Dublin Institute of Technology

BSc (Hons) in Information Technology
(Part-time)

Part B - Course Document

This document was prepared by the Course Committee on behalf of the School of Computing.

December 2001
# Table of Contents

1. **INTRODUCTION** .............................................................................................................................4

2. **COURSE AIMS AND OBJECTIVES** ..............................................................................................5
   2.1 PHILOSOPHY AND RATIONALE .....................................................................................................5
   2.2 COURSE AIMS ...................................................................................................................................6
   2.3 COURSE OBJECTIVES ....................................................................................................................6

3. **DURATION AND GENERAL STRUCTURE** ....................................................................................8

4. **ADMISSION CRITERIA** ..................................................................................................................9
   4.1 ENTRY REQUIREMENTS ................................................................................................................9
   4.2 EXEMPTIONS ....................................................................................................................................9

5. **CURRICULUM** ...............................................................................................................................10
   5.1 TEACHING METHODS ..................................................................................................................11

6. **COURSE ASSESSMENT** ................................................................................................................12
   6.1 ASSESSMENT REGULATIONS .......................................................................................................12
   6.2 ASSESSMENT OF SUBJECTS ........................................................................................................12
   6.3 EXAMINATION SCHEDULE .........................................................................................................12
   6.4 FINAL GRADE ................................................................................................................................12

7. **COURSE PROGRESSION** ..............................................................................................................14

8. **SYLLABI** .........................................................................................................................................16
   8.1 SENG1200 INFORMATION SYSTEMS .........................................................................................16
   8.2 PROG1100 SOFTWARE DEVELOPMENT ...................................................................................18
   8.3 TECH1100 COMPUTER TECHNOLOGY .......................................................................................20
   8.4 SENG2100 OBJECT ORIENTED SOFTWARE ENGINEERING .................................................22
   8.5 SENG2200 DATABASE SYSTEMS .................................................................................................24
   8.6 SENG2300 IS PROJECT MANAGEMENT ......................................................................................26
   8.7 SENG2301 PROFESSIONAL ISSUES IN IS PRACTICE ...............................................................28
   8.8 PROG2200 OBJECT ORIENTED PROGRAMMING ...................................................................30
   8.9 TECH2100 ARCHITECTURE & SYSTEM SOFTWARE ................................................................32
   8.10 TECH2300 NETWORKS & DATA TRANSMISSION ..................................................................34
   8.11 SENG3100 SOFTWARE ENGINEERING .....................................................................................36
   8.12 SENG3200 ADVANCED DATABASE SYSTEMS .......................................................................38
   8.13 SENG3300 MANAGEMENT INFORMATION SYSTEMS ............................................................40
   8.14 SENG3301 E-COMMERCE ........................................................................................................42
   8.15 TECH3400 INTERNET DEVELOPMENT .....................................................................................44
   8.16 TECH3500 DISTRIBUTED INFORMATION SYSTEMS ...............................................................46

9. **GUIDELINES FOR PROFESSIONAL PROJECT** ...........................................................................48
   9.1 INTRODUCTION ............................................................................................................................48
   9.2 AIM ..................................................................................................................................................48
   9.3 STAGES OF THE PROJECT ............................................................................................................48
   9.4 ROLE OF THE SUPERVISOR .........................................................................................................52
   9.5 ROLE OF THE STUDENT ..............................................................................................................52
   9.6 ROLE OF PROJECT CO-ORDINATORS AND MONITORS .......................................................53
   9.7 ASSESSMENT OF WORK ..............................................................................................................53
Table of Tables and Figures

TABLE 1: STAGE 1 SUBJECTS AND WEEKLY LEARNING HOURS ................................................................. 10
TABLE 2: STAGE 2 SUBJECTS AND WEEKLY LEARNING HOURS ............................................................. 10
TABLE 3: STAGE 3 SUBJECTS AND WEEKLY LEARNING HOURS ............................................................. 11
TABLE 4: CLASSIFICATION OF FINAL AWARD ....................................................................................... 13

FIGURE 1 COURSE STRUCTURE .............................................................................................................. 15
1 Introduction
This is the course document for the BSc (Hons) in Information Technology. The course will be offered on a part-time basis at evenings and/or weekends. The documentation is prepared in accordance with the DIT Course Quality Assurance Handbook 1997 to satisfy the quality assurance requirement for a Dublin Institute of Technology academic award.

The School of Computing is responsible for supporting the course.

The Faculty of Science presently consists of four Schools

- School of Computing
- School of Physics
- School of Chemistry
- School of Biological Sciences
- School of Mathematical Sciences

The major courses currently being offered by the School are:

- The Computer Science stream of a two subject four-year Hons. BSc in Applied Science - with Mathematics, Physics, Chemistry or Software Engineering (only in fourth year)
- Three-year Diploma in Computer Science
- Four-year Honours Degree, BSc. in Computer Science
- MSc. in Information Technology for Strategic Management
- MSc. in Distributed Information Systems
- Professional Examinations of British Computer Society (Certificate, Diploma and Professional Graduate Diploma).

The School also offers computer application courses to other Schools in the Faculty of Science.
2 Course Aims and Objectives

2.1 Philosophy and Rationale

The structure and philosophy of this course was significantly influenced by “Institutes of Technology and the New Economy” by Dr. Sean McDonagh, which was commissioned by the Irish Government and has been accepted by them as their policy. This report outlines the strategy for the Institutes of Technology in Ireland in meeting the well-documented skills shortage in IT in Ireland. This document concludes, among other things, that

“Institutes of Technology in order to promote widespread access and participation ... introduce Flexible Learning i.e. credit-based learning enabling the freer choice of modules ...”

This course is a part-time course. It is aimed at individuals who are or wish to become professionals in the IT industry. The IT industry has many and varied career paths. The type of graduate we expect from this course is an individual who will be equipped with the skills necessary to function as an IT professional in a broad range of environments.

Students will be expected to have significant IT experience of at least one year’s duration prior to entry to the course. As a consequence of this and considering the part-time nature of the course, prospective students, including those seeking advanced entry, will have a diverse range of IT skills and experience. We are seeking to accommodate this experiential knowledge and to build on it. We are not aiming to produce graduates in any one specific area. This presents certain challenges in the design of the course. We provide core modules which give the student a comprehensive understanding of various aspects of IT including Information Systems, Software Development and Computer Technology. We then provide a broad range of more specifically focused modules, for example Object-Oriented Programming, Software Engineering, Advanced Database Systems, E-Commerce and Internet Development, which allow a student to pursue different areas of interest. We allow the student to select the specific areas of IT which interest him or her and which will complement existing skills or academic knowledge.

The course actively promotes the idea of life-long learning and ladders of opportunity. The course positively supports the entry of students with prior work experience and accredited prior learning. To facilitate this, all applications, whether at advanced or basic entry level will be evaluated by the course Admissions and Exemptions committee which is detailed in Part A of the course documentation. This committee considers all prior work experience and accredited prior learning and, where necessary, will advise students individually on how to attain the appropriate pre-requisite experience in order to be able to successfully complete specific modules or stages of the course.

Due to the demands of the IT workplace and in accordance with the McDonagh report, we allow students to progress through the course at their own pace. A formal review of the syllabi takes place every five years in accordance with QA procedures.
However, it is departmental policy to internally review syllabi each year to take account of the changing nature of IT.

2.2 Course Aims

The main aim of the course is to produce graduates with the necessary skills to function as information technology professionals in a broad range of commercial, industrial and public sector environments.

A further aim of the course is that, in view of the rapidly changing nature of the computer industry, graduates are equipped with a wide and educationally sound background in order that they may possess the flexibility likely to be required in the future.

