**Philadelphia University**

**Faculty of Information technology**

**Department of Computer Information Systems**

# Course Catalogue

**November, 2010**

# CONTENT

# CHAPTER 1

# INTRODUCTION TO CURRICULUM DESIGN

This Handbook contains a set of module descriptions and some information on the curriculum design and organisation that mostly follow the report of the Computing Curricula 2001 project (CC2001-CC2005), particularly (IS 2002: An Update of the Information Systems Model Curriculum). The CC2001 is a joint undertaking of the Computer Society of the Institute for Electrical and Electronic Engineers (IEEE-CS) and the Association for Computing Machinery (ACM) that developed curricular guidelines for undergraduate programs in computing.

These modules are offered at the Department of Computer Information System, Faculty of Information Technology/ Philadelphia University, to obtain the four years B.Sc. (honour) degree in Computer Information Systems (CIS).
The information given in this Handbook is extracted for the Program Specifications for the Degree programme. These specifications are published separately.

## 1.1 Fundamental Concepts

The most important concepts for understanding the module descriptions are as follows:

- **The CIS Body of Knowledge.** The modules described in this Handbook are defined in relation to a general taxonomy of that portion of Computer Information Systems appropriate for an undergraduate curriculum. That taxonomy represents the body of knowledge for Computer Information Systems. The body of knowledge is organised hierarchically into three levels. The highest level of the hierarchy is the **area**, which represents a particular disciplinary sub-field. The areas are broken down into smaller divisions called **units**, which represent individual thematic modules within an area. Each unit is further subdivided into a set of **topics**, which are the lowest level of the hierarchy.
  For coding the modules, the Department of CIS has applied the following scheme of coding. Each area is identified by a one-digit number, such as 1 for Programming Fundamentals or 3 for Architecture / Operating Systems. Each unit is identified by adding a numeric suffix to the area number; as an example, 31 is a unit on Computer Logic Design of the area “Architecture / Operating Systems”.
  The whole computer science areas are listed in Table (1-1).

**Table (1-1) The Areas of Computer Science**

| | |
|---|---|
| 1 | Discrete Structures (DS) |
| 2 | Programming Fundamentals (PF) |
| 3 | Algorithms and Complexity (AL) |
| 4 | Architecture and Organisation (AR) |
| 5 | Operating Systems (OS) |
| 6 | Net-Centric Computing (NC) |
| 7 | Programming Languages (PL) |
| 8 | Human-Computer Interaction (HC) |
| 9 | Graphics and Visual Computing (GV) |
| 10 | Intelligent Systems (IS) |
| 11 | Information Management (IM) |
| 12 | Social and Professional Issues (SP) |
| 13 | Software Engineering (SE) |
| 14. | Computational Science and Numerical Methods (CN) |

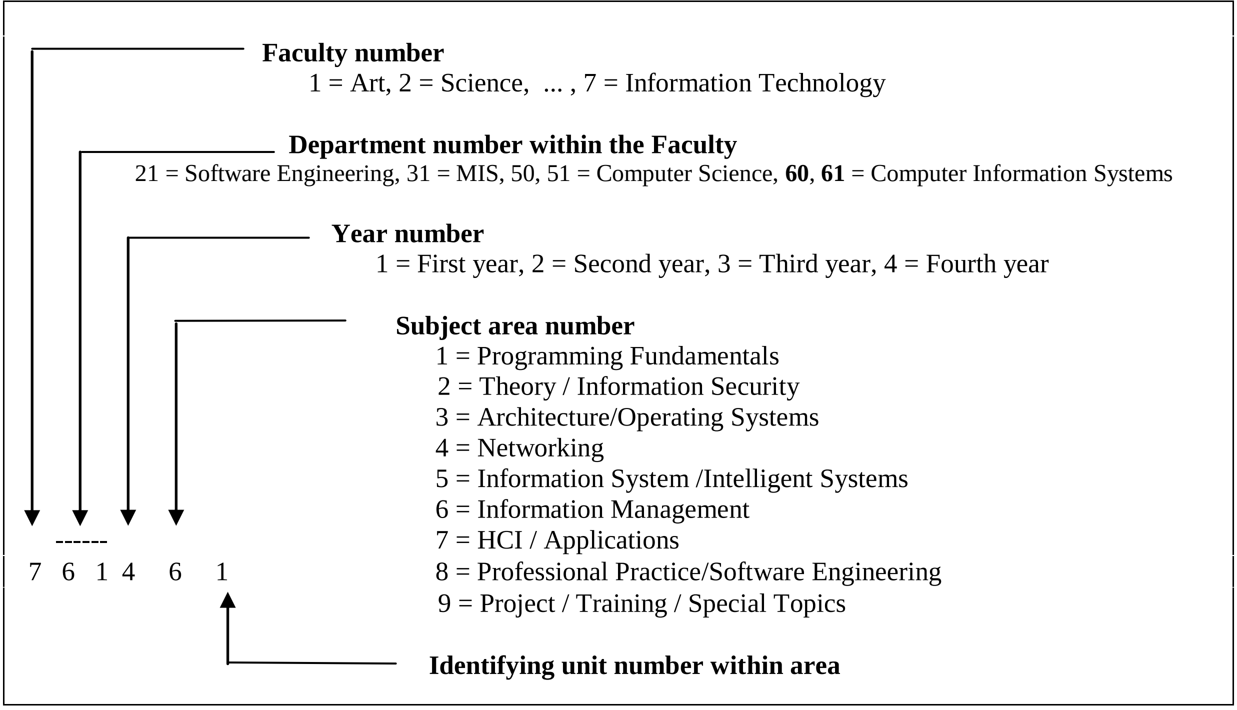
- **Core and Elective Units**. Given the expanding scope of the computing discipline, it is impossible to insist that every undergraduate learn all the topics that were at one time considered fundamental to the field. The **CC2001** report defines a minimal set of core units for which there is a broad consensus that the material is essential to anyone obtaining an undergraduate degree in computer science. Because the core is defined as minimal, the core alone cannot constitute a complete undergraduate curriculum. The undergraduate program must include additional elective units from the body of knowledge. These elective units could be chosen according to the needs of the individual student. Note that, occasionally, timetabling difficulties restricts elective units.

- **Credit Hours.** To give a sense of the time required to cover a particular unit, a time metric should be chosen. The system of study at Philadelphia University is based on the credit hours. The basic measure unit of the curriculum is 3 credit hours module (or course unit). A module, which delivers at least 3 hours per week of lectures or tutorial time, is worth 3 credit hours. Some modules may also provide extra 2-hours per week for laboratory, but the module is still classified as 3 credit hours. In general, over a 16 weeks semester, a typical module provides minimum 50 hours of contact time. The final week of the semester is used for the examinations. The contact time corresponds to the in-class time required to present the material in a traditional lecture oriented format. Note that this time does not include the instructor's preparation time or the time students spend outside of class. As a general guideline, the time required outside of class is twice the time of the in-class time. Thus, a unit that is listed as requiring 3 credit hours will typically entails a total of 9 hours (3 in class and 6 outside). It is also important to keep in mind that the time associated with each unit represents the minimum number of hours required for adequate coverage, and that it is always appropriate to spend more time than the listed minimum.

