

Department: <i>ICT & Computing (Computer Science)</i>						Year Group: 9
Term	Topic/Subject	Assessment Objectives and Knowledge	Skills	Literacy, Numeracy (including wider reading)	Personal Development (SMSC, British Values, Careers, Healthy Living, Citizenship Equality and Diversity, Preparation for next stages)	AFL/Summative Assessment
Autumn 1	<p>Introduction to Python(lessons 1-7 PG resources)</p> <p>Design, Testing and IDEs</p> <p>Ethical, Legal, Cultural and Environmental Concerns</p> <p>Data Representation: Logic</p>	<ul style="list-style-type: none"> Understand how to create basic programs using Python coding. Understand purpose of CPU Understand the use of variables, constants, operators, inputs, outputs & assignments. <p>• how to investigate and discuss Computer Science technologies while considering:</p> <ul style="list-style-type: none"> ethical issues legal issues cultural issues environmental issues. privacy issues. <p>• how key stakeholders are affected by technologies</p> <p>• environmental impact of Computer Science</p> <p>• cultural implications of</p>	<ul style="list-style-type: none"> the use of variables, constants, operators, inputs, outputs and assignments the use of the three basic programming constructs used to control the flow of a program: <ul style="list-style-type: none"> sequence selection iteration (count and condition controlled loops) the use of basic string manipulation 	<ul style="list-style-type: none"> Strings Variables Data Types Selection Sequence Iteration Pseudocode Von Neuman architecture Arithmetic Logic Unit Cracking codes with Python http://inventwithpython.com/cracking/ Invent with Python http://inventwithpython.com/invent4thed/ 	<p>Ethical and Cultural Issues:</p> <ul style="list-style-type: none"> Censorship Surveillance Privacy Sharing economy Digital Divide Trolling Cyber bullying Social media <p>Environmental Issues:</p> <ul style="list-style-type: none"> Natural resources Energy E-waste Recycling <p>Computer Legislation:</p> <ul style="list-style-type: none"> Data Protection Act Freedom of Information Act Computer Misuse Act Copyright, 	<ul style="list-style-type: none"> Assessment & Feedback through topic based questions, and through practical programming activities.

		<p>Computer Science</p> <ul style="list-style-type: none"> • open source vs proprietary software • legislation relevant to Computer Science: <ul style="list-style-type: none"> ○ The Data Protection Act 1998 ○ Computer Misuse Act 1990 ○ Copyright Designs and Patents Act 1988 ○ Creative Commons Licensing ○ Freedom of Information Act 2000. 			<p>Designs and Patents Act</p> <ul style="list-style-type: none"> • Creative Commons licenses 	
Autumn 2	<p>Python Programming (lessons 7-16 PG resources).</p> <p>Computational Logic</p> <p>Data Representation: Binary, Hex, ASCII, Unicode, images, sound and compression.</p>	<ul style="list-style-type: none"> • Units <ul style="list-style-type: none"> ○ bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte ○ how data needs to be converted into a binary format to be processed by a computer. • Numbers <ul style="list-style-type: none"> ○ how to convert positive denary whole numbers (0–255) into 8 bit binary numbers and vice versa ○ how to add two 8 bit binary integers and explain overflow errors 	<ul style="list-style-type: none"> • how to understand and use basic string manipulation • how to understand and use basic file handling operations: <ul style="list-style-type: none"> • open • read • write • close • how to define and use arrays (or equivalent) as appropriate when solving problems • how to understand and use functions/sub programs to create structured code. • how to analyse and identify the requirements for a solution to the problem • how to set clear objectives that show an awareness of the need for real world utility 	<ul style="list-style-type: none"> • Strings • Variables • Data Types • Selection • Sequence • Iteration • Pseudocode • While Loops • Arrays • Functions • Abstraction • Validation • Cracking codes with Python • http://inventwithpython.com/cracking/ • 	<p>Computer Science careers:</p> <ul style="list-style-type: none"> • Programming • Science • Engineering • Finance • Medical • Computer Modelling <p>Academia:</p> <ul style="list-style-type: none"> • FE • University • College • Apprentices 	<ul style="list-style-type: none"> • Assessment & Feedback through topic based questions, and through practical programming activities

