Module Descriptor

Title	Programming with Objects		
Session	2025/26	**Status**	Published
Code	COMP07070	**SCQF Level**	7
Credit Points	30	**ECTS (European Credit Transfer Scheme)**	15
School	**Computing, Engineering and Physical Sciences**		
Module Co-ordinator	Dr. Thomas Hainey		

Summary of Module

his is an entry level programming module requiring no prior programming experience aimed at providing students with fundamental programming knowledge in C# and C++ with a view to developing in Unreal and Unity.

The module will introduce object oriented programming (OOP) in C# and C++ emphasising computational thinking and problem solving to provide a foundation for software development and games development required by students for their programmes of study. Many practical programming exercises will be used either in the laboratory or remotely to develop experience in using an Integrated Development Environment (MS Visual Studio 19/22) to develop programming skills.

The module introduces the 4 pillars of OOP: encapsulation, abstraction, polymorphism and inheritance. Essential basic programming concepts which are the building blocks will also be covered including: variables, logical and arithmetic operators, control flow, classes including data members and member functions, parameter passing, scope, static and dynamic data structures. The module will also discuss designing and implementation of simple graphical user interfaces. The more advanced concepts of inheritance and polymorphism are introduced but will be covered in more depth in the follow up module - Software Development for Games.

The module will be student centred by providing remote demonstrations combined with on campus support. Class sessions will be hybrid and flexible - recorded, in a written format, directly posted on to the VLE (AULA) and live Teams/Zoom and Discord meetings depending on what students demand. Assessment will be an authentic implementation of a game inventory.

- This module embeds the key "I am UWS" graduate attributes and in particular: Work Ready(Digitally Literate, Problem-solver, Creative, Imaginary, Resilient), Successful(Autonomous, Innovative)

- The module will be student cantered and delivered in a number of formats: online, remotely, optional on-campus and live code demonstrations will be provided with support included to bolster student knowledge with authentic programming exercises.

- The module will be simple and coherent as it will be a rudimentary pre-requisite for a number of modules building on the fundamentals presented including Software Development for Games and Game Engine 1 and 2. This will also be beneficial for other modules including Programming in AI and Games Console Development.

- In terms of inclusivity the module will recognize the diverse cohort that the Programmes attract i.e. students that are more interested and have an innate interest in

programming and students that are more interested in design. Additional complex programming examples will be provided to students who wish to explore programming to greater depth.

Module Delivery Method	**On-Campus**[1] ☒	**Hybrid**[2] ☐	**Online**[3] ☐	**Work -Based Learning**[4] ☐
Campuses for Module Delivery	☐ Ayr ☐ Dumfries	☐ Lanarkshire ☐ London ☒ Paisley	☐ Online / Distance Learning ☐ Other (specify)	

Terms for Module Delivery	Term 1 ☐	Term 2 ☐	Term 3 ☐
Long-thin Delivery over more than one Term	Term 1 – Term 2 ☒	Term 2 – Term 3 ☐	Term 3 – Term 1 ☐

Learning Outcomes	
L1	Understand basic programming fundamentals including object orientation in C# and C++
L2	Demonstrate use of standard programming constructs for selection, iteration and data structures and collections in C# and C++
L3	Create a simple graphical user interface for a program and use this to create interactive software in C# and C++
L4	Demonstrate use of object based programming in creating an application in C# and C++
L5	

Employability Skills and Personal Development Planning (PDP) Skills	
SCQF Headings	**During completion of this module, there will be an opportunity to achieve core skills in:**
Knowledge and Understanding (K and U)	**SCQF 7** A broad knowledge of the following programming concepts and principles: Code and data in programs, simple data types and operators; simple expressions and statements; classes and collections; constructs

[1] Where contact hours are synchronous/ live and take place fully on campus. Campus-based learning is focused on providing an interactive learning experience supported by a range of digitally-enabled asynchronous learning opportunities including learning materials, resources, and opportunities provided via the virtual learning environment. On-campus contact hours will be clearly articulated to students.

[2] The module includes a combination of synchronous/ live on-campus and online learning events. These will be supported by a range of digitally-enabled asynchronous learning opportunities including learning materials, resources, and opportunities provided via the virtual learning environment. On-campus and online contact hours will be clearly articulated to students.

[3] Where all learning is solely delivered by web-based or internet-based technologies and the participants can engage in all learning activities through these means. All required contact hours will be clearly articulated to students.

[4] Learning activities where the main location for the learning experience is in the workplace. All required contact hours, whether online or on campus, will be clearly articulated to students

	for selection and repetition, iterating over a collection, functions, methods, and parameters; user-interactions, types of error, exception handling, objects. A rudimentary knowledge of the 4 pillars of object orientation including: abstraction, encapsulation, inheritance, and polymorphism. Abstraction and encapsulation will be practical, and inheritance and polymorphism will be theoretical.
Practice: Applied Knowledge and Understanding	**SCQF 7** Use the following basic computing skills, practices, techniques, and materials: Selecting an appropriate data type, developing simple algorithms using expressions and statements, use of selection and repetition constructs, defining more complex data types involving records and arrays, managing a collection, iterating through a collection, creating functions, calling functions, passing parameters into and out of functions, creating a simple user-interface, interactivity in code. Algorithm design, writing and correcting code, using development tools to build programs, finding errors in programs. Employing encapsulation as an object oriented programming design technique.
Generic Cognitive skills	**SCQF 7** Use object based programming as an approach to solving routine programming problems.
Communication, ICT and Numeracy Skills	**SCQF 7** se an integrated development environment in developing a software application.
Autonomy, Accountability and Working with Others	**SCQF 7** Employ the principles of pair programming. The assessment will allow students to work together in pairs from a version control perspective and can also be implemented remotely with students receiving guidelines on how to share their screens in MS Teams to review each others code