## 1.2 Format of the Module Coding Adopted

- Each module in the CIS programme is identified by a code and a title. For example, "761462 Information Retrieval" represents a module offered by Faculty of Information Technology, Department of Computer Information Systems in the fourth year, in the area of Information Management, and the module title is Information Retrieval. Figure (1-1) illustrates the scheme of module coding and numbering, where the Information Retrieval module is presented as an example.

**Figure (1-1) Module Coding and Numbering Scheme**



**Faculty number**
1 = Art, 2 = Science, ... , 7 = Information Technology

**Department number within the Faculty**
21 = Software Engineering, 31 = MIS, 50, 51 = Computer Science, **60**, **61** = Computer Information Systems

**Year number**
1 = First year, 2 = Second year, 3 = Third year, 4 = Fourth year

**Subject area number**
1 = Programming Fundamentals
2 = Theory / Information Security
3 = Architecture/Operating Systems
4 = Networking
5 = Information System /Intelligent Systems
6 = Information Management
7 = HCI / Applications
8 = Professional Practice/Software Engineering
9 = Project / Training / Special Topics

**Identifying unit number within area**

# CHAPTER 2

# CURRICULUM DESIGN, ORGANISATION, AND CONTENT

## 2.1 Outlines of the Degree Programme

Within the general area of Computer Information Systems (CIS), the modules recognise several major subject themes. This represents fundamental material on programming, algorithms, software engineering, and information systems, the structure and operation of computer systems including a high-level view of processing, memory, data communication and input/output devices, plus operating systems and compilers, graphics and user interfaces. This includes the theoretical foundations of computing, including programming languages and formal analysis of algorithms and machines.
Details of each module are set out in Chapter (3).

## 2.2 Requirements for the Degree Programme

The CIS programme is covered with different requirements. For obtaining the full award, students should complete 44 modules, each of 3 credit hours, (i.e. a total of 132 credit hours) summarised as follows:

| | | | |
|---|---|---|---|
| - | 9 | modules (University requirements) | 27 hours |
| - | 8 | modules (Faculty requirements) | 24 hours |
| - | 21 | modules (Departmental Compulsories) | 63 hours |
| - | 2 | modules (Departmental Electives) | 6 hours |
| - | 4 | modules (Supportive requirements) | 12 hours |

## 2.3 Design, Organisation, and Content of Curriculum

- ***Organisation of Modules:*** The modules are organised into three levels according to the year at which they occur in the curriculum:
  1- Level 1: **Introductory** modules,
  2- Level 2: **Intermediate** modules,
  3- Level 3: **Advanced** modules.

  Modules designated as **Introductory** are offered in the first and second years of the Department curriculum. Modules listed as **Intermediate** are usually offered in the second or third year and build a foundation for further study in the field. Modules designated as **Advanced** tend to be taken in later years (third and fourth) and focus on those topics that require significant preparation in the earlier coursework. For these modules, the Department wishes to orient such modules to its own areas of expertise**.**

  While these distinctions are easy to understand in their own right, it is important to recognise that there is no necessary relationship between the notions of core and elective - which apply to units in the body of knowledge - and the level of the module. The introductory and intermediate modules concentrate on core material, and the advanced modules include some core material and elective modules.

  The point of organising the modules into three levels: **Introductory**, **Intermediate**, and **Advanced** is to provide natural boundaries for defining implementation strategies. The CC2001 report defined many strategies. Figure (2-1) shows these strategies and their relationship in the curriculum.

**Figure (2-1) Module Levels and Implementation Strategies**

| | | | | | | |
|---|---|---|---|---|---|---|
| **Introductory Modules** | Imperative First | Object First | Functional First | Breadth First | Algorithmic First | Hardware First |
| **Intermediate Modules** | Topic-Based Approach | Compressed Approach | Systems-Based Approach | Web-Based Approach | | |
| **Advanced Modules** | Additional modules used to complete the undergraduate program | | | | | |

- For Introductory Modules, the Department adopted the **Imperative-First (or Procedural-First) strategy**. The imperative language is C++. Then C# is adopted to introduce Object Oriented concepts.
- For Intermediate Modules, the Department adopted **Topic-Based strategy** to preparing for specific areas.
- Some Advanced Modules are selected to attend the departmental objectives and the areas of expertise.

The CIS programme is organised to cover some specified areas selected from the general areas listed in Table (1-1). Table (2-1) shows the areas covered by the specialisation Modules and the number of modules in each of them. Note that the ratios in Table (2-1) are calculated according to the total number of modules (i.e. 44).

**Table (2-1) Specialisation Areas**

| | Area | Compulsory Modules | | Elective Modules | | Total No. of Modules |
|---|---|---|---|---|---|---|
| | | **No** | **(No /44) %** | **No** | **(No /44) %** | |
| 1 | Programming Fundamentals | **5** | 11.36 % | **2** | 4.55 % | 7 |
| 2 | Theory / Information Security | **2** | 4.55 % | **0** | 0 | 2 |
| 3 | Architecture / Operating Systems | **2** | 4.55 % | **0** | 0 | 2 |
| 4 | Networking | **2** | 4.55 % | **1** | 2.27 % | 3 |
| 5 | Information Systems / Intelligent Systems | **3** | 6.81% | **3** | 6.81 % | 6 |
| 6 | Information Management | **6** | 13.63 % | **2** | 4.55 % | 8 |
| 7 | Human-Computer Interaction/Applications | **4** | 9.09 % | **2** | 4.55 % | 6 |
| 8 | Professional Practice/Software Engineering | **3** | 6.81% | **0** | 0 | 3 |
| 9 | Project / Training / Special Topics | **2** | 4.55 % | **1** | 2.27 % | 3 |
| | Total | **29** | **65.90 %** | **Any 2** | **4.55 %** | **31 (70.45%)** |

- ***The Study Plan.*** The whole modules of the curriculum offered by the CIS Department are shown in Appendix A of this Handbook.

- ***The Guidance Plan***. The Department guides students in their registration and selection of modules during the four years. The Department organizes a guidance plan that is shown in Table (2-2), where UR, FR, DR, and SR indicate University Requirements, Faculty Requirements, Department Requirements, and Supportive Requirements, respectively.