		<ul style="list-style-type: none"> ○ which may occur ○ binary shifts ○ how to convert positive denary whole numbers (0–255) into 2 digit hexadecimal numbers and vice versa ○ versa ○ how to convert from binary to hexadecimal equivalents and vice versa ○ check digits. ● Characters <ul style="list-style-type: none"> ○ the use of binary codes to represent characters ○ the term ‘character-set’ ○ the relationship between the number of bits per character in a character set and the number of characters which can be represented (for example ASCII, extended ASCII and Unicode). 	<ul style="list-style-type: none"> ● how to use abstraction and decomposition to design the solution to a problem ● how to identify the data requirements for their system ● how to identify test procedures to be used during and after development to check their system against the success criteria ● how to use validation to ensure a robust solution to a problem. 	<ul style="list-style-type: none"> ● Invent with Python ● http://inventwithpython.com/invent4thed/ 		
Spring 1	<p>Python Programming</p> <p>Translators and faculties of languages</p>	<ul style="list-style-type: none"> ● characteristics and purpose of different levels of programming language, including low level languages ● the purpose of translators ● the characteristics of an assembler, a compiler and an interpreter 	<ul style="list-style-type: none"> ● how to design suitable algorithms to represent the solution to a problem ● how to design suitable input and output formats and navigation methods for their system ● how to identify suitable 	<ul style="list-style-type: none"> ● Data Capacity ● Characteristics ● Algorithms ● Variables ● Validation ● Making games with Python & Pygame 		<ul style="list-style-type: none"> ● Feedback through topic based questions, and through practical programming activities.

		<ul style="list-style-type: none"> common tools and facilities available in an integrated development environment (IDE): <ul style="list-style-type: none"> editors error diagnostics run-time environment translators. 	<p>variables and structures with appropriate validation for their system</p> <ul style="list-style-type: none"> how to use appropriate data types in their system how to use functions/sub programmes to produce structured reusable code how to select suitable techniques for the development of the solution. 	<ul style="list-style-type: none"> http://inventwithpython.com/pygame/ 		
Spring 2	Data Representation	<ul style="list-style-type: none"> Numbers <ul style="list-style-type: none"> how to convert positive denary whole numbers (0–255) into 8 bit binary numbers and vice versa how to add two 8 bit binary integers and explain overflow errors which may occur binary shifts how to convert positive denary whole numbers (0–255) into 2 digit hexadecimal numbers and vice versa how to convert from binary to hexadecimal equivalents and vice versa check digits. Characters <ul style="list-style-type: none"> the use of binary 	<ul style="list-style-type: none"> how to develop a solution to the identified problem using a suitable programming language(s) how to demonstrate testing and refinement of the code during development how to explain the solution using suitable annotation and evidence of development how to use suitable techniques to solve all aspects of the problem how to take a systematic approach to problem solving how to deploy practical techniques in an efficient and logical manner how to show an understanding of the relevant information by presenting evidence of the development of their solutions how to show an understanding of the technical terminology/concepts that arise from their investigation through analysis of the data collected how to use the 	<ul style="list-style-type: none"> how to investigate and discuss Computer Science technologies while considering: <ul style="list-style-type: none"> ethical issues legal issues cultural issues environmental issues. privacy issues. Making games with Python & Pygame http://inventwithpython.com/pygame/ 		<ul style="list-style-type: none"> Assessment & Feedback through topic based questions, and through practical programming activities.

Summer 1	System Security Programming project	<ul style="list-style-type: none"> • Programming project (practice) 	<ul style="list-style-type: none"> • how to produce a full report covering all aspects of the investigation • how to present the information in a clear form which is understandable by a third party and which is easily navigatable • how to critically appraise the evidence that they have presented • how to test their own solution • how to present their evaluation in a relevant, clear, organised, structured and coherent format • how to use specialist terms correctly and appropriately • how to present a conclusion to the report • how to justify their conclusions based on the evidence provided. 		<p>Making games with Python & Pygame</p> <p>http://inventwithpython.com/pygame/</p>	<ul style="list-style-type: none"> • Feedback practical programming activities.
Summer 2	Programming project	<ul style="list-style-type: none"> • Programming project (practice) 				Feedback practical programming activities.