SCARBOROUGH UTC
Technical Excellence, Employable Graduates

Long Term Plan Year 13 Computer Science

Half Term	Unit Title	Key Knowledge/Content to learn and retain	Essential Skills to acquire (subject & generic)	Link to intent and ethos	Anticipated misconceptions	Links to previous KS	Link to future KS	Opportunity for stretch and high prior attainers	SMSC & British Values	Cultural Capital	Career Link
One	2.1 Understand what is meant by computational thinking 2.1.1 Thinking abstractly 2.1.2 Thinking ahead 2.1.3 Thinking procedurally 2.1.4 Thinking logically 2.1.5 Thinking concurrently	This component will incorporate and build on the knowledge and understanding gained in the computer systems component (01). This unit focuses on what is meant by computational thinking; several thinking approaches to solve a problem to ask the question; is the problem computable? This unit is directly linked to component (03) NEA (non-exam assessment)	Drawing and labelling abstract diagrams. Extended writing Develop a line of enquiry based on observation	This unit directly relates to the intent and ethos of computer science. Learners will acquire the knowledge and skills to use computational methods to create, design, develop and test a software solution to a given problem.	The differences between abstraction and reality can have many layers. When identifying inputs this means the data/variables that are needed for the program. Procedures and functions are often mistaken as the same; a function returns a value whereas a procedure does not. Concurrent computing is different to parallel computing.	Computational methods are embedded throughout KS4 in all units; learners use these approaches when learning new content which directly links to metacognition.	Learners will have the required knowledge and skills in this unit to apply them in the workplace or to progress further in their studies at University. This unit will enable learners of the key mathematical underpinnings of computer science to aid learners in problem solving and programming.	Learners will have the opportunities to read further around the subject of computational thinking. Learners will explore computational methods through analysing practical examples.	Learners will have the opportunities to explore where computational methods are studied in other subject programmes such as Queen's University, Belfast which offers a programme of study that emphasises how the law interacts with technological innovation generating an interdisciplinary space between law and computational reasoning.	Learners will have the opportunities to explore how computational methods can be applied in all areas of education and as a life skill. Introduce learners to scholarly articles; Jeannette Wing	This unit covers all areas of careers as these skills can be applied in any workplace or at university.
Two	2..2 How computers can be used to solve problems and programs can be written to solve them. 2.2.1 Programming techniques 2.2.2 Computational methods **NEA; Content of non exam assessment programming project (component 3) 3.2 Design of the solution** 3.2.1. Decomposition of the problem 3.2.2 Describe the solution 3.2.3 Describe the approach to testing	This unit focuses on how computers can be used to solve problems and programs can be written to solve them; learners will deepen their understanding of the practical elements of the course. -programming constructs -Recursion -Global and local variables -Use of an IDE -Use of object-oriented techniques -Modularity Learners are expected to analyse, develop, evaluate and document a program written in a suitable programming language. Topic 3.2 is the second phase of the NEA; Design Learners are required to describe what the proposed solution will look like and describe how it will function.	Drawing and labelling abstract diagrams. Technical processes Extended writing Develop a line of enquiry based on observation and provide justification Evaluate benefits and drawbacks Visualisation to Problem Solving Mathematical problems Use decomposition to break the problem down into small sub-problems for computational solutions and explain the structure/ algorithms of the solution. Create a systems diagram Evidence test data Use trace tables write pseudocode Draw flow diagrams Write algorithms Design interfaces	Learners will develop knowledge and understanding which underpins the practical elements of the course through computational methods directly linked to key knowledge: -Abstraction -Decomposition -Algorithmic thinking - Data mining - Heuristics Learners will develop knowledge and understanding of technical content in this unit through computational approaches by topics that have been previously studied; Decomposition	The most important technique is the ability to break down a complex task into simple sub-tasks (decomposition) and write self-contained code in the form of functions and procedures therefore functions and procedures are introduced to learners as early as possible. Learners are required to evidence proposed data structures to form part of their intended solution. Learning data structures is similar to learning about data types. For some learners these concepts will be brand new and can only be related to computers, but some can be expressed in terms of real-life problems such as queues.	Learners should know basic programming techniques studied at KS4; concepts such as sequence, selection and iteration including subroutines (functions and procedures) which prepares learners to study recursion at KS5. Learners should know computational methods such as decomposition and abstraction.	Learners will have the required knowledge and skills in this unit to apply them in the workplace or to progress further in their studies at University. This unit prepares learners for further study in computer science courses at university such as the Programming Portfolio module , a compulsory first year module offered by the University of Hull. This unit prepares learners for further study in computer science courses at university such as the Honours Stage Project, a core third year module offered by the University of Hull.	Learners will have the opportunities to read further around the subject and to learn other programming paradigms and languages. Learners will have the opportunities to read further around the subject such as HCI (Human Computer Interaction) which provides an insight with the design and use of computer and mobile technology, focusing on the interfaces between people and systems. This is typically a degree module offered at university.	Ethical discussions on open and closed source code and the impacts this has in the industry. Ethical discussions on Ethical hacking by governments and well-known hacking groups. Ethical and social considerations are discussed surrounding software development. Laws regarding PEGI rating in the game development industry.	Learners will have the opportunity to discuss the origins of algorithms and programming languages and why so many are still in use today such as Ada, named after Ada Lovelace, which is a programming language used by the military Learners will have the opportunity to discuss the origin and history of computer interfaces and research emerging technologies.	https://www.gchq-careers.co.uk/ GCHQ Apprenticeships https://www.mi5.gov.uk/careers https://www.sis.gov.uk/explore-careers.html https://www.yhrocu.org.uk/vacancies/ https://nationalcrimeagency.gov.uk/careers/vacancies https://www.gov.uk/apply-apprenticeship University of Hull Computer Science courses University of York Top Cyber Security Universities in the UK