## Table (2-2) Guidance Plan for the CIS Curriculum

| Year | Semester | Module Number | Module Title | Prerequisites | Type of Requirements |
|---|---|---|---|---|---|
| First | First (18 Credit Hours) | 110101 | Arabic Language Skills (1) | ---- | (UR) |
| | | 130101 | English Language Skills (1) | ---- | (UR) |
| | | ---- | University Elective (1) | ---- | (UR) |
| | | 750112 | Programming Fundamentals | ---- | (FR) |
| | | 210101 | Mathematics (1) | ---- | (SR) |
| | | 210104 | Discrete Structures | | (SR) |
| | Second (18 Credit Hours) | 111101 | National Education | ---- | (UR) |
| | | ---- | University Elective (2) | ---- | (UR) |
| | | 130102 | English Language Skills (2) | 130101 | (FR) |
| | | 721120 | Object-Oriented Paradigms | 750112 | (FR) |
| | | 731150 | Introduction to Information Systems and Technology | 750112 | (FR) |
| | | 210103 | Mathematics for Computing | 210101 | (SR) |
| Second | First (18 Credit Hours) | ---- | University Elective (3) | ---- | (UR) |
| | | 731270 | Introduction to Web Programming | 750112 | (FR) |
| | | 761211 | Windows Programming | 721120 | (FR) |
| | | 210231 | Introduction to Statistics and Probabilities | ---- | (SR) |
| | | 721210 | Introduction to Software Engineering | 731150 | (DR) |
| | | 721221 | Object Oriented Data Structures | 721120+210104 | (DR) |
| | Second (18 Credit Hours) | 111100 | Military Sciences | ---- | (UR) |
| | | ---- | University Elective (4) | ---- | (UR) |
| | | 721240 | Computing Ethics | 731150 | (FR) |
| | | 761272 | Multimedia Systems | ---- | (FR) |
| | | 750232 | Computer Architecture | 731150 | (DR) |
| | | 760261 | Database Fundamentals | 721221 | (DR) |
| Third | First (18 Credit Hours) | ---- | University Elective (5) | ---- | (UR) |
| | | 731331 | Database Applications | 760261 | (DR) |
| | | 731332 | Systems Analysis and Design | 760261 (or concurrently) | (DR) |
| | | 750322 | Design and Analysis of Algorithms | 210103 + 721221 | (DR) |
| | | 750333 | Principles of Operating Systems | 750232 | (DR) |
| | | 750351 | Fundamentals of Artificial Intelligence | 721221 | (DR) |
| | Second (15 Credit Hours) | 721331 | Software Projects Management | 731332 | (DR) |
| | | 731351 | Information Systems Modelling | 731332 | (DR) |
| | | 760398 | Practical Training | Dept. Agrt. | (DR) |
| | | 761340 | Fundamentals of Computer Networks | 721221 | (DR) |
| | | 761373 | E-Commerce Applications | 731270 | (DR) |
| Fourth | First (12 Credit Hours) | 760421 | Information Security | 750333 | (DR) |
| | | 760463 | Data Warehousing and Data Mining | 731331 | (DR) |
| | | 761443 | Wireless and Mobile Computing | 761340 | (DR) |
| | | 761462 | Information Retrieval | 760261 | (DR) |
| | | ---- | Department Elective (1) | ---- | (DR) |
| | Second (15 Credit Hours) | 760499 | Research Project | 760398 | (DR) |
| | | 761442 | Advanced Web Programming | 731270 | (DR) |
| | | 761467 | Information Systems Testing | 731351 | (DR) |
| | | ---- | Department Elective (2) | ---- | (DR) |

# CHAPTER 3

# FULL DESCRIPTION OF MODULES

This chapter presents the full description of the Department modules and those modules from the Faculty and University requirements that are computer-oriented modules.

## 3.1 Module Descriptor

The Department organised a format for the module descriptor that includes much information on the module. This sub-section presents the components of the adopted module descriptor that are shown in Figure (3-1). The University Quality Assurance Handbook explains in details the components of the module descriptor.

**Figure (3-1) Components of the Module Description**

**Module Number, Module Title**
**Module Coordinator(s):**
**Year:**
**Credit:**
**Prerequisites:** Required modules or background
**Module Aims:**
**Teaching Methods:**
**Learning Outcomes:**
**Assessment of Learning Outcomes:**
**Contribution to Programme Learning Outcome:**
**Syllabus:** A list providing an outline of the topics covered.
**Modes of Assessment:**
**Textbook and Supporting Materials:**

## 3.2 Introductory Modules

Table (3-1) presents the Introductory (Level 1) modules whose full descriptions are given below.

**Table (3-1) Introductory Modules in Computer Information Systems Department**

| Module Number | Module Title | Prerequisite |
|---|---|---|
| 210101 | Mathematics (1) | None |
| 210103 | Mathematics for Computing | 210101 |
| 210104 | Discrete Structures | None |
| 210231 | Introduction to Statistics and Probabilities | None |
| 721120 | Object-Oriented Paradigms | 750112 |
| 721210 | Introduction to Software Engineering | 731150 |
| 731150 | Introduction to Information Systems and Technology | 750112 |
| 731270 | Introduction to Web Programming | 750112 |
| 750112 | Programming Fundamentals | None |
| 761211 | Windows Programming | 721120 |
| 761272 | Multimedia Systems | None |

## 210101, Mathematics (1)

3 hours per week, 3 credit hours, prerequisite: **none**

***Teaching Methods:*** 32 hours Lectures (2 per week) + 16 hours Tutorials (1 per week)

***Aims:*** This module aims to provide students with some background in different topics in mathematics such as derivatives, applications of derivatives, integrals, and applications of integrals, transcendental functions and inverses of functions

***Learning Outcomes:***
On completing this module, student should
1- Have an understanding of the basic topics of calculus such as derivatives and integrals
2- Have an understanding of the concepts of applications of derivatives and integrals.
3- Apply the principles of calculus in some real world problems.
4- Be familiar with the concepts of transcendental functions and inverse functions.

***Assessment of Learning Outcomes***
All learning outcomes are assessed by two tests during the semester and final examination, and by coursework.

***Contribution to Program Learning Outcomes***:
A1, B1

***Synopsis:*** General Introduction: (Inequalities, functions);Limits and continuity; differentiation: (rate of change, chain rule, implicit differentiation); the mean value theorem:(maxima and minima, applications, concavity, curve sketching); Integration: (the fundamental theorem of calculus, change of variables, applications (area, motion, solids of revolution); the transcendental functions: (differentiation and integration) .

***Textbooks and Supporting Material:***
1- Salas, Hille, and Etgen, Calculus, 9th Edition, Wiley, 2002
2- Howard Anton, Calculus, Wiley, 2002

***Modes of Assessment:*** Two 1-hour midterm exams (15% each); Assignments (10%); Tutorial Contribution (10%); 2-hours Final Exam (50%)
……………………………………………………………………………………………………..

## 210103, Mathematics for Computing

3 hours per week, 3 credit hours, prerequisite: **210101**

***Teaching Methods:*** 32 hours lectures (2 hours per week) + 16 hours Tutorials (1 per week).

***Aims:*** This module aims to provide students with some background in sequences and series, and multi-variable calculus, and systems of linear equations and their solutions.

