

Module Title	Software Engineering: Creating Quality Products		Date of Approval		Mar-17
Module Code	6CC550	Module Level	6	Credit value	20
Module Delivery Mode	Online x	Blended/Face to Face		Work-Based Learning	
Module Description	<p>This module provides a wide-ranging and highly practical introduction to the Software Lifecycle – from software specification and design through to programming, testing and documentation. Basic programming techniques are introduced at an early stage and so previous programming experience is not assumed. However, those who have programming skills will gain the opportunity to extend their understanding of software development as an engineering process and to apply this knowledge in the implementation of a larger software development task.</p> <p>Practical programming is taught within a framework of software engineering techniques thereby allowing you to better appreciate that the ability to ‘cut code’ represents only one (albeit crucial) part of the software Life Cycle. Learners are introduced to a range of highly transferable skills which are needed in order to produce fully-documented high quality software products.</p> <p>As part of the assessment for this module, you will design, develop, test and document several programmes. You may undertake this work individually or form a group (software development team). Groups normally comprise three members. Group work is strongly encouraged – but is not a requirement.</p> <p>You are expected to make regular contributions to the virtual classroom’s Discussion Forum.</p>				
Module Learning Outcomes	<p>On successful completion of the module, you will be able to:</p> <ol style="list-style-type: none"> 1. Demonstrate competence in basic algorithm design, program coding, documentation development, and debugging. 2. Apply software engineering techniques across key areas of the Software Lifecycle. 3. Critically appraise software development processes with particular reference to software specification, documentation and testing. 				
Module Content	<p>Indicative topics:</p> <ul style="list-style-type: none"> • Origins and evolution of software engineering techniques • Examples of good and bad practice, software quality • Software project management • Human factors 				

	<ul style="list-style-type: none"> • Introduction to programming • Algorithm design • Software Lifecycle • Sequential development strategy – strengths and weaknesses • Agile development • Software specification • Programming strategies • Development environments • Testing strategies and techniques • Exhaustive testing • Documentation techniques. 	
Module Learning and Teaching	Scheduled Learning and Teaching Activities	25%
	Guided Independent Study	75%
	Placement Learning	0%
Module Assessment	<p>Component 1: COURSEWORK 1: <i>Software Engineering and Programming Basics</i></p> <p>Summary of Assessment Method: Online, self-paced and supported, this includes the following:</p> <ul style="list-style-type: none"> • Regular and considered contributions to the Discussion Forum • Written responses to set questions/activities • Development of introductory programmes. <p>You will be asked to undertake several programming activities and demonstrate your understanding of a number of key Software Engineering concepts.</p> <p>Weighting: 40 %</p> <p>Assesses Learning Outcomes: 1,3</p> <p>Component 2: COURSEWORK 2: <i>Larger Program Development Exercise</i></p> <p>Summary of Assessment Method: Online, self-paced and supported, this will include the following:</p> <ul style="list-style-type: none"> • The use of software engineering techniques so as to create a high quality 'software product' which meets the objectives of a given requirement specification. • The demonstration of effective design, development, documentation and testing strategies. 	

	<p>You will undertake a software development exercise with the aim of meeting the requirements of a given specification. The final 'product' should be accompanied with appropriate documentation so as to demonstrate all aspects of its design and testing within a structured Software Engineering framework. You should also submit a video presentation overviewing your work.</p> <p>You are expected to proactively and professionally liaise with the Academic Associate (Tutor) to discuss progress.</p> <p>You may request to be allowed to undertake this work as a group activity (a group usually comprises up to three students). It is important to note that the work undertaken by each team member will be marked separately. Should you wish to form such a 'development team', you will be able to make contact with others via the discussion forum.</p> <p>Detailed information concerning the requirements for individual activity and team-work will be provided (including the length and structure of the report and duration of the video).</p> <p>Weighting: 60%</p> <p>Assesses Learning Outcomes: 1,2,3</p>
<p>Reading List</p>	<p>Required Reading</p> <p>Leach, R. (2016). <i>Introduction to Software Engineering</i>. 2nd ed. CRC Press.</p> <p>Rajlich, V. (2011). <i>Software Engineering: The Current Practice</i>. CRC Press.</p> <p>Indicative Recommended Reading</p> <p>Pressman, R. and Maxim, B. (2015). <i>Software Engineering: A Practitioner's Approach</i>. 8th ed. McGraw-Hill.</p> <p>Herman, T. (2013). <i>A Functional Start to Computing with Python</i>. CRC Press.</p>