Module M1 (Mandatory)	**Computational Methods**
Module code and mode of delivery	Code: EEP55C22 Delivery: Hybrid Learning through Blackboard VLE/LMS and F2F as appropriate (see below).
Module ECTS Weighting	10
Semester of delivery	S1 + S2
Module Contact Hours	44 hours lectures (hybrid synchronous online and in-situ F2F), 96 hours independent student learning, 10 hours tutorials (in-situ F2F as appropriate) 100 hours continuous assessment in the form of student assignments which require the design of mini-computational algorithms.
Module Coordinator	New AP
Module teaching staff and academic titles	Prof. Anil Kokaram, New AP
Module description—content	This module gives students hands on instruction and guidance in the implementation and deployment of algorithms in real computational applications. Students build working system blocks from the ground up in applications selected from speech and audio signal processing, image and video processing, biomedical systems and geophysical modeling. Git is used for version control and the core development uses Matlab, NumPy, SciPy, PyTorch and MatplotLib. The module introduces best practice engineering coding methodology including the serious challenge of building tests for working infrastructure. A key element is the deployment of the latest technologies at scale including the tools for accelerating development cycles using Deep Learning, Signal processing and numerical optimisation libraries. The Mathematics which underpins this module is refreshed alongside the pre and co-requisites. This module therefore provides a foundation for building engineering applications out of pure research.
Module learning aims/objectives	The key objectives are as follows: • To enable students to design and build engineering solutions by integrating and implementing numerical solutions in an appropriate platform. • To effectively deploy modern software programming practice in the design of computational methods • To visualise results and design experiments in a computational framework
Module learning outcomes	On successful completion of this module, students should be able to:

	LO1. Express mathematical engineering solutions in terms of an algorithmic implementation LO2. Design efficient computational frameworks in Matlab and Python adapted to the domain of the data used e.g. video or audio or biomedical. LO3. Deploy standard libraries in Matlab and Python to solve problems in system modeling, machine learning and signal processing. LO4. Design and implement tests for computational algorithms. LO5. Use Git for version control and software management. LO6. Quantify and manipulate resource trade-offs concerning speed, memory, energy consumption and power dissipation in the design of computational implementations. LO7. Adapt a computational implementation for large scale deployment and deploy high level scripting for large scale experimentation. LO8. Assess the suitability of algorithms for deployment in cloud-based systems or "on premise". LO9. Communicate the key aspects of a computational implementation succinctly and correctly.
Module assessment, separate components and their weighting to be mapped into SITS	This module is assessed through 100 % Continuous Assessment (A single SITS component). There will be one assignment to be assessed every 6 weeks in the two semesters (hence 2 assignments per semester for 2 semesters in total). The last assignment consists of a 4-page technical summary of a computational system implementation and an associated presentation and demonstration by the student to the class.