Module Title	**Computer Architecture and Operating Systems**
Course Title	BEng (Hons) Electronic and Computer Systems Engineering
School	☐ ASC ☐ ACI ☐ BEA ☐ BUS ☒ ENG ☐ HSC ☐ LSS
Division	Electrical and Electronic Engineering
Parent Course **(if applicable)**	
Level	5
Module Code *(showing level)*	EEE_5_CAO
JACS Code (completed by the QA)	
Credit Value	20 credit points
Student Study Hours	Contact hours: 52 Student managed learning hours: 148 Placement hours: 0
Pre-requisite Learning	Familiarity with basic data structures, hardware and software of a typical computer system. Fundamental skills of Computer Programming and knowledge of basic algebra.
Co-requisites	None
Excluded combinations	None
Module co-ordinator	
Short Description **(max. 100 words)**	This module consists of two parts. The first part, Computer Architecture, provides the basic engineering foundations of detailed design and implementation of a digital computer system; designs for the CPU, I/O subsystems, and memory organizations; ALU design and computer arithmetic. Bus, memory organization and interrupt structures, control function implementation, pipelining, performance measurement and further Computer Organisation topics. The second part, Operating Systems, is designed to teach students about the role that the operating system has in computing. It also considers the applications in systems such as distributed, multimedia systems and the role that the OS has in supporting the functioning of these. Students will view the operating system from several vantage points.
Aims	The module focuses on Computer Organization, Computer architecture and on Systems Resource management - Operating Systems. General aims are to provide an understanding of the

	operation of computing devices and their operating systems including their role in resource management and systems operation. Students will become familiar with the computer architectures and operating systems as well as the challenges facing engineers in providing operating systems support. This module - for code examples - uses predominantly C+, with some Java and additional programming in Unix/Linux using a scripting programming language such as Python/Perl. Students can still understand operating systems algorithms without a thorough knowledge of these languages. Finally, it will be fully covered the computer architecture, organization, and its role in computer engineering as well as the standards and design tools used in computer architecture and organization.
Learning Outcomes **(4 to 6 outcomes)**	Knowledge and Understanding: • Knowledge of the underlying principles and practices to be able to utilise mathematical/computer methods to solve and simulate operating systems algorithms and practices to build a computer system and measure its performance. Apply computer architecture methodology (A1, A2, A3). Intellectual Skills: • Develop and manage system resources and perform resource management in operating system environments. Discuss and examine the need of operating systems protection and security. Analyse computer organisation and architectures strategies (B1, B2, B3). Practical Skills: • Ability to design, implement, manage and test operating systems and computing devices/components. Breakdown and apply UNIX/Linux and Windows-type operating systems which are massive and complex containing millions of lines of source code (C1, C2, C8). • Master design rules that dictate how the components can be constructed and how they can be interconnected to form systems by understanding that an operating system is large and complex, and it must be created piece by piece and each of these pieces should be a well-delineated portion of the system, with carefully defined inputs, outputs, and functions (C3, C4, C6). Transferable Skills: • Design and implement UNIX/Linux scripts using fundamental concepts of shell scripting (programming). Design and construct various computer system circuits using fundamental building blocks (D2, D3). • Knowledge and understanding of the commercial, economic and social context of tradeoffs associated with data path, control unit, performance enhancements in computer processor