Half Term	Unit Title	Key Knowledge/Content to learn and retain	Essential Skills to acquire (subject & generic)	Link to intent and ethos	Anticipated misconceptions	Links to previous KS	Link to future KS	Opportunity for stretch and high prior attainers	SMSC & British Values	Cultural Capital	Career Link
Three	**NEA; Content of non exam assessment programming project (component 3) 3.3 Developing the solution** 3.3.1 Iterative development process 3.3.2 Testing to inform development	Learners are expected to analyse, develop, evaluate, and document a program written in a suitable programming language. Topic 3.3 is the third phase of the NEA; Development	Apply agile software methodology Annotate prototypes for each stage of the iterative development process Annotate testing at each stage to justify the reason for the test	Learners will develop knowledge and understanding of technical content in this unit using computational approaches through topics that have been previously studied; problem solving and programming techniques	This phase of the project will be done in an agile way, which means that each part of the problem will be solved in turn, coding a procedure, module or function, testing it, modifying it then moving onto the next part. During the process of development, you will regularly get feedback from your stakeholder, they will provide comments on how your solution is developing.	Learners should know defensive design considerations: -Anticipating misuse -Authentication -Input validation -Maintainability	This unit prepares learners for further study in computer science courses at university such as the Honours Stage Project, a core third year module offered as part of an undergraduate degree in computer science.	Learners will have opportunities to explore and analyse examples to model an iterative development process.	Learners will have the opportunity to discuss algorithms and ethics; Companies such as Facebook, Amazon, and Google use algorithms and therefore have significant power and the responsibility that goes with it. Algorithms can determine whether a stop and search can be carried out on the street and are used in decision-making in hiring and firing, healthcare, and advertising.	Learners will have the opportunity to discuss the wider implications on algorithms when creating software.	Software Engineer at GCHQ Games tester
Four	**NEA; Content of non exam assessment programming project (component 3) 3.4 Evaluating the solution** 3.4.1 Testing to inform evaluation 3.4.2 Success of the solution 3.4.3 Describe the final product 3.4.4 Maintenance and development	Learners are expected to analyse, develop, evaluate, and document a program written in a suitable programming language. Topic 3.4 is the final phase of the NEA; Testing.	Create a full test plan to include valid/invalid inputs and extreme cases.	Learners will develop knowledge and understanding of technical content in this unit using computational approaches through topics that have been previously studied; problem solving and programming techniques	This phase of the project is evidencing a full test plan to carry out a final set of acceptance testing with stakeholders and users. Test tables must show everything that needs to be tested by users of the proposed solution to be assured the system works as intended.	Learners should know defensive design considerations: -Anticipating misuse -Authentication -Input validation -Maintainability	This unit prepares learners for further study in computer science courses at university such as the Honours Stage Project, a core third year module offered as part of an undergraduate degree in computer science.	Learners will have the opportunities to read further around the subject and produce testing videos to evidence functionality of the proposed solution.	Learners will have the opportunities to discuss the ethics in relation to user experience of testing; has accessibility been considered in the proposed solution.	Learners will have opportunities to discuss project management and why testing is so important in the Alpha and Beta stages of a project.	
Five	1.5 Legal, moral, cultural, and ethical issues 1.5.1 Computing related legislation 1.5.2 Moral and ethical issues	This unit focuses on the individual moral, social, ethical, and cultural opportunities, and risks of digital technology. Legislation surrounding the use of computers and ethical issues that can or may in the future arise from the use of computers	To analyse four computing laws: - The Data Protection Act 1998 -The Computer Misuse Act 1990 -The Copyright Design and Patents Act 1988 -The Regulation of Investigatory Powers Act 2000 Extended writing; to be able to discuss the risks of digital technology in areas such as; - Computers in the workforce -Automated decision making -Artificial Intelligence	Learners will develop knowledge and understanding of digital technologies in this unit using computational approaches.	This unit requires learners to critically think in providing a constructive argument for and against in extended writing. Learners are provided with guidance on how to approach extended questions therefore should be followed.	Learners should know some legislations such as. -The data Protection Act -Computer Misuse Act	This unit prepares learners for further study in computer science courses at university such as the Honours Stage Project, a core third year module offered as part of an undergraduate degree in computer science.	Learners are provided with several case studies to support their understanding of this unit exploring issues such as. -Edward Snowden (whistle-blower of America's National Security Agency (NSA) -Cyber -attacks; Estonia suffers a cyber-attack; Sony Pictures is hacked -Google Street View which covers 99% of UK streets.	Learners will have the opportunities to discuss and debate the social, moral, ethical, and legal issues of computing such as challenges facing legislators in the digital age and driverless cars. This unit has a direct relation to British values. -Rule of law -Democracy -Respect -Tolerance -Liberty	Learners will have the opportunities to discuss and debate the challenges of the digital age such as the economic impact of the internet, the Internet of Things (IoT) and what happens to computing waste?	
Six	EXAM REVISION										