

Module Title	Advanced Programming
Level	5
Reference No.	CSI_5_ADP
Credits	20
Student Study Hours	Total: 200 Contact hours: 52 Student managed learning hours: 148
Pre-Requisites	Software Development 3: Object-oriented Programming
Co-requisites	None
Excluded combinations	None
Module coordinator	TBC
Division	Division of Computer Science and Informatics
Short Description	This module explores the development of concurrent and parallel applications and the need for multi-threaded programming in GUI development. The design of effective GUIs which launch long-running background tasks on dedicated threads of execution is covered. The theoretical relationship and distinction between concurrent programming and parallel processing to accomplish a task faster is also investigated. Network connections and distributed models of development are investigated.
Aims	This module aims to provide students with the ability to develop multi-threaded applications in event-driven (GUI), collaborative and parallel scenarios. Students will learn both how individual local applications use concurrency either necessarily or in order to boost performance through parallelism, and how distributed applications can collaborate in a related but much larger way. The aim is to give students an overview of how practical applications must be built. In many cases software development examples are simplified to focus on individual concepts and this often means disregarding the need for multi-threading and collaboration between processes; this module seeks to elaborate this bigger picture as well as introducing students to the practicalities of concurrent programming.
Learning Outcomes	<p>LO1: Knowledge and Understanding</p> <ul style="list-style-type: none"> • Demonstrate an understanding of software development using a variety of software engineering techniques. • Critically evaluate the quality of a software artefact. (Maps to: BCS 2.2.1 a1-a6, a9) <p>LO2: Intellectual Skill</p> <ul style="list-style-type: none"> • Use sources of information to improve your knowledge and understanding. (Maps to: BCS 2.2.1 a7-a9) <p>LO3: Practical Skills</p> <ul style="list-style-type: none"> • Develop multi-threaded applications. (Maps to: BCS 2.2.1 b1-b4) <p>LO4: Transferable Skills</p> <ul style="list-style-type: none"> • Effectively plan the development of a practical project from design to implementation. (Maps to: BCS 2.2.1 c1, c2)
Employability	GUI-based applications are the primary form of software used directly by individuals, and accordingly represent a large part of the software development industry's product range. The ability to create GUI based applications is therefore essential for a developer, while the ability to critique GUI design is a key aspect of assessing software quality and suitability in other employment roles. Multi-threaded programming is of increasing importance and also represents an essential skill for developers.

Teaching and Learning Pattern	<p>This module emphasises learning through practical exercises and the development of actual software artefacts. Short lectures will be used to inform the laboratory activities and provide a forum for discussion of issues students have encountered in the practical work. The lab sessions will occupy the majority of the contact time and will involve much independent working. Students are required to keep a clear record of the work they have done and are encouraged to experiment and investigate beyond the basic material being taught.</p>
Indicative Content	<ul style="list-style-type: none"> • Concurrency • Threads, co-ordination and interference • Synchronisation, Atomic access, Memory Consistency • Threading in GUI applications • Event-driven programming • Parallelism • Network programming • Client/Server paradigm • Distributed processing
Assessment Elements & weightings	<p>EXAM 40% : COURSEWORK 60%</p> <p>Summative Assessment</p> <p>Exam: The examination will be a 2-hour closed book unseen paper. (LO1, LO2, LO3)</p> <p>Coursework: a multi-stage software development project and its documentation to be carried out through the majority of the teaching weeks. (LO3, LO4)</p> <p>Formative Assessment</p> <p>Skills for the summative assessment will be embedded throughout formative opportunities in Lectures and Workshops. Formative assessment will take different forms, such as:</p> <ul style="list-style-type: none"> • interactive revision quizzes • verbal feedback on tutorial activities • observation and questioning to provide instant feedback as the student takes part in learning activities
Indicative Sources (Reading lists)	<p>Core:</p> <ul style="list-style-type: none"> • Liang, Y. (2019) <i>Introduction to Java Programming and Data Structures, Comprehensive Version</i>. Global Edition, 11th Edition • Baesens, B. et al (2015) <i>Beginning Java Programming: The Object Oriented Approach</i>. Wrox. ISBN 1118739493 • Weisfeld, M. (2013) <i>The Object-Oriented Thought Process (Developer's Library)</i>. Addison Wesley. ISBN 0321861272