	organization, resource management in different operating environments. (D4, D5, D6)
Employability	This module is suitable for students who intend to work in operating systems and computer architecture enterprises as operating systems engineers, computer systems engineer, analysts, programmers, managers or developers. The computer architecture part of the module aims to develop your skills on most important topics that form the basis for a complete knowledge of a computer system and its operation, including computer organisation giving you experience in the implementation of computer architecture and systems technologies. This module aims to develop your operating systems and computer architecture skills in demand in both industry and research. You will attain the technical expertise and knowledge to take a good idea from conception through to a viable OS design, as well as the OS maintenance and administration, a key characteristic for many employers.
Teaching and learning pattern	Contact hours includes the following: (please click on the checkboxes as appropriate) √ Lectures ☐ Group Work: ☐ Seminars ☒ Tutorial: ☒ Laboratory ☒ Workshops ☒ Practical ☒ VLE Activities
Indicative content	Part 1: Computer Architecture Introduction to Computer Engineering (Overview of Computer Systems) Fundamental concepts of Computer Organization and Architecture Cooperation of hardware and software including sophisticated selection of algorithms aiming to improve computer system performance. Number representation and arithmetic algorithms for manipulating numbers in various number systems. Number Systems, Computer Arithmetic, Digital Logic, The Control Unit, The Central Processing Unit (CPU), Computer Functions and Interconnections. Internal and External Memory, Cache Memory. Input Output (I/O) Instruction set architectures, including machine and assembly level representations and assembly language programming. Part 2: Operating systems A clear description of the concepts that underlie operating systems. Operating systems structures. Process management (CPU Scheduling, Threads, deadlocks) Concurrent processing Memory Management (Main and Virtual memory, Paging) Storage management (File systems, I/O systems) Protection and security of operating systems A brief introduction to Virtualization and the Cloud Operating systems design techniques and methodologies Distributed operating systems Multimedia operating systems Modern operating system case studies (Windows, Unix, Linux, Android)

Assessment method (Please give details – of components, weightings, sequence of components, final component)	Formative assessment: Verbal feedback in Tutorial and weekly workshops (lab exercises). Quizzes as part of the lab exercises (Logbook). Summative assessment: **Component 1 (CW1_40) – Coursework (CW) 40%** comprises two subcomponents (SC1_60 and SC2_40): • **SC1_60**: Logbook (Lab Exercises - 60% of the CW) (C1, C2, C6, C8, D3, D6) • **SC2_40**: Assignment (Formal Report - 40% of the CW) (C3, C4, D2, D3, D4, D5) **Component 2 (EX1_60) – Written examination 60%** • Exam: 2-hour unseen written exam (60%) (A1, A2, A3, B1, B2, B3). Workshop logbook detailing the practical work constitutes 60% of the coursework. A laboratory manual will be provided for each student containing details of laboratory organisation and procedures and the instructions for the individual exercises/assignments/mini projects. The practical workshop includes use of modern operating systems, design and implementation of algorithms as well as computer architecture principles. Assignment (formal report) will be in the region of 2000 words. This component will constitute 40% of the coursework. Students will have to write a technical report in the area of operating systems and computer architectures. Formal report mark scheme and grade descriptors will be provided. To be awarded a pass in the module a student must: (a) achieve an overall weighted average mark for the module of at least 40% and (b) achieve the minimum threshold mark (30%) in each of the two components.
Mode of resit assessment (if applicable)	Summative assessment: Standard mode of referral 2-hour unseen written exam (60%) – Coursework as described in the previous Assessment method section.

Indicative Sources ***(Reading lists)***	Core materials: 1. **Operating Systems Internals and Design Principles**, 9th edition (2018), William Stallings, Prentice Hall, Print ISBN: 9780134670959, 0134670957, eText ISBN: 9780134700113, 0134700112 **2. Operating System Concepts**, 10th edition (2018), Abraham Silberschatz; Greg Gagne; Peter B. Galvin, John Wiley & Sons, Print ISBN: 9781119329480, 1119329485, eText ISBN: 9781119320913, 1119320917 **3. Computer Organization and Architecture** Designing for Performance,11th edition (2019), William Stallings, Pearson, Print ISBN: 9780134997193, 0134997190, eText ISBN: 9780135160930, 0135160936 4. **Structured Computer Organization**, 6th edition (2013), Andrew S. Tanenbaum, Pearson, Prentice Hall, ISBN 10: 0-13-291652-5, ISBN 13: 978-0-13-291652-3 5. **UNIX the Textbook**, 3rd Edition (2017), Syed Mansoor Sarwar, Robert Koretsky, Chapman & Hall, Print ISBN: 9781482233582, 1482233584, eText ISBN: 9781482233599, 1482233592 Optional reading: 1. Modern Operating Systems: Global Edition, 4th edition (2015), Andrew S Tanenbaum; Herbert Bos, Pearson Education Limited, ISBN-10 1292061960, ISBN-13 9781292061962
Other Learning Resources	VLE (Moodle) site for this module: it contains weekly lecture notes, workshop laboratory exercises, tutorial problems and additional support teaching and learning material.