Prerequisites	**Module Code**	**Module Title**
	Other	
Co-requisites	**Module Code**	**Module Title**

Learning and Teaching	
In line with current learning and teaching principles, a 20-credit module includes 200 learning hours, normally including a minimum of 36 contact hours and maximum of 48 contact hours. Class sessions will be used for exposition and exploration of topics, provide context and suggest appropriate background material. The emphasis will be on students developing their own programming skills on-campus lab sessions will provide practical experience in developing programming solutions to problems. Pair programming will be employed where possible and the primary assessment can be attempted in pairs.	
Learning Activities During completion of this module, the learning activities undertaken to achieve the module learning outcomes are stated below:	**Student Learning Hours** (Note: Learning hours include both contact hours and hours spent on other learning activities)

Lecture / Core Content Delivery	18
Laboratory / Practical Demonstration / Workshop	26
Tutorial / Synchronous Support Activity	18
Independent Study	238
Please select	
Please select	
TOTAL	300

Indicative Resources

The following materials form essential underpinning for the module content and ultimately for the learning outcomes:

A PC will be required with the ability to run the Integrated Development Environment Visual Studio (2019 or 2022).

Miles, R. (2015). C# Yellow Programming Book. "Bananas" Edition 7.0. [PDF free download from: http://www.csharpcourse.com/ or available for a nominal price in Kindle format at www.amazon.co.uk]

Purdum, J. (2012). Beginning Object Oriented Programming with C#. John Wiley & Sons. [Also available as an e-book in Kindle format from www.amazon.co.uk]

(N.B. Although reading lists should include current publications, students are advised (particularly for material marked with an asterisk*) to wait until the start of session for confirmation of the most up-to-date material)

Attendance and Engagement Requirements

In line with the Student Attendance and Engagement Procedure, Students are academically engaged if they are regularly attending and participating in timetabled on-campus and online teaching sessions, asynchronous online learning activities, course-related learning resources, and complete assessments and submit these on time.

For the purposes of this module, academic engagement equates to the following:

The School of Computing, Engineering and Physical Sciences considers attendance and engagement to mean a commitment to attending, and engaging in, timetabled sessions. You will scan your attendance via the scanners each time you are on-campus and you will login to the VLE several times per week. Where you are unable to attend a timetabled learning session due to illness or other circumstance, you should notify the Programme Leader that you cannot attend. Across the School an 80% attendance threshold is set. If you fall below this, you will be referred to the Student Success Team to see how we can best support your studies.

Equality and Diversity

The University's Equality, Diversity and Human Rights Procedure can be accessed at the following link: UWS Equality, Diversity and Human Rights Code.

Lab based

Aligned with the University's commitment to equality and diversity, this module supports equality of opportunity for students from all backgrounds and learning needs. Using the VLE, material will be presented electronically in formats that allow flexible access and manipulation of content. This module complies with University regulations and guidance on inclusive learning and teaching practice. This module has lab-based teaching and as such

you are advised to speak to the Module Co-ordinator to ensure that specialist assistive equipment, support provision and adjustment to assessment practice can be put in place, in accordance with the University's policies and regulations.
(N.B. Every effort will be made by the University to accommodate any equality and diversity issues brought to the attention of the School)

Supplemental Information

Divisional Programme Board	**Computing**
Overall Assessment Results	☐ **Pass / Fail** ☒ **Graded**
Module Eligible for Compensation	☐ **Yes** ☒ **No** **If this module is eligible for compensation, there may be cases where compensation is not permitted due to programme accreditation requirements. Please check the associated programme specification for details.**
School Assessment Board	Creative Computing
Moderator	Dr. Gavin Baxter
External Examiner	Prof Syvester Arnab
Accreditation Details	TIGA
Module Appears in CPD catalogue	☐ **Yes** ☒ **No**
Changes / Version Number	1.07

Assessment (also refer to Assessment Outcomes Grids below)
Assessment 1
Small programming lab exercises
Assessment 2
A class test
Assessment 3
An implemented program utilising classes, data structures, menus, files and a user interface
(N.B. (i) Assessment Outcomes Grids for the module (one for each component) can be found below which clearly demonstrate how the learning outcomes of the module will be assessed. (ii) An indicative schedule listing approximate times within the academic calendar when assessment is likely to feature will be provided within the Student Module Handbook.)

Component 1							
Assessment Type	**LO1**	**LO2**	**LO3**	**LO4**	**LO5**	**Weighting of Assessment Element (%)**	**Timetabled Contact Hours**
Portfolio	☒	☐	☐	☐	☐	20	

Component 2

Assessment Type	LO1	LO2	LO3	LO4	LO5	Weighting of Assessment Element (%)	Timetabled Contact Hours
Class Test	☐	☒	☐	☒	☐	20	

Component 3							
Assessment Type	**LO1**	**LO2**	**LO3**	**LO4**	**LO5**	**Weighting of Assessment Element (%)**	**Timetabled Contact Hours**
Portfolio	☐	☐	☒	☐	☐	60	
Combined total for all components						100%	hours

Change Control

What	When	Who
Engagament and EDI policies	24/03/2025	Thomas Hainey
Assessment Components to prevent over assessment	24/03/2025	Thomas Hainey