



### **DBLN CPSI 2290**

This course is being offered at Griffith College, CAPA's academic partner in Dublin. The Irish academic system differs from the US, particularly with grading. Griffith College professors expect students to undertake a good deal of independent study to achieve a high mark in their classes.

For additional information about this class, please contact the Boston Program Advising Team at 1-800-793-0334.

#### **Computer Programming**

Continuous Assessment: 70%

Exam: 30%

#### **Intended Module Learning Outcomes**

On successful completion of this module learners will be able to:

1. design, develop, test and debug software applications utilising core programming concepts

2. specify precisely the syntax and semantics of programming language
3. select an appropriate program construct (or datatype) to achieve a given task
4. design a systematic suite of tests for a given program and implement it
5. prepare the text of a program in a well-formatted, conventional manner and develop these programmes using an integrated development environment
6. implement basic algorithms and data structures using a programming language
7. document accurately the design of a program on-the-fly
8. determine the basic efficiency of an algorithm

### **Module Objectives**

This module teaches the learner how to design high-quality programs in a systematic way. All the relevant concepts and techniques are explained and exemplified in the clearest, simplest language. The module aims to introduce the learner to the concepts of programming and problem solving.

### **Module Curriculum**

#### **Introduction to problem solving**

- How do you complete a task?
- Identifying sub-components on larger task
- Defining order of subcomponents
- Creation of algorithms
- Stepwise design of programs

#### **Introduction to programming**

- Expressions and statements
- Basic arithmetic
- Comments
- Variables and assignment (Integers, doubles, booleans, characters)
- Boolean expressions and logic
- Conditional statements
- Iteration statements
- User input
- Output
- String manipulation
- Sub-routines (Parameters, Signature, Procedures, Copy rule.)
- Arrays
- 2D arrays
- Text File I/O
- Binary File I/O

- Records
- Simple sorting algorithms (selection sort, bubble sort, insertion sort)
- Binary search

### **Professional Practice**

- Developing a good coding style
- Using comments effectively
- Naming conventions
- Indentation
- Code structure