As such, the course incorporates a wide range of information technology skills with a strong practical component. In addition, all students undertake a final year project which requires them to use those skills they have developed to research, design and produce a thesis of professional standard. The project will not be confined to the implementation of a software system but may be of a theoretical nature.

2.3 Course Objectives

The main course objectives are to

- Enable students to obtain Honours Degree level qualifications in information technology.
- Increase the students' knowledge of and expertise in information technology.
- Facilitate progression to senior positions in the information technology or related sectors.

On completion of the course and depending on the options taken the students will be expected to:

- have developed the necessary skills to design and implement well structured and documented computer programs using a variety of programming languages,
- have a detailed knowledge of systems development and software engineering techniques using a variety of approaches,
- have a thorough understanding of the principles and techniques of computer based information systems,
- have a detailed knowledge of the Internet and World Wide Web and a practical understanding of the skills needed to develop applications using the main Web programming technologies,
- have a thorough understanding of the basic principles of computer hardware, data transmission and local area networking as well as a fundamental knowledge of operating systems,
- be familiar with a range of modern computer packages and tools,
- have developed the necessary skills to prepare management reports, investigative reports on new systems and be familiar with interviewing techniques,
• understand the principles of management functions and the structure of business and other organisations,
• have developed a thorough understanding of the context in which the information technology professional will be expected to work.
3 Duration and General Structure
The course is normally of five years duration. It is modular and divided into two stages. Stages 1 & 2 leading to a Diploma in Information Technology, normally take three years. Students are required to complete three core modules (Stage 1) and four core and one option modules (Stage 2).

Stage 3 leading to a BSc. (Hons) in Information Technology may take two further years but students may take longer. Students are required to complete three option modules and a project. Continuous assessment is an integral part of every module offered on the course.

Graduates will be awarded an Honours Degree in Information Technology by the Institute. However, students who do not wish to complete Stage 3 may graduate with a Diploma in Information Technology if all of the Stages 1 & 2 modules and a project have been completed satisfactorily.

Each subject is awarded the appropriate ECTS credits. The subjects and the associated ECTS credits are detailed in the Curriculum section, page 10.
4 Admission Criteria

4.1 Entry Requirements
There are a number of methods of entry to the course listed below. However students will be expected to have significant information technology experience of at least one year's duration prior to entry to the course. This could be attained through work experience in the computing industry or other activities where information technology is extensively practised. Entry is at the discretion of the Admissions & Exemptions Committee (a sub-committee of the Course Committee, the roles and responsibilities of which are detailed in Part A of the validation documentation).

Irish Leaving Certificate
Minimum requirements for the course are the Irish Leaving Certificate with a grade D3 or higher in six subjects at ordinary level including a grade C3 or higher in at least 2 subjects at higher level.

Application with an equivalent qualification
Places may be offered to applicants with a qualification considered equivalent to the minimum requirements, as determined by the Course Committee.

Non-standard application
Places may be offered to non-standard students who meet certain criteria with respect to suitability, analytical skills and professional experience. Students may be requested to attend for interview.

Mature students
This course positively supports the entry of mature students; experiential knowledge will be favourably considered. This is in keeping with the concept of lifelong learning and the long-established philosophy of the Dublin Institute of Technology.

4.2 Exemptions
Students with qualifications from recognised institutions will be considered for exemptions from particular modules or stages as determined by the Admissions and Exemptions Committee. No exemptions will be available from modules or the project at Stage 3.

This course positively supports and allows students to build on pre-existing qualifications at a number of different levels. This will accommodate Advanced Entry.

Students are referred to “Guide to Transfers and Exemptions” published with the Student Handbook.
5 Curriculum

Table 1, Table 2 and Table 3 below outline the curriculum for each stage of the course, detailing the learning hours and ECTS credits that apply to each subject.

Students who wish to exit the course on completion of Stage 2 may do so with a Diploma provided that they also complete a project of appropriate standard.

To attain the Degree, students must have successfully completed or be exempt from Stages 1 & 2 and satisfactorily complete three modules from the list of options as detailed in Table 3 and the project. In particular, students who may not have completed a course in Professional Issues in IS Practice may be required to complete the appropriate module, at the discretion of the Exemptions Committee.

All options may not be offered each year.

Table 1: Stage 1 Subjects and Weekly Learning Hours

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Subject Title</th>
<th>Learning Hours /Week</th>
<th>Core/ Option</th>
<th>ECTS Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>SENG1200</td>
<td>Information Systems</td>
<td>2 1 2</td>
<td>C</td>
<td>20</td>
</tr>
<tr>
<td>PROG1100</td>
<td>Software Development</td>
<td>2 1 2</td>
<td>C</td>
<td>20</td>
</tr>
<tr>
<td>TECH1100</td>
<td>Computer Technology</td>
<td>2 1 2</td>
<td>C</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 2: Stage 2 Subjects and Weekly Learning Hours

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Subject Title</th>
<th>Learning Hours /Week</th>
<th>Core/ Option</th>
<th>ECTS Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>SENG2100</td>
<td>Object Oriented Systems Analysis and Design</td>
<td>2 1 2</td>
<td>C</td>
<td>20</td>
</tr>
<tr>
<td>SENG2200</td>
<td>Database Systems</td>
<td>2 1 2</td>
<td>C</td>
<td>20</td>
</tr>
<tr>
<td>SENG2300</td>
<td>IS Project Management</td>
<td>2 1 2</td>
<td>C</td>
<td>20</td>
</tr>
<tr>
<td>SENG2301</td>
<td>Professional Issues in IS Practice</td>
<td>1 2</td>
<td>C</td>
<td>10</td>
</tr>
<tr>
<td>PROG2200</td>
<td>Object Oriented Programming</td>
<td>2 1 2</td>
<td>O</td>
<td>20</td>
</tr>
<tr>
<td>TECH2100</td>
<td>Architecture &amp; System Software</td>
<td>2 1 2</td>
<td>O</td>
<td>20</td>
</tr>
<tr>
<td>TECH2300</td>
<td>Networks &amp; Data Transmission</td>
<td>2 1 2</td>
<td>O</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Project (for graduation with Diploma only)</td>
<td>1 4</td>
<td>C</td>
<td>30</td>
</tr>
</tbody>
</table>

3 L - Lecture hours, T/P - Tutorial and Practical hours, S/S - Self-Study hours.
Table 3: Stage 3 Subjects and Weekly Learning Hours

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Subject Title</th>
<th>Learning Hours/Week</th>
<th>Core/Option</th>
<th>ECTS Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>SENG3100</td>
<td>Software Engineering</td>
<td>3 L 1 T/P 3 S/S</td>
<td>O</td>
<td>20</td>
</tr>
<tr>
<td>SENG3200</td>
<td>Advanced Database Systems</td>
<td>3 L 1 T/P 3 S/S</td>
<td>O</td>
<td>20</td>
</tr>
<tr>
<td>SENG3300</td>
<td>Management Information Systems</td>
<td>3 L 1 T/P 3 S/S</td>
<td>O</td>
<td>20</td>
</tr>
<tr>
<td>SENG3301</td>
<td>E-Commerce</td>
<td>3 L 1 T/P 3 S/S</td>
<td>O</td>
<td>20</td>
</tr>
<tr>
<td>TECH3400</td>
<td>Internet Development</td>
<td>3 L 1 T/P 3 S/S</td>
<td>O</td>
<td>20</td>
</tr>
<tr>
<td>TECH3500</td>
<td>Distributed Information Systems</td>
<td>3 L 1 T/P 3 S/S</td>
<td>O</td>
<td>20</td>
</tr>
<tr>
<td>Project</td>
<td></td>
<td>1 L 6 T/P</td>
<td>C</td>
<td>30</td>
</tr>
</tbody>
</table>

5.1 Teaching Methods

Course delivery is effected through lectures, tutorials, web-based support, seminars and practical classes. The breakdown for these elements is detailed with the syllabus of each subject in this document. Practical classes take place in the computer laboratories.

