Computing Curriculum Overview

Subject Curriculum Intent Statement

Our curriculum vision within Computing is to develop student's skills and knowledge in digital literacy and technologies to prepare them for life both now and in the future. We develop computational thinking and software skills to create confident, creative and resilient problem solvers who will become lifelong users of IT. Teaching and learning focusses on solving (scalable) real-life problems and proficiency in software while promoting strong links to literacy and numeracy.

This will be achieved through;

- Understanding and applying the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
- Analysing problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
- Evaluating and applying information technology, including new or unfamiliar technologies, analytically to solve problems
- Developing a wide range of digital software skills including the ability to select the most appropriate software for a task and create documents that are fit for purpose and audience.
- Understanding the benefits and risks of living in a digital world and making informed judgments around the suitability, safety and sustainability of digital sources
- Being responsible, competent, confident and creative lifelong users of information and communication technology.

Computing Curriculum Offer @ SNA

- **Year 7 – Computing** – one period per week – all students
- **Year 8 – Computing** – one period per week – all students
- **Year 9 – Computing** – one period per week – all students

In addition, we offer the following optional courses:

Key Stage 4 – Years 9-11

- GCSE Computer Science

Key Stage 5 - Years 12-13

- A-Level Computer Science

Computing Curriculum Map

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 7	7.1 - Office 365	7.2 Web Awareness	7.3 Under the hood - Intro to Computer Systems	7.4 – Scratch	7.5 Digital Graphics	7.6 Microbits
Year 8	8.1 - AppLab		8.2 - Networking	8.3 - Python	8.4 – Data Representation	8.5 – Make code arcade
Year 9	Cyber security	9.3 – Python	9.3 - Python	9.4 – Boolean Logic	9.5 – Physical Computing	
GCSE Computer Science						
Python Programming Interleaving						
Year 10	1.1 System Architecture 1.2 Memory 1.2 Storage 2.2 Programming Fundamentals	2.2 Programming Fundamentals 2.3.1 Defensive Design 2.3.2 Testing	1.6 Ethical, legal, cultural 1.2.3 Units 1.2.4 Data Storage 2.1 Algorithms	1.2.4 Numbers, Characters 2.4 Boolean logic	1.2.4 Images, sound 1.2.5 Compression	1.6 Ethical, legal, cultural
Python Programming Interleaving						
Year 11	1.3 Networks 1.3.2 Wired and wireless networks, protocols and layers 1.4.1 Threats to computer systems and networks 1.4.2 Identifying and preventing vulnerabilities 1.5 Systems software	Mock Exam Preparation	1 Preparing for the Exam: Exam Technique Develop Underperforming Topics			

- **Key skills** - To develop skills and knowledge in **digital literacy** and **digital technologies** and being **responsible**, **competent**, confident and

Assessment Approach

Within our curriculum, we look at a variety of methods to assess our students. Below is the assessment plan which gives an overview of our assessment approaches with each year group.

Assessment approach	**Description**	**Year 7**	**Year 8**	**Year 9**	**Year 10**	**Year 11**	**Year 12**	**Year13**
Low stakes quizzing/ questioning and class discussion	Short answer questions from current or previous topics which is peer or self - marked, Knowledge checks beginning of each lesson							
Lesson review	Students use Reflection activities on graded lesson outcomes through DIRT task activities, allowing students to make improvements							
Teacher, peer and self-assessment of contextual research, skill-based activities and evaluation	Summative assessments are graded by either the teacher, student or peer using the assessment criteria feedback, feedback is given to allow the students to improve their work							
Whole class feedback	Whole class feedback is given to students to overcome misconceptions and provide the class the opportunities to discuss high level outcomes							
Summative end of unit and past assessments	summative assessments are used to establish current knowledge/ acquired knowledge overtime to determine gaps in knowledge							
Mock exams using past papers	Past GCSE/A-level papers and exam style questions used to test how student can apply the knowledge into the exam context.							

Cross Curricular links

Within our computing curriculum, we offer a variety of opportunities for cross curricular links, that benefits students at all levels. Our cross curricular links are as follows;

- Literacy - Students write extended answers around different topics such as Hardware, memory, ethical and cultural issues with emphasis on quality of written communication.
- Numeracy – Students use their numeracy skills to convert between different number systems (Binary/Hex). Pupils need to use basic arithmetic skills and operators within their programs and algorithms to solve computational solutions.
- PSHE – Throughout all key stages, students are exposed to numerous e-safety issues such as malware, cyber bullying, online relationships, reporting abuse etc. Equally, students are introduced to ethical, cultural and environmental issues that allow for wider discussion and debates.

Preparing for Life

At SNA, our Computing curriculum supports and further develops the following skills within students to prepare them for life beyond school and the world of work. These include:

- **Problem Solving** – Students are able to see a problem or scenario and use computational methods to find a solution through using algorithms or programs.
- **Creativity** – Pupils use their creativity skills to find different ways and forms to solve a problem through flowcharts and pseudocode.
- **Listening/Speaking** – Within lessons, students use their listening skills to decipher issues and problems which they come across. An important concept within our lessons is also listening to each other when pupils are expressing their points/findings.
- **Team work** – Collaboration is key within Computing and is something that is frequently used within lessons from team programming challenges to paired research activities. These group tasks allow students to effectively build their communication skills as well as discover their main strengths and weaknesses when working in a group.
- **Staying positive** – Within our curriculum, we emphasise on being positive and resilient as programming can throw up numerous errors/issues, which can take a lot of time to find a solution. Attention to detail is an important skill we teach students and being patient when doing this makes them even more successful.

Co-Curricular

At KS3, we offer students the chance to expand their programming skills using specialised hardware and software (Microbits, Scratch, Python). These opportunities allow students to expand their problem-solving skills while also having fun with their peers.

At KS4, we offer extra intervention sessions for students who want extra support or want to learn beyond the curriculum, which they can take into their post 16 studies.