***Learning Outcomes:***
On completing this module, student should
1- Have an understanding of the basic topic of Sequences and infinite series.
2- Have an understanding of the concepts of general vector spaces.
3- Be able to apply the principles of calculus in some real world problems.
4- Be able to apply the concepts of linear algebra in simple experiments,

***Assessment of Learning Outcomes***
All learning outcomes are assessed by two tests during the semester and final examination, and by coursework.

***Contribution to Program Learning Outcomes***:
A1, B1

***Synopsis:*** Infinite series (convergence, Taylor series, power series), vector Calculus (functions of several variables, partial Derivatives; double and triple integral over a region), vectors, Linear algebra, Linear equations, Gaussian elimination, Eigen values and Eigenvectors, introduction to Linear transformation).

***Textbooks and Supporting Material:***

1- Salas and Hille, Calculus, 7th Edition, Wiley, 1995
2- Howard Anton, Elementary Linear Algebra, 9th Edition, Wiley, 2000
3- Silverman, R.A., Calculus With Analytic Geometry, Wiley, 1998

***Modes of Assessment:*** Two 1-hour midterm exams (15% each); Assignments (10%); Tutorial Contribution (10%); 2-hours Final Exam (50%)
…………………………………………………………………………………………………………..

## 210104, Discrete Structures

3 hours per week, 3 credit hours, prerequisite: **None**

***Teaching Methods:*** 32 hours Lectures (2 per week) + 16 hours Tutorials (1 per week)

***Aims:*** This module will introduce the student to the basic language and ideas of discrete mathematics that occur in all branches of information technology. It will also begin the process of training the student to argue correctly, both informally and formally, about these structures. The student will begin to learn the use of abstract analysis to solve concrete problems.

***Learning Outcomes:***
On completing this module, student should

1. be familiar with the idea of a discrete structure, and the notions of formal language and parse tree.
2. have an understanding of the basic ideas of sets and functions, including Boolean combination of sets, and be able to manipulate such expressions
3. have an understanding of the standard propositional logic connectives and be able to convert logical expressions into conjunctive and disjunctive normal form.
4. have an understanding of the universal and existential quantifiers
5. be familiar with the general concept of binary relation, equivalence and order relations and methods of combining relations; be familiar with the standard graphical representations of relations,
6. be familiar with the principle of mathematical induction and be able to perform proofs using this principle, also be aware of simple examples of structural induction on lists.
7. be able to apply the inclusion-exclusion principle in simple counting examples,
8. be familiar with the basic ideas of probability, and be able to calculate probabilities in simple experiments.

***Assessment of Learning Outcomes***
All learning outcomes are assessed by two tests during the semester and final examination, and by coursework.

***Contribution to Programme Learning Outcomes:***
A1, B1, D4, D6

***Synopses:*** **Arithmetic:** The standard discrete number systems and the arithmetical operations on them with their properties; **Sets and Functions:** Standard set and function notation and terminology. Boolean operations on sets. Injective and surjective functions. Composition of functions; **Logic:** The connectives (or, and, not, implies, if and only if), Formulae of propositional logic, Truth tables, Tautologies and logical equivalence, Normal forms, The quantifiers (for all, there exists); **Binary Relations:** Definitions and examples, Properties of relations, Digraphs and representations of relations, Equivalence relations and Partitions, Combining relations and closure operators, Order relations, **Recurrence Relations:** Construction an solutions; **Induction:** The principle of mathematical induction, with many examples. Structural induction; **Combinatory:** Inclusion Exclusion principle, Binomial coefficients and permutations, Pascal's triangle. Summing series involving binomial coefficients.

***Textbooks and Supporting Material:***

1- TRUSS, J.K. Discrete Mathematics for Computer Scientists. (ISBN 0-201-175-649) 2nd Edition, Addison Wesley 1998.

There is not a book, which covers exactly the material in this module. The above book covers a large part of the module but also contains additional material, some of which is covered in later modules.

There are many books on discrete mathematics, which have useful features. For example

1- MATTSON, H.F. Discrete Mathematics with Applications (ISBN 0-471-599-662), Wiley 1993.
2- GARNIER, R. and TAYLOR, J. Discrete Mathematics for New Technology (ISBN 0-750-301-35X) Institute of Physics Publishing 1992.
3- ECCLES, P.J, An Introduction to Mathematical Reasoning, (ISBN 0-521-59718-8) C.U.P. 1997.

***Modes of Assessment:*** Two 1-hour midterm exams (15% each); Coursework (15%); Tutorial Contribution (5%); Final (unseen) 2-hour examination (50%)

……………………………………………………………………………………………………..

## 210231, Introduction to Statistics and Probabilities

3 hours per week, 3 credit hours, prerequisite: **none**

***Teaching Methods:*** 32 hours Lectures (2 per week) + 16 hours Tutorials (1 per week)

***Aims:*** This module aims to help students grasp basic statistical techniques and concepts, and to present real-life opportunities for applying them.

***Learning Outcomes:***

At the end of this module, the student will:

- Have an understanding of the range of discrete mathematics and applications.
- Be able to read and construct mathematical proofs in the right logical sequence.
- Be able to apply techniques of counting in combinatorial problems.
- Be able to make conversion between different base number systems.
- Understand concepts of graphs and be able to solve algorithmic problems.

***Assessment of Learning Outcomes***

All learning outcomes are assessed by two tests during the semester and final examination, and by coursework

***Contribution to Program Learning Outcomes:***

A1, B1

***Synopsis:*** Descriptive statistics and probability distribution; Sampling distribution Estimation for the mean, variance and proportions; Testing for the mean, variance and proportions; Regression and correlation; One-way analysis of variance.

***Textbooks and Supporting Material:***

1- D.C. Montgomery and. G.C. Runger, Applied Statistics and Probability For Engineers, 2nd Edition, Wiley, 2002.

2- William, Probability and Statistics in Engineering and Management, Wiley, 2002.

***Modes of Assessment:*** Two 1-hour midterm exams (15% each); Assignments/Quizzes (10%); Tutorial Contribution (10%); 2-hours Final Exam (50%).

………………………………………………………………………………………………………..

## 721120 Object Oriented Paradigms

5 hours per week, 3 credit hours, prerequisite: **750112**

***Teaching Methods:*** 32 hours lectures (2 hours per week) + 16 hours Tutorials (1 per week) + 32 hours Laboratory (2 per week)

***Aims:*** This module provides the student with a framework for thinking about object-oriented concepts and the main focus is the general object-oriented and programming concepts from a software engineering perspective. It aims to develop an understanding of the principles of the object-oriented paradigm; provide familiarity with approaches to object-oriented modelling and design; provide a familiarity with the syntax, class hierarchy, environment and simple application construction for an object-oriented programming language.

***Learning Outcomes:***
On completion of this module, student will:

- Acquire a full Object Oriented Thinking
- Have a clear understanding of the object-oriented concepts such as objects, classes, inheritance, and polymorphism.
- Have an informal understanding of the operational semantics of object-oriented programs in terms of creation of objects and messages passing between them.
- Be able to design small object oriented programs which meet requirements expressed in English, with a strong software engineering foundation
- Have knowledge of Object Oriented Design guidelines.
- Be able to code small software systems in Java language.
- Be able to maintain large, high-quality software systems
- Developing different projects using java programming language.

