Module Descriptor

Title	Software Development for Games		
Session	2025/26	**Status**	Published
Code	COMP08091	**SCQF Level**	8
Credit Points	20	**ECTS (European Credit Transfer Scheme)**	10
School	**Computing, Engineering and Physical Sciences**		
Module Co-ordinator	Dr. Thomas Hainey		

Summary of Module

This module will further consolidate knowledge of rudimentary programming, software development and software engineering concepts used within computer games programming for students to solidify their knowledge of these principles and to allow Direct Entry students to ease into the same level as continuing students . As computer games are themselves pieces of software, the first half of the module will enable students to further synthesis their knowledge of the basic programming constructs of games with practical lab examples and current contextual computer games case study examples. Students will have the opportunity to undertake an exercise on games design where they will be able to select a popular computer game of their choice and decompose or reverse engineer it in terms of the software components/objects. The objective being that the students will be able to create abstract models in terms of the objects and models of games they have played to appreciate the software components, constituents, and interactions within a recognised computer games context. The students will be able to present these findings in the class context to receive peer feedback. In the second half of the module the students will delve deeper into data structures and algorithms for games developers including: arrays, recursion, sorting, lists, stacks, queues, trees and graphs to give them a firm footing of the underlying data structures and algorithms used within software engineering and games development. The students will have a class test on data structures and small practical exercise implementing a particular data structure.

•To provide a basic understanding of the software development/engineering constituents utilised in computer games programming. 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)

•To provide students with consolidated knowledge on rudimentary programming concepts and aid integration of direct entrants

•To provide students with a basic understanding of the underlying data structures and algorithms utilised in computer games

•To introduce a number of data structures to inform future implementation choices

•To provide a theoretical and practical overview of sorting and recursion

•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 centred, and can will be delivered in a hybrid fashion with authentic assessment.

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	To provide a basic understanding of the software development/engineering constituents utilised in computer games programming and to allow students to be able to reverse engineer computer game implementations to the design phase using modelling of objects, structures and relationships
L2	To provide students with a refresher course on rudimentary programming concepts and to provide students with a basic understanding of the underlying data structures and algorithms utilised in computer games by introducing a number of data structures to inform future implementation choices
L3	To produce a practical implementation using a data structure and algorithmic concepts
L4	
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:**

[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

Knowledge and Understanding (K and U)	**SCQF 8** Develop a broad knowledge of the following concepts and principles: object oriented design, data structures, algorithms, sorting, recursion, software development/engineering for games, rudimentary programming.
Practice: Applied Knowledge and Understanding	**SCQF 8** Theoretical and practical implementation of algorithms and data structures in software engineering and in a computer games context. Modelling and presenting a popular case study video game/computer game using a recognised software design method.
Generic Cognitive skills	**SCQF 8** Use a range of approaches to solving routine programming problems including design, implementation, and testing.
Communication, ICT and Numeracy Skills	**SCQF 8** Be able to present the software engineering/development constituents of a game for peer review and discussion as part of in class interactive activities.
Autonomy, Accountability and Working with Others	**SCQF 8** Be able to work autonomously and with others to solve programming problems and implement data structure/algorithm solutions.

Prerequisites	**Module Code** COMP07070	**Module Title** Programming with Objects
	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 and on-campus lab sessions will provide practical experience in developing programming solutions to problems. Pair programming will be employed in a remote capacity 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	24
Laboratory / Practical Demonstration / Workshop	22
Independent Study	154
Please select	

Please select	
Please select	
TOTAL	200

Indicative Resources
The following materials form essential underpinning for the module content and ultimately for the learning outcomes: Penton, R. and LaMonthe, A. (2003). Data Structures for Game Programmers. Premier Press Sherrod, A. (2007). Data Structures and Algorithms for game Developers. Charles River Media Game Development Series
(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. Non-lab based (contact hours = 36) 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. Specialist assistive equipment, support provision and adjustment to assessment practice in accordance with the University's policies and regulations. Lab based (contact = 48) 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 Sylvester Arnab
Accreditation Details	TIGA
Module Appears in CPD catalogue	☐ **Yes** ☒ **No**
Changes / Version Number	2.05

Assessment (also refer to Assessment Outcomes Grids below)
Assessment 1
Class test theoretically assessing data structures, recursion, and sorting algorithms (40%)
Assessment 2
Practical implementation assessing the implementation of an abstract data structure and various sorting algorithms in a computer game context (60%)
Assessment 3
(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**
Class Test/Quiz	☒	☒	☐	☐	☐	40	

Component 2

Assessment Type	LO1	LO2	LO3	LO4	LO5	Weighting of Assessment Element (%)	Timetabled Contact Hours
Portfolio	☐	☒	☒	☐	☐	60	

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

Change Control

What	When	Who
Engagement and EDI regulations	24/03/2025	Thomas Hainey