

Module Title:	Software Development I: introduction to programming
Module Code:	03C
Level:	4
Credits:	15
Pre-Requisites:	None

Module Description:

This module provides students with the opportunity to learn and practise key procedural programming concepts. Students learn to design, write and test programs that solve a range of small problems. Simultaneously, students are introduced to a range of professional practices that form a key part of the software engineer's role, such as the industry standards for writing software and maintaining version control.

Indicative Content:

- Awareness of programming environments, languages, libraries and tools
- Design and creation of readable, maintainable code, making use of modified data flow diagrams
- Sequence, selection, and iteration
- Data types, variables, constants and operators
- Arrays, memory addresses and pointer arithmetic
- Standard input and output functionality
- Functions and parameter passing by value and by reference
- Design, execution and documentation of testing
- Debugging strategies and tools
- Importance of version control, and software tools that facilitate it
- Documentation for administrator and for end user; professional standards of documentation

Learning and Teaching Methods:

The module will be delivered through a combination of lectures and workshops (3 hours per week).

Module Specifications: Schools of Business & Management & Information Technology

Specific Learning Resources:

Visual Studio Express 2012

Text editor

Bibliography

Highly Recommended

Perry, G., (2013) *Absolute Beginner's Guide to C*. Harlow: Pearson Education Ltd.

Vickers, P., (2008) *How to Think Like a Programmer: Problem-Solving for the Bewildered*. London, UK: Cengage.

Recommended

Deitel, H., (2013) *C How To Program. (7th edition)*. Harlow: Pearson Education Ltd

Module Learning Outcomes

Subject Specific Learning Outcomes

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

LO	Analyse a problem and design an algorithm to solve it
LO	Write, test and debug effective programs to accurately implement simple algorithms
LO	Appreciate the importance of professional practices in software engineering, and be able to apply these practices when designing, writing, and documenting programs

Assessment Title or element	Weighting (%)
Assignment 1: basic programming exercises; design diagrams, code listings and test plans documented, plus reflection (300 words) [mid semester]	30%
Assignment 2: more advanced programming exercises; design diagrams, code listings and test plans documented, plus reflection (500 words) [late semester]	40%
Examination: fundamental concepts of programming (1 hour) [end semester]	30%