***Synopsis:*** A brief survey of programming paradigms. Object-oriented programming: Object-oriented design; encapsulation and information-hiding; separation of behavior and implementation; classes, subclasses, and inheritance; polymorphism; class hierarchies, Fundamentals of event-driven programming, Application and applet programs, Building GUIs.

***Textbooks and Supporting Material:***
1- Dietel and Dietel, Java How to Program, 3rd ed., Prentice Hall, 2000 (or later)
2- C. Thomas Wu: "An Introduction to Object-Oriented Programming with Java", 2nd edition, published by McGraw-Hill, 2001, ISBN 0-07-118195-4.
3- Ben Shneiderman, Designing the User Interface: Strategies for Effective Human-Computer Interaction, (3rd edition) Addison-Wesley, 1998.

***Modes of Assessment:*** Two midterm exams (15% each), Laboratory (15%), Tutorial contribution (5%), Final exam (50%).

………………………………………………………………………………………………………..

## 721210 Introductions to Software Engineering

3 hours per week, 3 credit hours, prerequisite: **731150**

***Teaching Methods:*** 38 hours Lectures (2-3 hours per week) + 10 hours Tutorials (0-11 per week)

***Aims:*** This module aims to provide students a comprehensive introduction to software engineering. It gives an introduction to basic concepts, principles and techniques used in software engineering. It discusses the nature of software and software projects, software development models, software process maturity, project planning, management, and communication. This module gives an introduction to methods for analysis, design, testing, and implementation of large, complex software systems.

***Learning Outcomes:***
On completion of this module, student should:

- Understand a wide range of principles and tools available to the software engineer such as specification, design, coding and testing methodologies, and user interface techniques.
- Design software systems of small size through academic and realistic case studies (tutorials).

***Synopsis:*** Introduction to Software Engineering: Motivation, Basic concepts (Life cycle, phases, software quality, process, method, methodology, mission and roles of phases, Abstraction, encapsulation, information hiding, etc.); The Software Process: Software Process, Life Cycles, Process Tools; The Software Project Management: Scheduling, Personnel, Estimation, Risk Management, Managing for Results; Software Requirements: Representing Requirements, Prototyping, Documentation, Participants, Requirements Analysis and Specification; Introduction to Formal Specification; Software analysis and design: Structured analysis and design (DFD, ERD, Decision tables), Object oriented analysis and design (Class diagram, State diagram, etc.), User Interface Design; Software Construction: Automatic implementation generation from design model, Programming with error avoidance, Programming with error tolerance, defensive programming, Programming for and with reuse, Programming Database applications with encapsulating blocks; Verification and validation: Relation to software quality - Software metrics, Unit Testing, Integration Testing & OO Testing, Test Planning System Testing, Test Management; Introduction to Maintenance: Introduction to Software configurations management, Introduction to Software Reengineering, Quality Assurance and risk management, Software engineering principles: tool, technique, method, methodology, process. Lifecycle models; sizing, estimation, planning and control; Support techniques: object-oriented paradigm, software database, parallel and distributed programming, Requirements specification; design; implementation, Integration and testing strategies; quality assurance; configuration management; Software maintenance; Reuse and reengineering.

***Textbooks and Supporting Material:***

1- Ian Sommerville**,** Software Engineering 7th, Addison Wesley Longman, Inc. August 2004
2- D. C. Ince, An introduction to discrete mathematics, Formal systems, and Z. Ed Clarendon Press, Oxford. 1992
3- D. F. D'Souza, Objects, Components and frameworks with UML, Addison Wesley, 1998
4- S. Benett, Object-Oriented System Analysis and Design, Ed Mc Graw Hill, 1999
5- R. S. Pressman Software Engineering: A Practitioner's Approach, 5th Edition, McGraw Hill; 2001

***Modes of Assessment:*** Two 1-hour midterm exams (15% each) + Assignments (15%) + Tutorial contributions (5%) + 2-hours final exam (50%).

…………………………………………………………………………………………………..

## 731150 Introduction to Information Systems and Information Technology

3 hours per week, 3 credit hours, prerequisite: **750112**

***Teaching Methods:*** 48 hours Lectures (3 hours per week)

***Aims:*** This module provides an introduction to information systems and information technology, information systems development concepts, and application software. It identifies the basic types of business information systems, the major steps of the systems development process and some of the strategies employed to lower costs or improve service. It explains how information is used in organizations and how IT enables improvement in quality, timeliness, and competitive advantage. It also defines the competitive advantages, types of roles, functions, and careers available in IS.

***Learning Outcomes:***

On completion of this module, student will

1. Have a wide range of principles and fundamentals of Information Systems and Information Technology.
2. Understand the application of IS and IT.
3. Understand basic analytical steps of Information Systems and be able to defining the specifications of the IT required in business contexts.
4. Plan and undertake a small individual project in IS and IT fields
5. Use the scientific literature effectively and make discriminating use of Web resources.
6. Present seminars in IS and IT fields.
7. Use appropriate computer-based tools.
8. Work effectively with and for others.
9. Strike the balance between self-reliance and seeking help when necessary in new situations.
10. Get a knowledge about self learning on the long run

***Synopsis:*** Introduction to Information Systems and Information Technology in organizations; Types of information systems; Competitive advantage; Computer hardware: CPU components, memory types and characteristics, input devices, output devices, and strategic hardware issues; Computer software: system software, application software, software issues, programming languages, enterprise software; Organizational data and information; Data Management; Data modelling; Database models; Functions of DBMS; Telecommunications, the internet, intranets, and extranets; Internet services; Net issues, management issues, service bottlenecks, privacy and security; An introduction to E-Commerce; E-Commerce Applications; E-Commerce technology components; Strategies for successful E-commerce; Transaction processing systems. Enterprise resource planning; Decision making and problem solving; MIS, Inputs and outputs of MIS; DSS, GDSS, ESS; Artificial intelligence, nature of intelligence, difference between natural and artificial intelligence; Expert systems, components of ES, ES development, applications of ES. Virtual reality; interface devices; Systems development life cycle. Prototyping, RAD, CASE; Phases of SDLC; System investigation; System analysis; System design; System implementation; System maintenance.

***Textbooks and Supporting Material:***

1. Fundamentals of Information Systems, Editors: Ralph M. Stair, George W. Reynolds, Course Technology; Published: 2001, Thomson Learning, Inc.
2. E. Turban, R. K. Rainer and R. E. Potter, Introduction to Information Technology, John Wiley, 2001.
3. Gerald M. Weinberg, an Introduction to General System Thinking, Silver Anniversary Edition, 2001.
4. Leonard M. Jessup and Josef S. Valacich, Information Systems Foundations, 1999, Que E&T.
5. James A. O'Brien, Introduction to Information Systems: Essentials for the e-Business Enterprise. 11th ed. 2003, McGraw-Hill Higher Education.
6. Steven Alter, Information Systems a management perspective, 2nd ed. 1996, The Benjamin/Cummings Publishing Company, Inc.
7. Senn, James A, Analysis and design of information systems, 1989, New York: McGraw-Hill
8. Thomas L. Floyed Digital Fundamentals, Pearson Education 2000.
9. Thomas C. Bartee, Computer Architecture and Logic Design, McGraw Hill 1991.