Self study is an important element of the course and every effort is made to encourage the students to take responsibility for their own learning. Course materials will be available online to facilitate the part-time nature of the course and the recognition that most candidates will be in full time employment.
6 Course Assessment

6.1 Assessment Regulations
The General Assessment Regulations (1998) of the Dublin Institute of Technology govern all examinations and assessment procedures on the course except where otherwise specified below.

Assessment on this course is by written examination, continuous assessment and project assessment.

6.2 Assessment of Subjects
Each module is presented as a single paper at the Examination Board.

For each module, the assessment of the paper consists of a 3 hour written examination for which 70% of marks are allocated and a program of continuous assessment for which 30% of the marks are allocated.

In all stages, a student must pass written examinations and continuous assessment individually to pass a module. The grade awarded is a weighted average of the two.

The project is presented as one paper to the Examination Board.

The pass mark for all papers (including the project) is 40%.

6.3 Examination Schedule
There will be two examination sittings per year, at the end of the academic year and in the Autumn. Students may sit for any number of papers that they wish, subject to the course progression regulations as detailed in Section 7.

Compensation between subjects is not allowed.

A student who is not successful in a paper may resit the paper and may carry the continuous assessment mark if and only if the student achieved 40% or more in the continuous assessment component of the paper. Otherwise the student will have to repeat both the written examination and the continuous assessment.

The final grade of the Degree is not prejudiced by resitting papers.

6.4 Final Grade
The Diploma is offered ungraded. This is to allow for exemptions and advanced entry into the course.

The final grade of the Degree is determined by the overall average mark attained on the four papers at Stage 3, i.e. three option modules and one project.
The award of BSc in Information Technology may be made with the classifications of First Class Honours, Second Class Honours and Pass in accordance with the schedule set out in Table 4 below:

Table 4: Classification of Final Award

<table>
<thead>
<tr>
<th>Average Mark Band</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;=70%</td>
<td>First Class Honours</td>
</tr>
<tr>
<td>60% - 69%</td>
<td>Second Class Honours, Upper Division</td>
</tr>
<tr>
<td>50% - 59%</td>
<td>Second Class Honours, Lower Division</td>
</tr>
<tr>
<td>40% - 49%</td>
<td>Pass</td>
</tr>
</tbody>
</table>
7 Course Progression

To attain the Diploma, students must satisfactorily complete or be exempt from eight modules and the Project.

Students progress from Stage 2 to Stage 3 and achieve the degree having successfully passed the appropriate numbers of papers as detailed in Section 5 above. Students may carry a maximum of one paper from any stage to a subsequent stage, at the discretion of the Examination Board, but will not be awarded the Diploma or the Degree until successful in all relevant papers and project. Students may not be awarded a degree unless they have successfully completed or been exempted from Stage 2. A student cannot sit an exam at any stage of the course if they have more than one module outstanding from the previous stage.

Therefore, students must pass or be exempt from all Stage 2 modules before Stage 3 can be attempted. In practice, if a student has one subject in Stage 2 outstanding, they can take that at the same time as some of the Stage 3 modules. However, their Stage 3 exam results cannot be considered before they have passed all of Stage 2. It should be noted, however, that any failure at Stage 2 means that consideration of Stage 3 results will have to be deferred.

It is, therefore, not feasible for students with more than one Stage 2 module outstanding to proceed to Stage 3. Consideration of exam results is a procedural thing but it means that (for example) in the case of someone taking their last Stage 2 module next year as well as their Stage 3 modules, if the Stage 2 module was failed the Stage 3 results must be held back until Stage 2 is completed.

Entry to specific modules at Stage 3 may require that certain pre-requisites have been met. This is at the discretion of the Admissions and Exemptions Committee.

There is flexibility on the number of papers that a student is required to take in any one academic year. In any academic year there is no minimum number of papers required. This means that a student may progress through this course at their own pace.

It should be noted that no paper may be attempted more than four times over six examination sessions.
A project must be taken in Stage 2 if the student wishes to exit from the course with a Diploma.

Figure 1 Course Structure
8 Syllabi

8.1 SENG1200  Information Systems

Breakdown of Hours per Week:  
ECTS Credits: 20
Lecture: 2
Tutorial/Practical: 1
Self-Study: 2
Core/Option: C

Course Aims:
- To develop an awareness of the nature and use of information and information systems in an organisational context,
- To introduce the various techniques used within systems analysis and design,
- To foster an appreciation of the different types of methodologies used in the system development process
- To provide an introduction to data management and organisation.

Learning Outcomes:
On completion of the course the student will:
- Have an understanding of the different types of information and how organisations use information
- Understand and appreciate the software development process and the different approaches to software development
- Display a basic knowledge and understanding of systems development techniques
- Understand the basic ideas of storing and manipulating data
- Have knowledge of data analysis and modelling techniques and the different database architectures
- Understand the different testing strategies and implementation techniques
- Be able to demonstrate an awareness of user interface design issues

General Subject Matter:
This module provides the student with a comprehensive introduction to the software development process, systems analysis and design, data analysis and management, database systems, software information systems and the organisations that use them.

Syllabus:
Data Management: The characteristics of data and information. Data acquisition and presentation. File management: organisation and access methods. Database design, entity modelling and normalisation. Database architectures and database management systems. Data security, integrity and control.
Systems Development: The systems development life-cycle and the activities and techniques of requirements analysis, software design, coding, testing, implementation and support. System methodologies. Structured systems analysis and design techniques. Rapid application development and prototyping techniques. Introduction to object-oriented development. Human computer interface (HCI) design aspects. Tools used in software development.
Teaching Approach:
- Lectures
- Case studies / Assignments
- Tutorials

Assessment Methods:
Written Examination – 70%
Continuous Assessment – 30%

Essential Reading List:
8.2 PROG1100 Software Development

Breakdown of Hours per Week: ECTS Credits: 20
Lecture: 2
Tutorial/Practical: 1
Self-Study: 2
Core/Option: C

Course Aims
- To understand the fundamental concepts of structures programming.
- To emphasise skills in problem solving and algorithm specification.
- To introduce the students to windows programming.

Learning Outcomes
On completion of the course the student will
- Have a high proficiency in a 3rd generation programming language
- Have a detailed knowledge and understanding of a range of fundamental algorithms and data structures.
- Understand the principles of multiple module program construction.
- Be competent in the techniques of structured problem analysis, program construction documentation and testing.
- Understand the basic concepts of windows programming and user-interface design.

General Subject Matter
This course gives the student a comprehensive introduction to programming practice including structured programming and visual programming techniques.

Syllabus
Fundamental Concepts of the Programming Process.
Concept of an algorithm
Development and semi-formal specification of algorithms, based on a simplified computer model

Programming Concepts
Data types: numeric and non-numeric, elementary and derived, subtypes Expressions, assignments, input/output
Control structures: selection and iteration
Subprograms: procedures and functions

Data Structures
Strings, single-dimensional arrays, two-dimensional arrays and enumerated data types.

Dynamic Data Structures
Pointers, dynamic variables, implementing and process stacks, queues and linked lists.

Algorithms
Sorting and searching algorithms – efficiency with respect to computation and storage.

Records and File Handling
File types, file processing and records.

