

MODULE DESCRIPTOR

MODULE TITLE	Advanced Programming		
MODULE CODE	CO2402 (L5)	CREDIT VALUE	20 / 10 ECTS
SCHOOL	SCHOOL OF SCIENCE		

MODULE AIMS

1. To enhance the student's programming competence and ability to use a range of programming techniques necessary for the development of effective and quality software.
2. To show how to select and apply appropriate data structures and algorithms to implement a software solution to a complex problem
3. To improve the student's software development skills using case studies and assignments to enable them to develop readable, maintainable and robust software
4. To provide a practical introduction to the object-oriented and event-driven programming paradigms

MODULE CONTENT

This module could be delivered using any modern object-oriented language such as C++, C# or Java. The following list is indicative using C++ as an example of an appropriate programming language:

The language (C++):

- Operator precedence
- Pointers
- Dynamic memory allocation
- Memory leak detection and avoidance
- Array and pointer relationship
- Functions (pass parameters by address, default parameters, overloading)
- In-line functions
- Classes
- Inheritance and virtual methods
- Polymorphism
- Operator overloading
- Exceptions
- Templates
- The Standard Template Library
- Deep and shallow copy
- Recent revisions to the language, e.g. C11 and C14

Programming paradigm

- Object-oriented

Data Structures and Algorithms:

For example:

- Linked lists
- Stacks
- Recursion
- Dynamic arrays
- Advanced sorting (e.g. quicksort)

Advanced searching (e.g. hashing and chaining)

INTENDED LEARNING OUTCOMES

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

1. Evaluate critically the features of the studied computer programming language
2. Implement and document a structured program to meet a given specification
3. Select and apply appropriate data structures and algorithms to a given problem
4. Critically evaluate a computer program with regard to robustness, usability, maintainability, readability and efficiency

TEACHING METHODS

Practical skills are emphasised. Lectures disseminate information and enable student feedback and interaction. Laboratory sessions develop students' practical skills as they implement topics discussed in the lectures. Tutorials focus on and discuss the more complex lecture material to provide help and feedback with laboratory and assignment problems. Tutorials also act as a general technical forum for any related technological issues. Questions arising from the students' private research are discussed and application of the taught concepts is related to real industrial applications. Tutorials in the latter part of the year are dedicated to examination revision.

Coursework measures the students' practical skills and competence by assessing their achievements in learning outcomes 2 to 4.

ASSESSMENT METHODS

This module is assessed through an examination (50%) and a Coursework: Software Development to solve a non-trivial problem with test strategy and critical evaluation (50%).