a given scenario ü Ability to create/complete a test plan. **Programming Languages and Integrated Development Environment** The differences between high- and low-level programming languages and the need for translators. The differences, benefits and drawbacks of using a compiler or an interpreter. Knowledge of the tools that an IDE provides and how each of the tools and facilities listed can be used to help a programmer develop a program. Practical experience of using a range of these tools within at least one IDE.
How will I be assessed?	**Formal Assessments** - Each unit of work is assessed with a written examination on completion. **Formative Assessment** - will take place as part of recall and retrieval practice, low stakes testing, self and peer assessment to identify gaps in knowledge, monitoring and tracking progress throughout.
Next Steps	Revision of year 10 and 11 ready for GCSE written exams in May/June.
Opportunities for Independent Learning	https://student.craigndave.org/videos/slr2-1-algorithms https://student.craigndave.org/videos/slr2-2-programming-fundementals https://student.craigndave.org/videos/slr2-3-producing-robust-programs https://student.craigndave.org/videos/slr2-4-boolean-logic https://student.craigndave.org/videos/slr2-5-programming-languages-and-ides Specimen Paper 1 Walkthrough - https://www.youtube.com/watch?v=YuUt37FoQig Specimen Paper 2 Walkthrough - https://www.youtube.com/watch?v=Rkc50S-tj4A https://www.gcsepod.com/ www.educake.co.uk https://www.bbc.co.uk/bitesize/examspecs/zmtchbk Essay Style Questions - https://www.bbc.co.uk/bitesize/guides/zmm7xfr/revision/1
Personal Development and CEIAG	Computational thinking encourages pupils to develop and explore their problem-solving skills. Computing allows pupils to apply their ICT and computing skills and to gain knowledge of how programming links between cross curricular subjects, especially Maths. Pupils are continually reflecting on their own lives and the lives of others as they look at various ICT and Computing case studies. Pupils debate and formulate their own set of values and beliefs through case studies as they share their own experiences. Computing provides opportunities for reflection of awe and wonder about the achievements in ICT today and the possibilities for the future. ICT lets pupils can reflect on how computers can sometimes perform better in certain activities than people. Jobs directly related to your degree include: • Application analyst • Applications developer • Cyber security analyst • Data analyst • Forensic computer analyst

	• Game designer • Games developer • Machine learning engineer • Penetration tester • Software engineer • Systems analyst • UX designer • Web designer • Web developer Jobs where your degree would be useful include: • Business analyst • IT sales professional • IT trainer • Nanotechnologist • Network engineer • Telecommunications researcher Further Career Research https://icould.com/explore/categories/subject/computing/
Enrichment Opportunities (Cultural Capital)	The Computer Science department have put together a list of enrichment opportunities available: **Computer Science Club** – Lunchtime activities to build coding skills. **Arcade Gaming Centre, Bury** – https://www.arcadeclub.co.uk/bury/ **Belong Gaming Arenas, Nationwide** - https://www.belong.gg/ • The Imitation Game (Problem Solving) • Source Code (Loops and Iteration) • Minority Report (Artificial Intelligence)