**Journals**

1. Peter Sappal, "E-Commerce Recruitment" the wall street journal, January 25,1999.p52.
2. David Penn, "Ericsson Screen Phone Runs Linux," Linux journal, February 29, 2000.

**Websites**

**1. www.macnn.com**

2. **www.applix.com**
3. **www.computerworld.com**
4. **www.allina.com**

***Modes of Assessment:*** Two 1-hour midterm exams (20% each) + Assignments (10%) + 2-hours final exam (50%).

……………………………………………………………………………………………………..

## 731270 Introductions to Web Programming

3 hours per week, 3 credit hours, prerequisite: **750112**

***Teaching Methods:*** 35 hours Lectures (3 hours per week) + 13 hours Optional Tutorials/Lectures (average 1 per week) + 24 hours Laboratories

***Aims:*** This module aims to give students an introduction and general concepts of the Internet and Intranet technology, the World Wide Web, TCP/IP and Web design languages (HTML, CSS, _JavaScript, and ASP). It also involves the necessary background that student needs to develop different tasks of programming aspects concerning the foregoing objectives. Sufficient study levels are supposed to be studied and learned by the students within the course for the sake of applying the different fields of education, learning, economical, E-Business and other approaches.

***Learning Outcomes:***
A student completing this module should be able to:

1. Understand the Internet and Intranet technology, the Web concepts (TCP/IP: Architecture & Protocols, Web Servers, Web Sites, DNS, and IP Addresses).
2. Design Web pages using HTML, XHTML and CSS.
3. Develop Web Sites using JavaScript language and the most structures fitting the problem under design.
4. Build Web Servers using ASP.Net

***Synopsis:*** Introduction to the Internet technology; TCP/IP: Architecture & Protocols (Client/Server) DNS, ISP; Internet Services: Email, WWW, Usenet News, Telnet, Web Servers, Search Engines; Introduction to Hyper Text Markup Languages HTML: Editing HTML, HTML Tags, Headers; HTML Tags: Text Styling & Formatting, Links, Multimedia, Images and Image Map; HTML: Lists, Tables, Frames; Advanced HTML Concepts; HTML Applications; HTML and XHTML; Cascading Style Sheets (CSS); JavaScript: Introduction, Arrays, Objects, Events; Rich Internet Applications Client/Server Technologies: Adobe Flash, ASP.NET(introduction) and applications

***Textbooks and Supporting Material:***
1- Deitel & Deitel, Internet & World Wide Web How to Program, Prentice Hall, 3rd Edition, 2004
2- Wendy Willard, HTML: A Beginner's Guide, McGraw Hill, 3rd Edition, 2007
3- Douglas Comer, “Computer Networks & Internets”, Prentice Hall, 2003
4- HTML for fun and profits. Prentice Hall, 1999.

**Websites**
www.w3schools.com
www.webteacher.org
www.microsoft.com
www.whatis.com
www.idocs.org
www.w3.org
www.webdeveloper.com
www.javascriptmall.com
www.javascripts.com/toc.cfm

***Models of Assessment:*** Two 1 – hour midterm exams (20% each) + Coursework (10%); Final written Examination (50%)

………………………………………………………………………………………………………..

## 750112, Programming Fundamentals

3 hours per week, 3 credit hours, prerequisite: **None**

***Teaching Methods:*** 32 hours Lectures (2 hours per week) + 16 hours Tutorials (1 per week) + 32 hours Laboratories (2 per week)

***Aims:*** This module aims to introduce computer programming and emphasis in problem solving on the fundamentals of structured design using the principles of Top Down problem solving strategy (divide and conquer). This includes development, testing, implementation, documentation. The module also aims to explore the logic of programming via the algorithm concepts and implement them in programming structures including functions, arrays, strings, and pointers.

***Learning Outcomes:***
On completion of this module, student will:
1- Understand the fundamental programming constructs.
2- Understand and write searching and sorting techniques.
3- Understand a typical C-like program environment.
4- Be able to understand and analysis any problem and derive its solution.
5- Be able to develop algorithms.
6- Be able to work as a team
7- Be able to write C-like programs including searching and sorting techniques.

***Synopsis: Problem Solving***: process, Analyze (requirement, Design algorithm, Tracing algorithm, Example, Design problems); ***Problem Analysis***: Algorithm discovery, Algorithm design strategies**,** Stepwise refinement**,** Control requirements; Implementing algorithm; ***Data Definition Structures***: Types, constants, variables, Expressions: Arithmetic, Logical; Precedence rules; ***Control Structures***: Sequencing, Input and output statements, Assignment statement, Selection: one-way (if .. then), two-way (if .. then .. else), multiple (switch), Repetition (while structure, do .. while structure, for structure); ***Functions:*** Parameters definition and passing (functions depth look), prototypes, Parameters definition and passing (Scope: local and global variables); ***Data Structures:*** One and two dimensional arrays; ***Abstract data type:*** Records **(struct** definition statement); Strings (use of main operations: Concatenate, string copy, compare, etc.); ***Strings; Files*** (use of main operations of a sequential file: open, reset, rewrite, read, write, eof); **Pointers**

***Textbooks and Supporting Material:***

1- D.S. Malik, Thomson, C++ Programming: From Problem Analysis to Program Design, Third Edition, Course Technology, 2007
2- Maureen Sprankle, Problem Solving and Programming Concepts, 7/E, Prentice Hall, ISBN: 0131194593, 2006
3- Friedman Frank and Koffman Elliot B., "*Problem Solving, Abstraction and Design using C++"*, Addison Wesley, Fourth Edition. 2004
4- Jeri R. Hanly and Elliot B. Koffman, Problem Solving and Program Design in C, Pearson Education, Inc., ISBN: 0-321-21055-7
5- Deitel & Deitel, C++ How to Program, Prentice-Hall, 2001.
6- A. Lambert Kenneth and Nance Douglas W., "*Understanding Programming and Problem Solving With C++"*, PWS Publishing Compny, Fourth Edition. 1996
7- Forouzan, B. A. & R. F. Gilberg. *"Computer Science: A Structured Programming Approach using C"*, Second Edition, Pacific Grove, CA: Brooks/Cole, 2001
8- Bruce Eckel, *"Thinking in C++*", Second Edition, Prentice Hall, 2000.
9- Herbert Schildt, *"Teach Yourself C++*", Third Edition, McGraw-Hill. 1998

**Website(s):**

- www.cee.hw.zc.uk/~pjbk/pathways/cpp1/cpp1.html
- www.edm2.com/0507/introcpp1.html
- www.doc.ic.ac.uk/~wjk/C++intro
- www.cprogramming.com/tutorial.html
- www.cs.umd.edu/users/cml/cstyle/ellemtel-rules.html
- www.deakin.edu.au/~agoodman/Ctutorial.html
- www.tldp.org/howto/c++programming.howto.html
- www.vb-bookmark.com/cpptutorial.html

***Models of Assessment:*** Two 1 – hour midterm exams (15% each) + Coursework (30%); Final written Examination (40%)

…………………………………………………………………………………………………..

