MODULE **DESCRIPTOR**

MODULE TITLE	Object Oriented Methods in Computing		
MODULE CODE	CO3402 (L6)	**CREDIT VALUE**	20 / 10 ECTS
SCHOOL	SCHOOL OF SCIENCE		

MODULE **AIMS**

1. To develop the student’s understanding of the concepts of object-oriented software development.
2. To enhance the student’s ability to design and implement systems that tackle complex problems.
3. To provide an opportunity for detailed investigation into applications of object-oriented programming techniques.
4. To encourage the student to apply skills in critical analysis to the evaluation of object-oriented languages and associated tools.

MODULE **CONTENT**

Object-oriented methods are popular in programming, analysis and design. This module will develop the student’s understanding of the underlying concepts and techniques and will apply them to system design and program implementation.

To develop an understanding of the benefits and problems of using the object-oriented approach, the student will study a range of examples of object-oriented applications.

There will be a substantial practical element to extend the student's programming skills.

Concepts of Object-Oriented Languages

Evaluation of OO features for software engineering: encapsulation and data abstraction, single and multiple inheritance, polymorphism, implementing compile-time and run-time (virtual methods) binding of methods, constructors and destructors, interfaces and contract-based programming.
Evaluation of features (e.g. memory management, access control, security, support for concurrency) of object-oriented programming languages, including C++, C#, and Java. Advanced language features and their interaction with object-oriented features: exceptions, templates, const correctness, type coercion, operator overloading.

Object-Oriented Programming

Class libraries, design criteria for the programming interface to a class, components and design issues, implementing components (e.g. in C# .NET or Java).
Generic Programming, Standard Template Library
Aspect-oriented programming
Development and analysis of object-oriented programs.

Object-Oriented Analysis and Design:

OO application architecture (e.g. frameworks, Model-View-Controller and Document/View); Design patterns (e.g. composite, singleton, observer, delegation, façade, adapter, observer). Evaluation of the design pattern approach to software development (including anti-patterns)
Unified Process

INTENDED **LEARNING OUTCOMES**

On successful completion of this module a student will be able to:

1. Explain and evaluate the concepts and applications of object-oriented methods.
2. Evaluate critically object-oriented design techniques and technologies (e.g. design patterns, OO languages features, and frameworks).
3. Apply object-oriented approaches to design and implementation.
4. Translate an object-oriented design into an object-oriented program using a modern development environment.
5. Design and implement an OO-based abstract data type.

TEACHING METHODS

The students will already have experience of developing programs in an object-oriented language and of using simple UML diagrams. This module will allow the students to apply that knowledge to a more in-depth study of object-oriented concepts and applications. Particular emphasis will be placed on discussion of, and practise with, object-oriented technology that is being applied in industry.

In evaluating programming languages, criteria will be discussed and concepts introduced in lectures. Students will carry out practical exercises in labs using different languages but it is not the purpose of this module to provide in-depth language training in multiple languages. Students will be expected to analyse language features more deeply using books, journals and Internet sources.

The assignment will assess practical programming skills. The students may be asked to implement an abstract datatype and compare it with a similar datatype provided by STL. The evaluation section of the assignment write-up will allow students to identify potential improvements as well as to assess the strengths and weaknesses of the tools and techniques used. The assignment will include a short evaluation of a relevant OO concept to allow the student to demonstrate the ability to find, interpret and apply published literature.

The examination will assess the students' ability to explain and evaluate OO concepts and techniques and to apply them to small examples. One question will give the students the opportunity to demonstrate learning from guided research.

Lectures: these will build on the students' previous software development experiences by explaining and comparing OO concepts across languages (e.g. C++, Java, C#) and by applying those concepts to software design.

Tutorials will include reviews of sample code and design exercises. Internet materials will also be used. Practical exercises will use different languages to illustrate the OO concepts and help the students to apply them to solving problems. Students will be expected to use appropriate tools and libraries to support design and implementation.

ASSESSMENT METHODS

This module is assessed through a Coursework: Design, implementation and evaluation (50%) and an examination (50%).