



# YEAR 12 A LEVEL COMPUTER SCIENCE SPRING TERM 2 – PAPER 1

'An ambitious curriculum that meets the needs of all'

## Medium Term Planning – Software Development

Curriculum Intent	Pupils will be taught the following National Curriculum guidelines this term:
<b>Skills/Assessment Objective Links</b>	<p><b>At the end of this Unit all students should be able to:</b></p> <ul style="list-style-type: none"> <li>list the stages in the waterfall lifecycle model</li> <li>name two other systems development models</li> <li>name and describe different types of testing</li> <li>write a pseudocode algorithm to solve a simple problem</li> <li>use a trace table to trace through an algorithm</li> <li>interpret simple algorithms to describe their purpose</li> <li>list two features of a good algorithm</li> <li>Define the term “programming paradigm” and give an example of two paradigms</li> <li>define the terms object, class, method, attribute, inheritance</li> <li>draw a simple inheritance diagram for a set of classes in an object-oriented approach</li> <li>follow through a simple program using the LMC instruction set</li> </ul> <p><b>Most students will be able to:</b></p> <ul style="list-style-type: none"> <li>briefly describe agile methodologies, extreme programming, the spiral model and rapid application development</li> <li>write pseudocode algorithms to solve problems</li> <li>describe different programming paradigms, including procedural, and object oriented paradigms</li> <li>explain the terms encapsulation and polymorphism</li> <li>write simple assembly code programs using the LMC instruction set</li> <li>distinguish between immediate, direct and indirect addressing modes in assembly language</li> </ul> <p><b>Some students will be able to:</b></p> <ul style="list-style-type: none"> <li>describe the relative merits and drawbacks of different software development methodologies and when they might be used</li> <li>design algorithms to solve complex problems</li> <li>explain why different programming paradigms are suited to different applications and the advantages of each</li> <li>describe and use four methods of addressing memory: immediate, direct, indirect and indexed</li> </ul>
<b>Numeracy</b>	Time slice
<b>Literacy</b>	<p><b>Vocabulary Tier 3:</b> black box testing, functional testing, white box testing, structural testing, alpha testing, beta testing, waterfall lifecycle model, spiral model, agile modelling, extreme programming, rapid application development, algorithm, trace table, programming paradigm, procedural language, functional programming language, declarative language, logic programming, object-oriented paradigm, class, object, method, attribute, inheritance, encapsulation, polymorphism, Little Man Computer (LMC), assembly language, machine code instruction, addressing modes (immediate, direct, indirect and indexed), accumulator</p> <p><b>Vocabulary Tier 2:</b> analysis, design, implementation, tests, extreme</p> <p><b>Reading:</b> Worksheets, presentations, answer sheets, exam questions, mark scheme, further reading for homework</p> <p><b>Writing:</b> Answer on the worksheet via word</p> <p><b>Oracy:</b> listening and using tier 3 words</p>
<b>Becoming future ready</b>	<p><b>Careers/Employability:</b></p> <p>Understand the grade requirements at universities and the topics that can be applied for. Explore apprenticeship opportunities with a range of industries.</p> <ul style="list-style-type: none"> <li>Software Architect.</li> </ul>

	<ul style="list-style-type: none"> <li>▪ Data Scientist.</li> <li>▪ Machine Learning Engineer.</li> <li>▪ Blockchain Developer</li> <li>▪ Cybersecurity Engineer.</li> <li>▪ Cloud Solutions Architect.</li> <li>▪ AI Research Scientist.</li> <li>▪ Full-Stack Developer.</li> </ul>
<b>Adaptation</b>	Throughout this topic, quality first teaching will provide differentiation:
<b>QFT/SEND Provision</b>	<p><b>By product:</b> Learners are asked to present outcomes in a different way via pieces of writing, targeted questioning, models and drawings and speaking.</p> <p><b>By resource:</b> Worksheets are well presented and accessible. Instructions are clearly outlined and separate from the information so that pupils know where to begin and end. Handouts are differentiated by outcome. Resources used will appeal to the range of preferred learning styles of pupils e.g. visual, auditory or kinesthetic learners. Scaffolding of tasks – word frames.</p> <p><b>By Intervention:</b> By providing different levels of supervision and support</p> <p><b>By Progressive Questioning:</b> Exploring pupils’ understanding through interactive dialogue using Blooms Taxonomy.</p> <p><b>By Grouping:</b> According to prior attainment, gender, social preference, preferred learning style.</p> <p><b>By Task:</b> Pupils identify targets which are meaningful to them via feedback sheets</p> <p><b>By Offering Optional Activities:</b> In class or as homework, to extend learning.</p> <p>This QFT/SEND provision will be explicit within the lesson by lesson schemes of work.</p>
<b>Implementation Curriculum Delivery</b>	To be able to:
<b>Learning Outcomes (Knowledge)</b>	<p><b>Topic 1 Systems analysis methods</b> Describe the waterfall lifecycle model, agile methodologies, extreme programming, the spiral model and rapid application development Describe the relative merits and drawbacks of different methodologies and when they might be used</p> <p><b>Topic 2 Writing and following algorithms</b> To understand the term ‘algorithm’ To learn how to write and interpret algorithms using pseudocode</p> <p><b>Topic 3 Programming paradigms</b> Understand the need for and characteristics of a variety of programming paradigms Describe the features of procedural languages Describe the features of declarative languages Describe the features of object-oriented languages Develop an understanding of classes, objects, methods, attributes, inheritance, encapsulation and polymorphism</p> <p><b>Topic 4 Assembly language</b> Write and follow simple assembly language programs Understand and apply immediate, direct, indirect and indexed addressing modes</p> <p>End of unit assessment</p>
<b>Current learning to be developed in the future within:</b>	links with programming Object Orientated Programming in paper 2.
<b>Assessment</b>	See assessment maps for formative and summative assessment opportunities.
<b>Impact</b>	<p>Review assessment results and target pupils that require further support via:-</p> <ul style="list-style-type: none"> <li>• Learning conversation</li> <li>• Changing seating plan</li> <li>• Plan lessons to address areas of concern in assessment</li> </ul>

- Targeted homework based on low performance areas identified in the assessment marked pieces
- Stretch and challenge high ability pupils by identifying ambitious next steps to extend knowledge



Create a feedback sheet for each student

Each student identifies areas of Green, Amber and Red using Mark Assessment on their feedback sheet

Complete NOW task on areas identified as Amber and Red