## 761211, Windows Programming

3 hours per week, 3 credit hours, prerequisite: **721120**

***Teaching Methods:*** 32 hours lectures (2 hours per week) + 16 hours Tutorials (1 per week) + 32 hours Laboratory (2 per week)

***Aims:*** This module aims to provide students capabilities to design and implement the applications using visual programming through Microsoft Visual Studio .Net and VC# to develop different types of applications using .Net platform**.**

***Learning Outcomes:***
On completion of this module, student will:
1- To be familiar with event driven programming and windows programming concepts (A)
2- To understand GUI programming using .NET platform (A)
3- To be able to develop applications and simple event-driven programs using C# language in the .NET framework.
4- To be able to use 2D and 3D graphics with lines, curves and can implement algorithms to rasterizing shapes and implement them
5- To have the ability to design and implement Group projects (B)
6- To have practical skills in the usage of .NET environment (C)
7- To be able to use C# build-in functions in developing a wide range of applications and graphical tools

***Synopsis****:* Introducing the Microsoft, .NET Platform, Visual Studio.NET IDE, Introducing C# Programming, Object oriented programming concepts, Windows Forms, Graphics, Multimedia, ADO.NET, Multithreading, Networking, ASP.NET

***Textbooks and Supporting Material:***
1- H. M. Deitel & J. Deitel, “C# How to Program”, Prentice Hall, 2005.
2. A.Turtschi et.al. “Mastering Visual C# .Net”, Sybex 2002
3. Eric Gunnerson, “A Programmer’s Introduction to C#”, Apress 2000
4. Anders Hejlsberg et.al., “ C# Language Reference”, Microsoft Corporation 2000.
5. Erric Buttow et al., “C#, your visual blueprint for building .Net application”, Hungry Minds 2002
6. Charles Carroll, “Programming C#”, O’Reily & Associates 2000
7. Karh Watson, “Beginning C#”, Wrox Press 2001.

**Websites**
www.deitel.com
www.csharp.com

***Modes of Assessment:*** Two 1-hour midterm exams (15% each); Assignments (20%); 2-hours Final Exam (50%).

……………………………………………………………………………………………………..

## 761272, Multimedia Systems

3 hours per week, 3 credit hours, prerequisite: **none**

***Teaching Methods:*** 40 hours lectures (2 hours per week) + 8 hours Tutorials (1 per 2weeks) + 32 hours Laboratory (2 per week)

***Aims:*** This module is an introduction to the major topics related to multimedia (desktop publishing, hypermedia, presentation media, graphics, animation, sound, video, and integrated authoring techniques), multimedia devices and development tools. It emphasizes hands-on experience for students to familiarize them with the range of tools used in creating computer-based multimedia.

***Learning Outcomes:***

On completion of this module, student will:

1- Understand basic multimedia concepts.

2- Acquire basic knowledge on Multimedia devices.

3- Understand current trends in multimedia by experiencing a variety of applications and development packages.

4- Understand the preproduction process including content acquisition and development, process flow, team management and integration, and legal issues surrounding multimedia

5- Demonstrate technical knowledge and limited proficiency in designing production elements in each of the multimedia disciplines.

6- Be able to design different application in M.M and use different tools like macromedia flash and mat lab to express image processing

***Synopsis**:* Introduction to Multimedia: Basic concepts, Applications (video on demand, Videoconferencing, virtual learning, entertainment, games, simulations, virtual reality…), Multimedia Hardware, Multimedia Software Tools (Overview on current available tools), Desktop Publishing, Graphics, Pictures: graphic modes and formats, still pictures and format (JPEG…), User Interface Design and Graphics: Graphic Elements and user interface considerations (Backgrounds, buttons, presentation elements), Production Planning and Design: (Research, content flow, Content acquisition, Multimedia team management using project management software, Budgeting considerations, Element and resource lists), Audio and Sound, Analogue Video (1), Digital video (2), Animation, Authoring, Hypermedia Authoring: Authoring: Web Based Multimedia, Multimedia Compression: Overview on techniques and standards.

***Textbooks and Supporting Material:***

1. Vaughan, Tay, Multimedia: Making it work, Berkeley Osborne McGraw-Hill, 7th Edition 2008.
2. Stephen McLoughlin, Multimedia: Concepts and Practice, Prentice hall, 2001
3. Ralf Steinmetz & Klara Nahrstedt, Multimedia fundamentals Volume 1: Media coding and content processing, Prentice-Hall, 2002.
4. Ze-Nian Li & Mark S Drew, Fundamentals of Multimedia, Prentice hall, 2004.
5. Jen Dehaan, Macromedia FLASH MX 2004, training from the source, Macromedia press, 2004.

**Multimedia Software Packages:**

Macromedia Flash (2004), Photoshop and mat lab Macromedia flash 2004

**Journals**

1- British Journal of Educational Technology ISSN 0007-1013

Cognitive style, gender and learning from multi-media materials in 11-year-old children pp. 43-56(14) **Authors:** *Riding R.; Grimley M.*

2- MultiMedia Forum: an Interactive Online Journal ,Germany

**Websites**

1- http://highered.mcgraw-hill.com/sites/0072230002/information_center_view0/
2- www.emeraldinsight.com/

***Modes of Assessment:*** Two 1-hour midterm exams (15% each); Assignments (20%); 2-hours Final Exam (50%).

……………………………………………………………………………………………………

## 3.3 Intermediate Modules

The Intermediate (Level 2) modules are listed in Table (3-2) and their full descriptions are given below.

