PRAC: Object Oriented Programming

Module Title:	Object Oriented Programming
Language of Instruction:	English

Credits:	5

NFQ Level:	6

Module Delivered In	1 programme(s)

Teaching & Learning Strategies:	The course material will be delivered by a mixture of traditional lectures and laboratory based lectures where learners can explore programming constructs as they are introduced. Students will also be assigned practical exercises that address the learning outcomes.

Module Aim:	To provide learners with object-oriented programming skills and use object-oriented techniques to solve problems of moderate complexity.

Learning Outcomes	
On successful completion of this module the learner should be able to:	
LO1	Develop small components in C++ using the object-oriented paradigm.
LO2	Demonstrate a practical knowledge of memory allocation and the application of pointers, smart pointers and references.
LO3	Use a profiling tool to identify potential bottlenecks in an application.

Pre-requisite learning
Module Recommendations *This is prior learning (or a practical skill) that is recommended before enrolment in this module.*
No recommendations listed
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

Module Content & Assessment

Indicative Content
Introduction and language features Compilation process, IO and standard libraries, addresses and pointers, fundamental language features (type checking, cast operators, function overloading, default function arguments, enumerations)
OOP core concepts part 1 Classes, members and construction functions, composition, header file organisation.
OOP core concepts part 2 Inheritance: generalisations, specialisation, abstract classes and polymorphism, RTTI operators.
Memory management Operators new, delete and delete [], rule of three, smart pointers, move semantics (rule of five).
Optimising code Performance and optimisations, introduction to profiling.

Assessment Breakdown	%
Continuous Assessment	20.00%
Practical	40.00%
End of Module Formal Examination	40.00%

Continuous Assessment

Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Other	Class exam	1	20.00	Week 6

No Project

Practical

Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Practical/Skills Evaluation	Participation in and completion of practical work	1,2,3	40.00	n/a

End of Module Formal Examination

Assessment Type	*Assessment Description*	*Outcome addressed*	*% of total*	*Assessment Date*
Formal Exam	90 minute written examination	1,2	40.00	End-of-Semester

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

Module Workload

Workload: Full Time

Workload Type	*Frequency*	*Average Weekly Learner Workload*
Lecture	12 Weeks per Stage	1.00
Laboratory	12 Weeks per Stage	4.00
Estimated Learner Hours	15 Weeks per Stage	4.33
	Total Hours	125.00

Module Delivered In

Programme Code	Programme	Semester	Delivery
CW_KCCGD_B	Bachelor of Science (Honours) in Computer Games Development	3	Mandatory