Windows Programming
Introduction to windows programming (Visual Basic), HCI and event driven Programming.
Teaching Approach:
- Lectures
- Case studies / Practicals
- Tutorials

Assessment Methods:
Written Examination – 70%
Continuous Assessment – 30%

Essential Reading List

Background Reading List
8.3 TECH1100 Computer Technology

Breakdown of Hours per Week: ECTS Credits: 20
Lecture: 2 Core/Option: C
Self-Study: 2

Course Aims
- To develop an understanding of the principles on which computer systems are based
- To introduce the fundamental building blocks of the digital computer
- To describe the functionality of the various peripherals that may be attached to computers
- To introduce networking and data communications

Learning Outcomes
On completion of the course the student will
- Be able to design the fundamental building blocks of a modern computer using digital logic techniques and Boolean algebra.
- Be familiar with the basic instruction set and simple addressing modes of a microprocessor and be able to use them to execute a program
- Appreciate the advantages and disadvantages of the various memory devices found in the memory hierarchy of a computer system
- Have a basic understanding of the principles and functionality of common PC peripherals
- Have a basic understanding of the components of an operating system
- Understand the functionality and purpose of compilers and compiler types
- Have a basic understanding of protocols and standards used in digital communications

General Subject Matter
Provide the students with an overview of the fundamentals of computer system hardware and system software.

Syllabus
Computer system fundamentals
Number representation and arithmetic, digital logic

Microprocessor architecture
Essential functions of processor and stored program concept, simple processor instruction set, storage/Memory devices, peripheral devices, peripheral interface to CPU, memory mapped I/O and interrupts

Operating Systems
Interrupts, concurrency, scheduling, memory management, resource allocation, i/o

Systems Software
Compilers, interpreters, assemblers, file organisation

Networking
Communication protocols, transmission media, network topology, modems, error detection, lans, wans and internetworking, internet protocols
Teaching Approach:
• Lectures
• Case studies / Practicals
• Tutorials

Assessment Methods:
Written Examination – 70%
Continuous Assessment – 30%

Essential Reading List

Background Reading List
8.4 SENG2100 Object Oriented Software Engineering

Breakdown of Hours per Week:                          ECTS Credits: 20
Lecture: 2                         Core/Option: C
Tutorial/Practical: 1
Self-Study: 2

Course Aims:
- to provide the students with an understanding and appreciation of the role of systems analysis and design, building and testing within an object oriented system development life cycle approach,
- to teach the different techniques used in object oriented analysis and design
- to teach software testing strategies and techniques and the application of testing techniques to object oriented developed systems.

Learning Outcomes:
On completion of the course the student will:
- understand the different approaches to object oriented software development and methodologies supporting them,
- be able to use appropriate methods and techniques and to perform an object oriented analysis and design for a given case study,
- know how to develop an appropriate testing strategy,
- understand the test process and be able to develop test cases,
- to appreciate the various tools available to assist with object oriented analysis and design and testing.

General Subject Matter:
This module enhances the knowledge of system development approaches developed in SENG1200 – Information Systems. It studies systems analysis and design in the context of object oriented development. On successful completion of the course students are expected to be able to apply object oriented analysis and design techniques. Students are also expected to appreciate and understand the differences between structured and object oriented design.

Syllabus:
Review of the software development process: Life cycle models; Development approaches; structure systems development, object oriented development, the strengths and weaknesses of different approaches.
Object Oriented Principles and Concepts: objects, classes, instances, encapsulation, abstraction, generalisation, specialisation, aggregation, inheritance, polymorphism, messaging.
Object Oriented Life Cycle: Inception, Elaboration, Construction, Transition, UML, object oriented methodologies.
Analysis and design techniques: Requirements gathering, feasibility studies, functional and non-functional requirements, fact-finding techniques. Techniques for modelling classes and transactions. Notation used. File and database organisations and structure, normalisation, using a database with an object oriented developed system.
Examples of techniques required: class diagrams, business and interface classes, use case diagrams, object sequence diagrams, object collaboration diagrams, state transition
diagrams, component deployment diagrams.

Testing: validation and verification, the review process, the testing process, test strategies, V-model of testing, developing test cases, unit testing, integration testing, system testing, debugging, regression testing, the management of testing.

Teaching Approach:
- lectures
- case studies / assignments
- tutorials

Assessment Methods:
Written Examination – 70%
Continuous Assessment – 30%

Essential Reading List:

Background Reading List:
8.5  SENG2200  Database Systems

Breakdown of Hours per Week:  
ECTS Credits: 20
Lecture: 2  
Tutorial/Practical: 1  
Self-Study: 2
Core/Option: C

Course Aims
- To provide the student with a detailed knowledge of the principles of database systems.
- To give the students the necessary skills to be an administrator of a database system.
- To provide the student with the skills to develop a relational database application.

Learning Outcomes
On completion of the course the student will
- Have a detailed knowledge and understanding of the theoretical aspects of a relational model.
- Be able to specify the functional requirements for a database application.
- Be able to create data models and their associated logical schema.
- Have a high level of proficiency in SQL programming.
- Have acquired the skills of database administration.
- Have created a relational database application.

General Subject Matter
A database system is the cornerstone of a modern information system. This course gives students the necessary skills required to understand, design and implement such a database system.

Syllabus
An Introduction to Databases
Fundamentals of Information Systems, Comparison of databases with file systems.
Classification of Databases
The Relational Model
Data Structures (tables, tuples, domains).
Data integrity (keys, entity integrity, referential integrity)
Relational algebra and Relational calculus.
Relational constraints and Exception handling
Database Design
Transforming a logical data model into a relational database.
Normalization (first, second and third normal forms).
Database Management Systems
Components of a database management system
The Database Language SQL
Standards and basic structure of SQL.
Data Definition, Data Manipulation
Inserts, Updates, Deletes, Referential Integrity Constraints, Indexes, Views
Procedures, Constraints and Triggers
Embedded SQL
Database Administration
Role of the database administrator.
Data protection, Recovery, Security and Integrity.

Teaching Approach:
• Lectures
• Case studies / Practicals
• Tutorials

Assessment Methods:
Written Examination – 70%
Continuous Assessment – 30%

Essential Reading List
Howe D., "Data Analysis for Database Design", (2nd Ed.), Chapman & Hall 1990,

Background Reading List
C.J. Date,"An Introduction to Database Systems", Addison Wesley (7th Ed.), 2000,
ISBN: 0-201-38590-2
8.6 SENG2300 IS Project Management

Breakdown of Hours per Week:  
ECTS Credits: 20  
Lecture: 2  
Tutorial/Practical: 1  
Self-Study: 2  
Core/Option: C

Course Aims  
Aims:  
• To develop a professional understanding of the functions, scope and interdisciplinary nature of IS project management.

Learning Outcomes  
• To review the stages in a project and its position in the systems development lifecycle.  
• To develop an awareness of project planning and estimating techniques.  
• To display an awareness of the human resources issues appropriate to team building and management; and professional conduct.  
• To understand and develop a practical knowledge of project control, progress monitoring, and reporting.  
• To explain interdisciplinary issues that impact the project manager’s domain.

General Subject Matter  
Project Management is fundamental to the information systems life cycle supply process. Consequently it is appropriate for candidates to have a full understanding of the issues, methods and techniques of IS Project Management early in their careers and studies. This module introduces candidates to the domain of the project manager.

Syllabus  
IS Projects and their stages  
Contractual arrangement, characteristic of projects. Project stages and life cycle processes (requirements elicitation, gathering, analysis, design of software, hardware and networks, build and/or OTS purchase, configuration and integration with current systems, installation issues, methods of going live). The use of post-implementation evaluation. Causes of project failure and areas of risk in each project stage.

Project Planning and Estimating  

Human Resources.  
Team building theory and practice, structures and responsibilities. Skill sets; recruiting, motivating, managing and retaining IS teams. Project management, interpersonal and transferable skills. IS employee Contracts of employment. Confidentiality clauses. Health and Safety issues. Human causes of project failure and areas of risk.

