PROG H2222: Programming II

Module Title:	Programming II
Language of Instruction:	English

Credits:	10

NFQ Level:	8

Module Delivered In	No Programmes

Teaching & Learning Strategies:	The course material will be delivered by a mixture of traditional lectures and laboratory based lectures where students can explore programming constructs as they are introduced. Students will also be assigned practical exercises that address the learning outcomes. During the academic year, students will work fulltime on a three week minor project that is undertaken in conjunction with the other second year course modules.

Module Aim:	To equip students with object-oriented programming skills and use object-oriented techniques to solve complex problems.

Learning Outcomes	
On successful completion of this module the learner should be able to:	
LO1	Develop software systems in C++ using the object-oriented paradigm.
LO2	Use a game engine API to develop properly architected short game prototypes.
LO3	Implement design patterns that are applicable to interactive applications.
LO4	Use a version control system to manage source code in a team project.

Pre-requisite learning		
Module Recommendations *This is prior learning (or a practical skill) that is recommended before enrolment in this module.*		
6876	PROG H2222	Programming II
Incompatible Modules *These are modules which have learning outcomes that are too similar to the learning outcomes of this module.*		
No incompatible modules listed		
Co-requisite Modules		
No Co-requisite modules listed		
Requirements *This is prior learning (or a practical skill) that is mandatory before enrolment in this module is allowed.*		
Successful completion of year 1 or equivalent		

PROG H2222: Programming II

Module Content & Assessment

Indicative Content
1. Compilation process, IO and standard libraries, pointers, fundamental language features (type checking, cast operators, function overloading, default function arguments)
2. Classes, members and construction functions, composition, header file organisation.
3. Memory management: operators new, delete and delete [], destructor, overloaded assignment, smart pointers and move semantics.
4. Inheritance: generalisations, specialisation, abstract classes and polymorphism, RTTI operators.
5. Version control systems: committing, checking out, branching and merging.
6. Exception handling.
7. Implementing common design patterns for games.
8. Performance and optimisations.

Assessment Breakdown	%
Continuous Assessment	10.00%
Project	20.00%
Practical	20.00%
End of Module Formal Examination	50.00%

Continuous Assessment

Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Multiple Choice Questions	Class exam	1	10.00	n/a

Project

Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Project	Use a game engine API to develop a short game prototype.	1,2,3,4	20.00	Sem 1 End

Practical

Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Practical/Skills Evaluation	Participation in and completion of practical work	1,2,3	20.00	n/a

End of Module Formal Examination

Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Formal Exam	Three Hour Theory Paper	1,3	50.00	End-of-Semester

SETU Carlow Campus reserves the right to alter the nature and timings of assessment

PROG H2222: Programming II

Module Workload

Workload: Full Time		
Workload Type	*Frequency*	*Average Weekly Learner Workload*
Lecture	30 Weeks per Stage	1.00
Laboratory	30 Weeks per Stage	4.00
Estimated Learner Hours	30 Weeks per Stage	2.00
	Total Hours	210.00