COMP08091 Software development for games

Title of Module: Software Development for Games			
Code: COMP08091	**SCQF Level: 8 (Scottish Credit and Qualifications Framework)**	**Credit Points: 20**	**ECTS: 10 (European Credit Transfer Scheme)**
School:	School of Computing, Engineering and Physical Sciences		
Module Co-ordinator:	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

Fully Online	HybridC	HybridO	Work-based Learning
	✓		

Fully Online
Instruction that is solely delivered by web-based or internet-based technologies. This term is used to describe the previously used terms distance learning and e learning.
HybridC
Online with mandatory face-to-face learning on Campus
HybridO
Online with optional face-to-face learning on Campus
Work-based Learning
Learning activities where the main location for the learning experience is in the workplace.

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)

Paisley:	Ayr:	Dumfries:	Lanarkshire:	London:	Distance/Online Learning:	Other:
✓						

Term(s) for Module Delivery					
(Provided viable student numbers permit).					
Term 1		Term 2	✓	Term 3	

Learning Outcomes: (maximum of 5 statements)
On successful completion of this module the student will be able to: 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

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 Level 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 Level 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 Level 8. Use a range of approaches to solving routine programming problems including design, implementation and testing.
Communication, ICT and Numeracy Skills	SCQF Level 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 Level 8. N/A	
Pre-requisites:	Before undertaking this module the student should have undertaken the following:	
	Module Code: COMP07070	**Module Title:** Programming with Objects
	Other:	
Co-requisites	**Module Code:**	**Module Title:**

* Indicates that module descriptor is not published.

Learning and Teaching	
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 remote 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** (Normally totalling 200 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
	200 Hours Total

****Indicative Resources: (eg. Core text, journals, internet access)**
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

Jain, H. (2017). Problem Solving in Data Structures & Algorithms Using C++: Programming Interview Guide

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)

Engagement Requirements

In line with the Academic Engagement Procedure, Students are defined as academically engaged if they are regularly engaged with timetabled teaching sessions, course-related learning resources including those in the Library and on the relevant learning platform, and complete assessments and submit these on time. Please refer to the Academic Engagement Procedure at the following link: Academic engagement procedure

Where a module has Professional, Statutory or Regulatory Body requirements these will be listed here:

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

Students are expected to access videos and other class materials through the VLE, complete tutorial exercises and lab exercise and meet submission deadlines, failure to do so will be regarded as an indicator of disengagement with the module. Disengagement from the module is defined as not having interacted within a 4 week period. If this happens then contact will be attempted for conversation about circumstances.

Supplemental Information

Programme Board	Computing
Assessment Results (Pass/Fail)	No
Subject Panel	Creative Computing
Moderator	Dr Gavin Baxter
External Examiner	N Whitton
Accreditation Details	

Changes/Version Number	2.02 The working of the description has been altered slightly to take into account the UWS Curriculum Framework and the module has been moved to T2.

Assessment: (also refer to Assessment Outcomes Grids below)
Class test theoretically assessing data structures, recursion and sorting algorithms (40%)
Practical implementation assessing the implementation of an abstract data structure and various sorting algorithms in a computer game context (60%)
(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 Handbook.)

Assessment Outcome Grids (Footnote A.)

Component 1					
Assessment Type (Footnote B.)	**Learning Outcome (1)**	**Learning Outcome (2)**	**Learning Outcome (3)**	**Weighting (%) of Assessment Element**	**Timetabled Contact Hours**
Class test (written)	✓	✓		40	0

Component 2					
Assessment Type (Footnote B.)	**Learning Outcome (1)**	**Learning Outcome (2)**	**Learning Outcome (3)**	**Weighting (%) of Assessment Element**	**Timetabled Contact Hours**
Creative output/ Audiotapes/ Videotapes/ Games/ Simulations		✓	✓	60	0
Combined Total For All Components				100%	0 hours

Footnotes

A. Referred to within Assessment Section above

B. Identified in the Learning Outcome Section above

Note(s): 1. More than one assessment method can be used to assess individual learning outcomes. 2. Schools are responsible for determining student contact hours. Please refer to University Policy on contact hours (extract contained within section 10 of the Module Descriptor guidance note). This will normally be variable across Schools, dependent on Programmes &/or Professional requirements.

Equality and Diversity
This module is suitable for any student. The assessment regime will be applied flexibly so that a student who can attain the practical outcomes of the module will not be disadvantaged. When a student discloses a disability, or if a tutor is concerned about a student, the tutor in consultation with the School Enabling Support co-ordinator will agree the appropriate adjustments to be made. The module will adhere to the 5th core principle of the Curriculum Framework by recognising the diversity of the student body and the requirement to be accessible to all i.e. a combination of remote and on-campus in the ethos of hybrid delivery. UWS Equality and Diversity Policy UWS Equality and Diversity Policy
(N.B. Every effort will be made by the University to accommodate any equality and diversity issues brought to the attention of the School)