

Module Title	Object-Oriented Programming
Level	5
Reference No.	CSI_5_OOP
Credits	20
Student Study Hours	Total: 200 Contact hours: 52 Student managed learning hours: 148
Pre-Requisites	Software Development 2: Data structures and algorithms.
Co-requisites	None
Excluded combinations	None
Module coordinator	TBC
Division	Division of Computer Science and Informatics
Short Description	This module introduces the more sophisticated programming constructs associated with object-oriented programming, the dominant programming paradigm in current use. It also introduces event driven programming and how it relates to the development of graphical user interfaces. This module explores practical application development using graphical user interfaces and will cover the use of classes and objects of a kind found in many real-world applications. The use of professional development tools gives a rounded picture of the development process.
Learning Outcomes	LO1: Knowledge and Understanding - Select appropriate design notations, software development environments and programming languages. LO2: Intellectual Skills - Read and understand object-oriented design documentation. - Make effective use of technical documentation. (Maps to: BCS 2.2.1 a1-a5, a7-a9; 2.2.3 a1-a3) LO3: Practical Skills - Develop programs using the object-oriented approach. (Maps to: BCS 2.2.1 b1-b4) LO4: Transferable Skills - Effectively plan the development of a practical project from design to implementation. (Maps to: BCS 2.2.1 c1-c2)
Aims	To provide an understanding of object-oriented programming and the basics of graphical user interface development.
Employability	This module delivers an increasingly sought-after set of skills that are greatly valued in the job market. There are a number of organisations such as Google campaigning for Programming and Software Development to be taught more widely. They need these skills. Programmers who can re-use components rather than re-write them are much more productive. The skills developed here are applicable to many different languages and development platforms and are commonly sought after. All Information Technology professionals need to understand the process of software development even if they do not directly develop software.

Teaching and Learning Pattern	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.
Indicative Content	• Classes and objects • Subtyping, overloading, inheritance and type compatibility • Generic typing • Object-oriented design and UML notation • Closures • Basic GUI Development • Event-driven programming
Assessment Elements & weightings	COURSEWORK 100% Summative Assessment Coursework: Expected to consist of an individually assessed practical development assignment involving design and development of an application program to a given specification. Expected to represent basic functionality for the application together with design documentation. Students are also likely to complete an application and associated documentation. (LO1 - LO4) (Maps to: BCS 2.2.1 a1-a5, a7-a9; b1-b4; c1-c2; 2.2.3 a1-a3) Formative Assessment Skills for the summative assessment will be embedded throughout formative opportunities in Lectures and Workshops. Formative assessment will take different forms, such as: • verbal feedback on tutorial activities • observation and questioning to provide instant feedback as the student takes part in learning activities
Indicative Sources (Reading lists)	Core: There is no core textbook defined for this module. Students are expected to refer to the indicative sources below: Optional: • 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