

## University of the West of Scotland

## Module Descriptor

Session: 2024/25

<b>Title of Module: Programming with Objects</b>			
<b>Code: COMP07070</b>	<b>SCQF Level: 7 (Scottish Credit and Qualifications Framework)</b>	<b>Credit Points: 30</b>	<b>ECTS: 15 (European Credit Transfer Scheme)</b>
<b>School:</b>	School of Computing, Engineering and Physical Sciences		
<b>Module Co-ordinator:</b>	Thomas Hainey		
<b>Summary of Module</b>			
<p>This 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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <ul style="list-style-type: none"> <li>• 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)</li> <li>• 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.</li> </ul>			

- 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

Face-To-Face	Blended	Fully Online	HybridC	Hybrid 0	Work-Based Learning
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**See Guidance Note for details.**

### Campus(es) for Module Delivery

The module will **normally** be offered on the following campuses / or by Distance/Online Learning: (Provided viable student numbers permit) (tick as appropriate)

Paisley:	Ayr:	Dumfries:	Lanarkshire:	London:	Distance/Online Learning:	Other:
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Add name

### Term(s) for Module Delivery

(Provided viable student numbers permit).

Term 1	Term 2	Term 3
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

### Learning Outcomes: (maximum of 5 statements)

**These should take cognisance of the SCQF level descriptors and be at the appropriate level for the module.**

At the end of this module the student will be able to:

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++
<b>Employability Skills and Personal Development Planning (PDP) Skills</b>	
<b>SCQF Headings</b>	During completion of this module, there will be an opportunity to achieve core skills in:
Knowledge and Understanding (K and U)	<p><b>SCQF Level 7</b></p> <p>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 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.</p>
Practice: Applied Knowledge and Understanding	<p><b>SCQF Level 7</b></p> <p>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.</p>
Generic Cognitive skills	<p><b>SCQF Level 7</b></p> <p>Use object based programming as an approach to solving routine programming problems.</p>
Communication, ICT and Numeracy Skills	<p><b>SCQF Level 7</b></p> <p>Use an integrated development environment in developing a software application.</p>
Autonomy, Accountability and Working with others	<p><b>SCQF Level 7</b></p> <p>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</p>

<b>Pre-requisites:</b>	Before undertaking this module the student should have undertaken the following:	
	<b>Module Code:</b>	<b>Module Title:</b>
	<b>Other:</b>	
<b>Co-requisites</b>	<b>Module Code:</b>	<b>Module Title:</b>

\*Indicates that module descriptor is not published.

<b>Learning and Teaching</b>	
<p><b>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.</b></p> <p>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.</p>	
<p><b>Learning Activities</b> During completion of this module, the learning activities undertaken to achieve the module learning outcomes are stated below:</p>	<p><b>Student Learning Hours</b> (Normally totalling 200 hours): (Note: Learning hours include both contact hours and hours spent on other learning activities)</p>
Lecture/Core Content Delivery	18
Laboratory/Practical Demonstration/Workshop	26
Tutorial/Synchronous Support Activity	18
Independent Study	238
	300 Hours Total
<p><b>**Indicative Resources: (eg. Core text, journals, internet access)</b></p>	

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](http://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](http://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:

Students are expected to access lecture materials and other class materials (e.g., videos) through the University's VLE and complete the coursework and meet submission deadlines. Failure to do so will be regarded as an indicator of disengagement with the module.

### 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](#).

Please ensure any specific requirements are detailed in this section. Module Co-ordinators should consider the accessibility of their module for groups with protected characteristics..

(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

<b>Divisional Programme Board</b>	Computing
<b>Assessment Results (Pass/Fail)</b>	Yes <input type="checkbox"/> No <input checked="" type="checkbox"/>

<b>School Assessment Board</b>	Creative Computing
<b>Moderator</b>	Dr. Gavin Baxter
<b>External Examiner</b>	N.Whitton
<b>Accreditation Details</b>	TIGA Accreditation
<b>Changes/Version Number</b>	1.06

<b>Assessment: (also refer to Assessment Outcomes Grids below)</b>
The assessment will consist of 3 components: -3 small programming exercises worth 20%. -1 class test covering the knowledge in the lectures worth 20%. -1 practical implementation worth 60%.
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) <b>Assessment Outcomes Grids</b> 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 <b>indicative schedule</b> listing approximate times within the academic calendar when assessment is likely to feature will be provided within the Student Module Handbook.)

## Assessment Outcome Grids (See Guidance Note)

Component 1							
Assessment Type (Footnote B.)	Learning Outcome (1)	Learning Outcome (2)	Learning Outcome (3)	Learning Outcome (4)	Learning Outcome (5)	Weighting (%) of Assessment Element	Timetabled Contact Hours
	✓					20	

Component 2							
Assessment Type (Footnote B.)	Learning Outcome (1)	Learning Outcome (2)	Learning Outcome (3)	Learning Outcome (4)	Learning Outcome (5)	Weighting (%) of Assessment Element	Timetabled Contact Hours
		✓	✓	✓		20	

Component 3							
Assessment Type (Footnote B.)	Learning Outcome (1)	Learning Outcome (2)	Learning Outcome (3)	Learning Outcome (4)	Learning Outcome (5)	Weighting (%) of Assessment Element	Timetabled Contact Hours
	✓	✓	✓	✓		60	
<b>Combined Total for All Components</b>						<b>100%</b>	<b>XX hours</b>

**Change Control:**

What	When	Who
Further guidance on aggregate regulation and application when completing template	16/01/2020	H McLean
Updated contact hours	14/09/21	H McLean
Updated Student Attendance and Engagement Procedure	19/10/2023	C Winter
Updated UWS Equality, Diversity and Human Rights Code	19/10/2023	C Winter
Guidance Note 23-24 provided	12/12/23	D Taylor
General housekeeping to text across sections.	12/12/23	D Taylor

Version Number: MD Template 1 (2023-24)