Project Management and Control.  
Teaching Approach:
- Lectures
- Case studies / Practicals
- Tutorials

Assessment Methods:
Written Examination – 70%
Continuous Assessment – 30%

Essential Reading List

Background Reading List
8.7 SENG2301 Professional Issues in IS Practice

Breakdown of Hours per Week:                  ECTS Credits: 10
Lecture: 1                                      Core/Option: C
Tutorial 1
Self-Study: 3

Course Aims
To understand the context of the IS Professional’s domain.

Learning Outcomes
• Understanding of the structure and role of Professional Institutions and codes of conduct and practice appropriate to IS professionals
• Understand the nature and legal standing of a range of organisations
• Understand the range of functions that exist in an organisation, the need for organisational structure and the characteristics of various types of structure
• Understand the principal legislation that applies to the systems profession and recognise situations to which they are relevant

General Subject Matter
This core module provides a broader perspective of the domain of the IS professional and introduce management, ethical, social, legal, and organisational issues. The module provides the students with the core knowledge to complement their practitioner experience to enable them to contribute as IS professionals in the modern workplace.

Syllabus
Professional Institutions
The role of professional institutions and their characteristics: established by Charter, self-governing, controlling entry to the profession and maintaining discipline; reservation of title and reservation of function. Some familiarity with the best-known professional institutions (law, medicine, engineering, computing and accounting).

Organisations and their Structure
Limited companies, Partnership and sole trader and the roles of directors and members; the advantages of limited company status for commercial organisations. The concept of delegation and specialisation. Management structures: structure by function, by product, and by region.

Legal Obligations

Intellectual Property

Professional Codes of conduct and practice
Professional Codes of Conduct and Practice. Their strengths and weaknesses.

Standards
International and de facto standards. The role of professional institutions in defining standards.
Teaching Approach:
• Seminars
• Lectures
• Case studies / Practicals
• Tutorials

Assessment Methods:
Written Examination – 70%
Continuous Assessment – 30%

Essential Reading List

Background Reading List


8.8 PROG2200 Object Oriented Programming

Breakdown of Hours per Week: ECTS Credits: 20
Lecture: 2
Tutorial/Practical: 1 Core/Option: O
Self-Study: 2

Aims
• To understand the concepts and principles behind object-oriented development
• To gain the programming skills required to develop object oriented software.
• To equip students with best practice techniques for object oriented development

Objectives
On completion of this course the student will:
• Understand the concepts of object oriented software development.
• Have a detailed knowledge of the principles of object oriented software development.
• Have a high proficiency in an object oriented programming language
• Be able to implement object oriented software from an object oriented design
• Be able to implement high quality OO software using best practice methods

General Subject Matter
This course gives students the skills and understanding required to program in an object oriented environment using best practice.

Syllabus
Foundations
Introduction to object oriented languages

Concepts
Objects and Classes. Encapsulation. Inheritance: single and multiple. Polymorphism. Overloading and late binding

Principles
Best practice in object oriented design (UML). Data and implementation hiding. Abstraction: generalisation behaviour to higher levels in a classification hierarchy. Reuse. Testing

Language
Classes and Objects, Constructors, Deconstructors. Operator and function overloading. Virtual and friend functions. Inheritance: Single and Multiple. Streams and Files. Exception handling

Teaching Approach:
• Lectures
• Case studies / Practicals
• Tutorials

Assessment Methods:
Written Examination – 70%
Continuous Assessment – 30%

Essential Reading List:
OOTC et al, "Developing Object Oriented Software", Prentice Hall/IBM, 1997,
Background Reading List
Deitel and Deitel, "C++, How to Program (3rd)\textsuperscript{a}," Prentice Hall, 2001.
Siegel S., Object Oriented Software Testing, John Wiley & Sons, 1996,
Stroustrop B., "The C++ Programming Language", Addison-Wesley, 1997,
ISBN: 0-201889-54-4
8.9 TECH2100 Architecture & System Software

Breakdown of Hours per Week: ECTS Credits: 20
Lecture: 2
Tutorial/Practical: 1
Self-Study: 2
Core/Option: 0

Aims:
- To understand the operation of digital computers and the interaction between their hardware and software components
- To understand the principles of digital computer communications
- To understand the principles of operating systems
- To understand the principles of compilers, interpreters and assemblers

Objectives
- Demonstrate an understanding of the operation of digital computers
- To describe facilities of simple operating systems
- To identify issues in memory and peripheral management
- To understand compliers, interpreters, assemblers and loaders
- To evaluate the hardware specification of computer equipment
- To understand the principles of digital communications

General Subject Matter
This module provides the knowledge of computer hardware and digital communications required to understand the operations of computer systems. It also provides an understanding of the ways in which programs work and interact with operating systems and the ways in which programs are prepared for execution.

Syllabus
Data Representation
Number bases and binary arithmetic, codes for data e.g. ASCII, EBCDIC, BCD

Digital Logic
Design of simple combinatorial and sequential logic circuits such as decoders, arithmetic circuits, registers and counters, Integrated circuits, Logic arrays, ASICS and VLSI

Processor Organisation
Structure of CPU, Relationship of hardware to, fetch/execute cycle, Instruction parallelism, concepts of pipelining and superscalar architecture, Performance considerations

Memory Systems
Memory hierarchy, cache and virtual memory

I/O
Interrupts, software polling, Direct memory access, Device interfaces e.g. RS232, I/O controllers

Communications
Principles of communication, Error detection and correction, Local and wide area networks, Communication protocols and standards

High-performance Architectures
Performance metrics e.g. benchmarks, reliability, Technology trends, current limits and future directions, Alternative architectures

**Operating Systems**
Operating system services, Components of operating systems, Interaction of operating system components, CPU utilisation and scheduling

**Use of Memory**
Memory allocation and de-allocation, Multilevel memory management, Control of memory in multi-user systems

**Peripherals**
Peripheral characteristics, Device allocation and management, Device independence and device drivers, File systems requirements: allocation and management

**Language Issues**
Types and levels of languages, Interpretation and translation, Translation of low level languages, Translation of high level languages

**Teaching Approach:**
- Lectures
- Case studies / Practicals
- Tutorials

**Assessment Methods:**
Written Examination – 70%
Continuous Assessment – 30%

**Essential Reading List**

**Background Reading List**
8.10 TECH2300 Networks & Data Transmission

Breakdown of Hours per Week:  
ECTS Credits: 20
Lecture: 2  
Tutorial/Practical: 1  
Self-Study: 2

Core/Option: O

Course Aims
- To develop a detailed understanding of network technologies.
- To have an understanding of local and wide area network technologies, protocols and applications.
- To identify limitations of existing networks and identify advances in technology that may solve them

Learning Outcomes
- To understand layered models especially the OSI model
- To understand the physical properties, performance characteristics of communication media
- To understand the application of existing data communication standards
- To appreciate the theory and practice of common local area networks.
- To show how networks of different standards can be inter-connected.
- To appreciate the significance of network and inter-network protocols especially TCP/IP.
- To understand the importance of network security and reliability, including examples of error recovery strategies and encryption systems.

General Subject Matter
The explosion in the use of networks has made the study of computer networks and the underlying communication technology as important as the more traditional foundations of computer science such as computer architecture, operating systems and programming. This module covers such topics as Digital Communication, Local Area Networks, Wide Area Networks, Inter Networks, Errors, Network Security.

Syllabus
Introduction
Theoretical and practical models of network architecture particularly the ISO OSI and TCP/IP Models. Example networks and services including prototype new technologies. These would include X.25, Frame Relay, ISDN and ATM

Digital Communication
Physical properties of copper media, fibre optics, radio communication, data communication standards. Maximum data rates (theoretical and practical) for different media including some simple analysis of signals. Data encoding of digital signals. The difference between narrow band and broad band technologies with particular reference to ISDN and ATM

Local Area Networks
Types of LAN covering standards, topology and performance. Example architectures such as ethernet and fast ethernet, token ring, FDDI and ATMs. Wireless LANs and emerging technologies. Bridges and routing

Wide Area Networks
Circuit vs packet switching and associated routing and flow control. Detailed examples
of existing and emerging architectures such as frame relay and ATM.

**Inter Networks**
Principles of inter networking, architectures and protocols. Particular reference to TCP/IP

**Errors**
The main causes of errors and the effect on transmission.

**Network Security**
Private and public keys and electronic signatures. Analysis of the effectiveness of algorithms and methods of attack.

**Teaching Approach:**
- Lectures
- Case studies / Practicals
- Tutorials

**Assessment Methods:**
Written Examination – 70%
Continuous Assessment – 30%

**Essential Reading List**

**Background Reading List**
8.11 SENG3100 Software Engineering

Breakdown of Hours per Week: Lecture/Tutorial/Practical: 3 Self Study: 3 ECTS Credits: 20 Core/Option: O

Course Aims:
• to provide the students with an appreciation and understanding of software development as an engineering discipline,
• to provide the student with comprehensive knowledge of the principles, and processes and techniques involved in building high quality software systems,
• to provide the student with a practical knowledge of the various development approaches and their associated techniques

Learning Outcomes:
On completion of the course the student will:
• understand the principles of software engineering,
• have a comprehensive knowledge of the various software process models and development methodologies,
• be able to compare and contrast the difference approaches to systems development,
• be able to demonstrate a practical knowledge of the various techniques of the development of a software system through a development lifecycle
• have a comprehensive knowledge of support activities such as project management, validation and verification, configuration management, quality assurance
• understand the techniques and methods involved in software process improvement

General Subject Matter:
This module develops in the students a theoretical and practical understanding of the software development lifecycle as an engineering discipline.

Syllabus:
Development Methods and Techniques: Structured Methods and Techniques e.g. SSADM Dataflow diagrams, Entity Relationship diagrams, Entity Life Histories, etc.; Object Oriented Methods and Techniques, e.g. UML techniques; Formal Methods, e.g. Z.
Software Development Life Cycle: A comprehensive review of each traditional life cycle phase including requirements analysis and specification, design, implementation, validation and verification, installation and maintenance and each object oriented life cycle phase including inception, elaboration, construction and transition, covering concepts, different techniques used and issues involved in each phase.
Project Management: Risk management, team management, cost estimation, project planning and scheduling, project monitoring and control.
Configuration Management: Configuration identification, the configuration management process, version control, change control, defect tracking, configuration
management software tools,
*Quality Assurance and Process Improvement: Software Quality Assurance, quality factors, software reviews, software process metrics, software process assessment techniques, process improvement methods (e.g. Capability Maturity Model), standards used in process improvement e.g. SPICE*
Advanced Topics: Software Re-engineering, Software engineering for client-server systems and Web-based applications, Computer Aided Software Engineering,

**Teaching Approach:**
- lectures
- case studies
- tutorials

**Assessment Methods:**
- Written Examination – 70%
- Continuous Assessment – 30%

**Essential Reading List:**

**Background Reading List:**
Aims:
• To understand advanced topics relating to the relational model.
• To understand alternate database models and techniques.
• To understand new applications for database systems.

Objectives:
On completion of the course the student will
• Understand advanced implementation aspects of the relational database model
• Understand Object orientated databases.
• Be able to compare and contrast emerging technologies for database systems.
• Understand the limitations of the relational model and understand how alternate
database models and techniques overcome these limitations.
• Understand new database applications.

General Subject Matter:
This course expands on the knowledge gained in SENG2200 by looking at the more
advanced features of the standard relational model and also by examining other models
And current trends in database technology.

Syllabus:
Advanced Implementation aspects of a relational model. Query optimisation,
concurrency control, performance tuning and advanced SQL to support these
implementation issues.

Object Oriented Databases
Features of OO databases, OO database architecture, OO standards, OODBs versus
Relational databases, Strengths & weaknesses of OO databases, Object orientated query
languages

Other Database Technologies:
Distributed databases :- Fundamental principle of Distributed databases, distributed
architecture, issues related to distributed DBMS.
Intelligent databases :- Intelligence in tools and database engine, knowledge based
systems and rule based reasoning.
Multimedia databases :- Multimedia data types, standards for images, video, audio,
performance related techniques for multimedia databases.
New database applications and environments
Data Warehousing :- Data warehousing architecture, data marts, data warehouse
development lifecycle, granularity, scalability and star schema.
On-Line Analytical Processing (OLAP):- Multidimensional data, OLAP tools.
Data Mining:- Data mining techniques, models and methods.
Teaching Approach:
- Lectures
- Case studies / Practicals
- Tutorials

Assessment Methods:
Written Examination – 70%
Continuous Assessment – 30%

Essential Reading List:
ISBN 0-201-38590-2

Background Reading List:
Kim, W., "Modern Database Systems", Addison Wesley, 1995
Hughes J.G., "Object-Oriented Databases", Prentice Hall.
ISBN 0-201-54400-8
8.13 SENG3300 Management Information Systems

Breakdown of Hours per Week:  ECTS Credits: 20
Lecture/Tutorial/Practical: 3  Core/Option: O
Self-Study: 3

Aims:
• To ensure that students have an appreciation of management issues and the role played by IT in the overall strategy of the business.
• To explore the issues surrounding the management of the IT infrastructure to an organisation.

Objectives
• A knowledge of MIS, and its applications
• The application of MIS knowledge to the selection and design of systems appropriate to management requirements.
• An awareness of how MIS may make a contribution to the strategic management of an organisation.
• A knowledge of the management of IT and how the contribution of IT might be maximised.

General Subject Matter
The aim of this course is to ensure that students have an appreciation of management issues and the role played by IT in the overall strategy of the business. The course takes a broader look at IT than the technical perspective. It is also intended to explore the issues surrounding the management of IT development projects and also the provision of an IT infrastructure to an organisation. Decision-makers and business managers cannot make decisions regarding IS/IT in a technological vacuum and it is important for the providers of IT to know about the business and management context within which their systems are developed.

Syllabus
Organisational foundations of MIS
Role of the management function within an organisation. The evolution of the MIS function in an organisation. MIS and decision-making. Strategic management and MIS. Using MIS as a tool for changing the ways of business.

Managing Data Resources

MIS Acquisition

Enterprise information systems

Future Trends

**Teaching Approach:**
- Lectures
- Case studies / Practicals
- Tutorials

**Assessment Methods:**
Written Examination – 70%
Continuous Assessment – 30%

**Essential Reading List**

**Background Reading List**
8.14 SENG3301 E-Commerce

Breakdown of Hours per Week:  
Lecture/Tutorial/Practical: 3  
Self-Study: 3  
ECTS Credits: 20  
Core/Option: O

Aims
- The aim of this course is to explain the underlying strategic management issues that impact E-Commerce and the design and development of solutions for the Internet.

Objectives
- to understand the administration, business and consumer perspectives of E-Commerce
- to create an E-Commerce strategic plan
- to use engineering-based skills for quality Internet software design and development
- to understand Internet and E-Commerce security and legal issues
- to specify and evaluate quality E-Commerce applications

General Subject Matter
The commercialisation of the Internet is revolutionising how business is being conducted, the way consumers buy goods and services and how government interacts with the public. This impacts significantly on the IS professional who must fully understand the consequences of these developments. This module enhances previous learning and equips the student to contribute to an organisation’s E-Commerce strategy.

Syllabus
The fundamentals of electronic commerce.

Business strategy for implementing E-Commerce
Fitting E-Commerce to the enterprise’s IS and business processes. Organisational changes necessary to support E-Commerce (e.g. incident handling, use of smart cards, secure payments, logistics). Funding and monitoring E-Commerce. Evaluation of business requirements. Conducting business transactions over the Internet. E-Commerce marketing strategy: Consumer focus for E-Commerce, brand image, relevance of security, and protection of company reputation. Marketing channels for Web site promotion. The strategic plan.

Legal and security issues of Internet and E-Commerce:

Developing Internet applications:
Quality Web sites: Characteristics that support ease-of-use and corporate superiority (visibility, intelligibility, credibility, engagability and differentiation). The role of standards in E-Commerce: Overview of E-Commerce standardisation activities.

**Impact of E-Commerce on society.**

**Teaching Approach:**
- Lectures
- Case studies / Practicals
- Tutorials

**Assessment Methods:**
- Written Examination – 70%
- Continuous Assessment – 30%

**Essential Reading List**

**Background Reading List**
http://ecommerce.internet.com/opinions/print/0,1282,3551_137791,00.html
http://www.ecml.org/spec.htm
IEEE and ISO standards publications.
8.15 TECH3400  Internet Development

Breakdown of Hours per Week: ECTS Credits: 20
Lecture/Tutorial/Practical: 3
Self-Study: 3
Core/Option: 0

Course Aims
The aims of this course are
• To explain the various technologies that are used in the development of software applications for the Internet,
• to provide students with the skills that are necessary for designing and developing software solutions for the Internet,
• to appreciate the issues associated with Web engineering.

Learning Outcomes
At the end of the course the students will:
• appreciate and be familiar with the variety of services available on the Internet,
• to understand, recognise and have practical experience of a variety of client side and server side technologies available for development of Web applications,
• be able to select, and justify, appropriate technologies for Web application development
• understand the importance and principles of good Web design,

General Subject Matter
This module equips the student with a comprehensive practical knowledge of the technologies in use in the development of Web applications. The student will also gain an understanding of the issues involved in Web engineering and design.

Syllabus
Using the Internet:
Ftp, email, remote login, audio and video across the Internet, the Web, URLs, hypertext, ports, protocols, IP addresses, DNS, domain names, World Wide Web consortium, browsers, search engines,

Client-side technologies:
Mark-up languages, style sheets, helper applications, plugins, applets, Active X, scripting languages, ECMAScript, the Document Object Model,

Server-side technologies:
Common Gateway Interface standard, server-side includes, server APIs, server-side scripting, database and file access.
Comparison of Web servers.

Developing Internet applications:
Web design issues: Information design, navigation design, user profiling, browser incompatibilities.
Associated Internet technologies: Cookies, agents,

Recent advances in technologies.
Comprehensive look at all the latest hot “off the press” technologies.

**Teaching Approach:**
- Lectures
- Practicals
- Tutorials

**Assessment Methods:**
- Written Examination – 70%
- Continuous Assessment – 30%

**Essential Reading List**

**Background Reading List**
Various web resources sites, e.g. http://www.webmonkey.com, http://www.w3.org, etc..
8.16 TECH3500 Distributed Information Systems

Breakdown of Hours per Week: 
Lecture/Tutorial/Practical: 3  
Self-Study: 3

ECTS Credits: 20
Core/Option: 0

Aims
• To gain an understanding of how to propose, develop, manage and review all aspects of distributed information systems.

Objectives
• Be able to develop distributed systems
• Understand information systems and network architectures for specific environments and applications
• Understand selection, deployment and management issues in DIS

General Subject Matter
Historical context of DIS. Advantages and disadvantages of distributed processing systems. Security, data integrity and availability of DIS. Operational network management issues, Local and wide area networks, Messaging and Middleware.

Syllabus
Context of Distributed Information Systems
Development of Internet the progression of NIS from mainframes to mini- and desktop-centred systems.

Distributed processing systems
Distributed processing systems - distributed applications and distributed data, client/server architecture. Building blocks and operating systems. Client/server tools and application development environments

Distributed objects

Security, data integrity and availability
Backup, user access, control, encryption, security certificates, digital signatures, electronic payment systems.

Operational network management issues

Networks
Compare currently available architectures, performance issues, scalability, bridging vs routing, cabling infrastructure, hubs, traffic management Compare different WAN structures.
Teaching Approach:
- Lectures
- Case studies / Practicals
- Tutorials

Assessment Methods:
Written Examination – 70%
Continuous Assessment – 30%

Essential Reading List

Background Reading List
9 Guidelines for Professional Project

Breakdown of Hours per Week:  
ECTS Credits: 30

Tutorial/Practical: 1  
Self-Study: 6  
Core/Option: C

9.1 Introduction  
This section is intended to act as a set of guidelines for staff and students involved in the Professional Projects. It aims to clarify the roles and responsibilities of students, supervisors and project monitors. Where appropriate, guidance is given on time scales and deliverables.

9.2 Aim  
The aim of the final year project is to test the student’s ability to apply the knowledge learned throughout the course. The emphasis for final year projects may be on the development of a software product or on academic research. Topics for final year projects can, therefore, cover a wide variety of areas. Even though these topics cover different areas, the work involved can be sub-divided into a number of generic stages. These stages are:

- Identify a problem area (i.e. project topic) and main objectives of the project.
- Submit a proposal articulating clearly the aims and objectives of the project.
- Research and/or analyse the project area to gain an understanding of the work involved.
- Design, implement and test a solution if appropriate.
- Write a manual which clearly documents the project and, if appropriate, how each of the above stages were carried out.
- Present the project.

9.3 Stages of the project

9.3.1 Identify a project topic  
Students are encouraged to identify their own project topic. Students who have difficulty choosing a topic should communicate this to the project co-ordinator or any member of the computer science staff who will assist them in choosing an area. If a student submits a project topic of their own then the student must write a brief description of the topic and submit this to the project co-ordinator who will assess if the project is feasible. All project students are allocated a suitable supervisor and second reader.
9.3.2 Submit a project proposal
The choice of project may vary widely but in all cases the student should produce a short report (five pages or less) articulating how he/she proposes to undertake the project. This report should be in the following form:

- **Introduction.** Define the context of the project. Assume that the reader has very little knowledge of the subject area. Justify to the reader the value of undertaking the project.

- **Background.** Identify past work in the area if appropriate. Identify key references to other people’s work and how this might relate to the project.

- **Aims and objectives.** Outline the aims of the project from an academic point of view and identify any measurable objectives. Identify the type of project and any particular difficulties envisaged in doing the project.

- **Methodology.** Projects vary widely but in all cases some methodology or formal approach is appropriate. In a research project it is appropriate to identify past work in the area in the form of a literature review. It is also appropriate to provide a project plan of how the student proposes to carry out the project. In the case of a project involving the development of a software system it is appropriate to follow some suitable formal development methodology. It is not intended to provide an exhaustive list of all possible methodologies here but examples such as Object-oriented analysis and design, SSADM, Rapid Application Development, Prototyping, the Spiral model or any of several other methodologies may be considered. Whatever approach is taken should be clearly defined and documented.

- **Deliverables.** A list of deliverables from the project should be provided. Deliverables will vary with project type but the student must plan to be in a position to present documentary evidence of reasonable progress by the last week of the first term. Final deliverables will include a project manual and completed software system if appropriate.

*The student should give a short presentation based on their proposal, which should then be approved by both supervisor and second reader.*

9.3.3 Research / analyse the project topic
During this stage the student is expected to research the project in order to fully understand the domain of the project and appropriate solutions. Depending on the project topic, this research could take the form of detailed system analysis or it could involve pure research in an area. By the end of this stage, the student should be fully acquainted with the project area and have a clear idea of a proposed solution if a software product is to be developed.

Students then present preliminary research findings in the case of a research project or a problem definition and detailed design document in the case of a software system. The purpose of this deliverable is to ensure that the student has made reasonable progress and that any difficulties with the project are identified early. **Students will be expected to give a presentation of progress to date at this stage.**
9.3.4 Design, implement and test a solution for the project topic
If the project involves developing a software system the student must design and document a solution to the problem. The design method used will vary depending on the project area but it should be presented in such a way that it is clear to the reader how the project will be implemented.

Obviously, a clear design will be invaluable to the student for the implementation stage and also will prove invaluable to future students who wish to continue the project. Therefore it is important that some recognised methodology is followed and whatever approach is taken should be clearly defined and documented. It is also appropriate to provide a project plan of how the student proposes to carry out the project.

Depending on the project area, the software product produced may be a full or partial implementation of the design. This stage must contain a programming element. Rapid application development tools may be used only in conjunction with substantial coding. Any exception to this will only be allowed under strict circumstances in consultation with the supervisor and project co-ordinator. The implementation must be tested with appropriate test data. This testing should be carried out as the product is being implemented and at the end of the implementation stage. At the end of this stage, the student should be fully aware of the strengths and weaknesses of their product and these should be documented.

9.3.5 Write a manual
The project manual will be delivered at the end of the project. This should not normally exceed 15,000 words (about 70 pages of 1½ spaced 12-point Times New Roman text). Quality is obviously more important than quantity. The manual should contain the following sections:

Title page. This should contain the title, author’s name, class/subject and date.

Abstract. An abstract should be included immediately following the title page. This should be a summary of the work covered in the project. It should be brief, not more than half a page, but sufficient to indicate the area and the depth of study as well as the conclusions.

Acknowledgements. The assistance of others during the project should be acknowledged.

Table of contents. Following the abstract there should be a table of contents giving headings and page numbers. All headings in the paper should be numbered to two levels maximum (e.g. 1. for level 1, 1.1 for level 2, etc.). A list of figures should also be given.

Introduction. In the introduction explain what the project is about. Then provide the necessary background information, such as the subject area being addressing, the background to, and context of the area. Finally, the reader should be informed how it is proposed to develop the subject under discussion. The introduction provides a broad general view of the subject.
At this stage the student is ‘telling the reader what he/she is going to do’.

**Body of the manual.** The issues outlined in the introduction should be elaborated in the body of the manual. The body should be divided into sections which should concentrate on one main issue. The development should be clear and logical, the reasoning should be clear to the reader. Sub-divisions should be used within each section. Each paragraph should focus on one key message. At the end of each section conclusions should be drawn from the material in that section.

At this stage the student is ‘telling the reader’.

**Conclusion.** The conclusion summarises the discussion in the main body of the manual. The major findings should be commented on and stress given to conclusions of major importance.

At this stage the student is ‘telling the reader what has been done’.

**References.** In the manual, if a diagram, quote or observation is used, then the author of the manual must indicate the source of such material (this will typically be a book or web site). This practice is known as referencing. The importance of referencing is that it tells the reader which parts of the manual are descriptions of previous work and which parts are the student’s own work. It also means that any reader who is interested in reading more about the subject area will know where to get the information. It is important to reference source material in the text. Citations should be used within the text in the form of (Name, Year) e.g. (Bloggs, 1996). A full list of references should be given at the end of the paper in the form of:

Name, Initial, Year, Title of publication, Publisher

e.g. Bloggs, J., 1996, Life the universe and everything, XYZ publishing.

This stage tests the student’s ability to clearly document the work involved in completing the project. The manual also serves as an indication of how well the student understood and was able to work on the project. Each project is corrected by the supervisor, a second reader and an external examiner. The supervisor is the examiner who is most aware of the amount of work completed by the student and so the manual is the student’s record of their work completed during the year. Therefore, it is imperative that the manual is as clear and thorough as possible and should not be written in the last few weeks of the project as an afterthought. Students are advised to write the manual *in conjunction* with carrying out the various stages of the project. Supervisors should encourage students to produce reports at the end of each stage and these reports should be assessed by the supervisor. Students may not plagiarise work from other students or from books. **Therefore, students should reference any external material that they use in their manual.** A bibliography should be included in the manual in an approved format (see above).

**9.3.6 Present the project**

Students give two project presentations during course of the project. The purpose of the first presentation is to measure progress and ensure that any difficulties are highlighted early so that remedial action can be taken.
At the first presentation the student should be able to articulate the preliminary research findings or how the project will work as appropriate. The student must be prepared to answer questions from the supervisor, second reader and project monitors. The student must also present documentary evidence of significant progress to the satisfaction of the project supervisor.

At the second presentation the student must demonstrate their project to their supervisor, second reader and the project monitors. In the case of a software system the student must demonstrate a working system and also be able to verbally articulate how the system works. The student must be prepared for detailed questioning from the supervisor, second readers and project monitors.

### 9.4 Role of the supervisor

The role of the supervisor is to *direct, advise and assess* the student through each stage of the project. The supervisor should meet the student once a week at an agreed time. It is important to remember that the supervisor is not there to do the work for the student but to guide and assess the work as it is being completed. The supervisor also gives technical assistance to the student as required. The supervisor should encourage initiative in the student so that the student learns to take responsibility for his or her own work and does not become overly dependent on the supervisor.

At the start of the project, the supervisor should assist the student in working out a time-scale for the various stages of the project and this should be regularly updated as the project continues. The supervisor should guide the student through each stage of the project and should advise the student on any difficulties he/she may experience. The student should also be regularly updated by the supervisor on their performance. If a supervisor is worried about the performance of a student, this should be communicated to the project co-ordinator so that corrective action can be taken.

Supervisors are expected to be courteous and considerate to their students. If a supervisor has to cancel a meeting, this should be communicated to the student in advance.

### 9.5 Role of the student

The project gives the student the opportunity to apply the skills they have acquired on the course to produce a substantial project. The ultimate responsibility for the completion of the project lies with the student and the project should be the work of the student. In consultation with the supervisor, students are expected to develop an initiative in completing their project and should not depend on the supervisor to actively sort out all of their problems. This means that the project should contain the ideas of the student under the guidance of the supervisor. Project meetings should consist of an exchange of views by student and supervisor and should not just involve the supervisor telling the student what to do and how to do it.

Students are expected to behave with maturity in respect to their supervisors and the project. This means that students should be courteous to their supervisors, accept direction, complete work as required and be punctual for meetings. Supervisors should be notified in advance if meeting has to be cancelled. If a student has any
queries or problems with their project that cannot be rectified by the supervisor, they can communicate this to the project co-ordinator.

9.6 Role of project co-ordinators and monitors

The role of the project co-ordinator is to oversee the management and administration associated with the projects. Any queries or problems experienced by either staff or students should be communicated to the project co-ordinator.

The role of project monitors is to ensure consistency in the marks allocated to projects by examiners. In order to achieve this, monitors randomly attend the demonstrations given by students and quality check the deliverables.

The role of the second reader is to act as a second internal examiner to a project.

9.7 Assessment of work

Projects are assessed under a number of categories, which are outlined below:

Presentation (10%) - the quality of the student’s presentations of the project

Project Report (15%) - the quality of the student’s project report

Research, Analysis and Design (30%) - the extent of the student’s background research and overall understanding of the project subject area; the quality of the student’s analysis of the project; and the quality of the overall and detailed design of the system to be implemented (as applicable)

Project Management (15%) - the overall ability of the student to carry out a project successfully, including the ability to seek advice from others and the ability to report progress to supervisor regularly

Completeness and Complexity (15%) - some projects are by nature technically more complex than others. In such cases, these projects are not required to be as complete as others; this category reflects how complete a particular project is, with respect to the complexity of the project

Achievement (15%) - the value and the usability of the project, including the project report and any software designed and implemented

The project supervisors are required to assess each project under each of these categories. The second assessors are not required to assess each project under Category 4 (Project Management), but are requested to submit an overall mark as a percentage.