**Table (3-2) Intermediate Modules in Computer Science Department**

| Module Number | Module Title | Prerequisite |
|---|---|---|
| 721221 | Object Oriented Data Structures | 721120+210104 |
| 721240 | Computing Ethics | 731150 |
| 721331 | Software Projects Management | 731332 |
| 731331 | Database Applications | 760261 |
| 731332 | Systems Analysis and Design | 760261 (or concurrently) |
| 731351 | Information Systems Modelling | 731332 |
| 750232 | Computer Architecture | 731150 |
| 750322 | Design and Analysis of Algorithms | 210103 + 721221 |
| 750333 | Principles of Operating Systems | 750232 |
| 750351 | Fundamentals of Artificial Intelligence | 721221 |
| 760261 | Database Fundamentals | 721221 |
| 760398 | Practical Training | Dept. Agrt |
| 761340 | Fundamentals of Computer Networks | 721221 |
| 761373 | E-Commerce Applications | 731270 |

## 721221 Object Oriented Data Structures

3 hours per week, 3 credit hours, prerequisite: 721120+210104

***Teaching Methods:*** 32 hours lectures (2 per week) + 16 hours Tutorials (1 per week) + 32 hours Laboratory (2 per week)

***Aims:*** This is a **programming-intensive** module where students learn the fundamentals of designing data structures for use in complex programs. Data structures module is an essential area of study for computer scientists and for anyone who will ever undertake any serious programming task. This module deals with the fundamentals of organizing and manipulating data efficiently using clean conceptual models. Students study many of the important conceptual data types, their realization through implementation, and analysis of their efficiency. Implementations in this module are carried out in the Java programming language, but the principles are more generally applicable to most modern programming environments.

Topics include recursion, the underlying philosophy of object-oriented programming, fundamental data structures (including stacks, queues, linked lists, hash tables, trees, and graphs), and the basics of algorithmic analysis.

***Learning Outcomes:***
On completion of this module, student will:

1- obtain a firm understanding of advanced data structures including when and how they should be used, how they are implemented, and the performance trade-offs for choosing one data structure over another,
2- Understand big-oh notation and what it means
3- Develop further proficiency in the Java programming language

***Synopsis:*** Introduction to Software Engineering, Data Design and Implementation, Abstract Data Types (ADTs); Algorithm complexity; Lists ADTs: static implementation and dynamic implementation, single linked list, doubly linked list and circular linked list; Stacks: Static implementation and dynamic implementation, Programming with Recursion; Queues: Static implementation and dynamic implementation, circular queue; Trees: Binary search tree, binary expression tree, and heap tree, Priority Queues and Heaps; Graph ADT; Sorting : Bubble sort, selection sort, insertion sort, Shell sort, Quick sort, Heap sort; Searching: Sequential search, Binary Search; Hashing : hash function, Separate chaining, open addressing

***Textbooks and Supporting Material:***
1- Nell Dale, Daniel T. Joyce and Chip Weems, Object-Oriented Data Structures using Java.
2- Goodrich and Tamassia, *Data Structures and Algorithms in Java*, *2nd edition*, John Wiley and Sons, 2000.
3- Arnold, Gosling, and Holmes, *The Java Programming Language, 3rd edition,* Addison-Wesley, 2000.
4. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", Addison-Wesley, 1999

**Websites**
Htpp:// oodatastructures.jbub.com

***Modes of Assessment:*** Two 1-hour midterm exams (15% each); Assignments (20%); 2-hours Final Exam (50%).
……………………………………………………………………………………………………..

## 721240 Computing Ethics

3 hours per week, 3 credit hours, prerequisite: **731150**

***Teaching Methods:*** 32 hours lectures (2 per week) + 16 hours Tutorials (1 per week) + 32 hours Laboratory (2 per week)

***Aims:*** This module will develop the ethical foundations of good professional practice in computing and will give students an informed awareness of the principal issues of ethics and professional responsibility in the development and use of computers and information systems. It will provide a basic survey of ethical theories and discuss the role of professional organizations in maintaining good practice, both in general and then specifically in the computing industry. It will also consider legislation that applies in the computing industry, including three major areas of ethical concern in computing: computer cracking, data privacy and intellectual property of software.

***Learning Outcomes:***
On completion of this module, student will:

1- Be able to better understand the difference between what is ethical and what is legal
2- Have a basic knowledge of IPR in relation to Copyright and Patents
3- Be aware of health and safety issues in IT products
4- Have improved knowledge and some experience of group working and distributed enterprises
5- Have acquired some basic discussion skills
6- Be aware of the requirements for professionalism in respect of the work of the professional societies and their codes of conduct and practice.

7- Satisfy the requirements for ACM/IEEE accreditation in respect of Ethical and Professional Issues
8- Have an appreciation of basic legal processes and of computer misuse, fraud and the law as it stands in the world with respect to IT
9- Have an appreciation of the law in relation to contracts and of safety-critical systems and legal liability.

***Synopsis:*** Basic and support material to be covered; Information Technology Changes; Impacts of IT Changes; Introduction to Ethics; Ethics Philosophical Issues; Intellectual Property Rights; Computer Crimes; Computer Crimes; Privacy; Evaluating and Controlling Technology; Errors, Failures, and Risks; Work and Wealth; Professional Ethics; Professional Ethics and Responsibilities; Code of Conduct

***Textbooks and Supporting Material:***
1- Michael J. Quinn, Ethics for the Information Age, 3rd Ed., Addison-Wesley 2009.
2- Gorge Reynoids, Ethics in Information Technology, Thomason, 2003.
3- Sara Baase, A Gift of Fire: Social, Legal and Ethical Issues for Computer and the Internet, 2nd ed., 2003.
4- Tavani H. T. and Hoboken N. J., Ethics and Technology, John Wiley, 3rd Ed, 2004.
5- Deborah G. Johnson, Computer Ethics. 3rd Edition, Englewood Cliffs, N.J., Prentice Hall, 2001.

**Journals**

مجموعة تشريعات الملكية الفكرية الأردنية

**Websites**

Center for Computing and Social Responsibility (CCSR):
http://www.ccsr.cms.dmu.ac.uk/

Computer Professionals for Social Responsibility (CPSR):
http://www.cpsr.org/

***ACM, IEEE and BCS Web Sites.***
www.cyberethics.cbi.msstste.edu
www.aitp.org
www.acm.org
www.prenhall.com
www.jcs.rg.jo

***Modes of Assessment:*** Two 1-hour midterm exams (15% each); Assignments (20%); 2-hours Final Exam (50%)

……………………………………………………………………………………………………..

## 721331 Software Project Management

3 hours per week, 3 credit hours, prerequisite: 731332

***Teaching Methods:*** 32 hours lectures (2 per week) + 16 hours Tutorials (1 per week) + 16 hours Laboratory (1 per week)

***Aims:*** Software management is concerned with knowledge about the planning, organization, and monitoring of all software life-cycle phases. Management is critical to ensure that : software development projects are appropriate to an organization, work in different organizational units is coordinated, software versions and configurations are maintained, resources are available when necessary, project work is divided appropriately, communication is facilitated, and progress is accurately charted.

***Learning Outcomes:***

On completion of this module, student will:

1- Understand A wide range of principles and tools available to the software engineer and software manager, such as planning, organization, and monitoring of all software life-cycle phases. All these direction informed by research. (A)
2- Understand the professional and ethical responsibilities of the practicing computer professional including understanding the need for quality. (A)
3- Understand the application of computing in a business context (A)
4- Solve a wide range of problems related to the software management (B)
5- Be able to manage small size software. (B)
6- Be able to design, write and debug software management tools in appropriate languages. (C)
7- Be able to plan and undertake a major individual project, and prepare and deliver coherent and structured verbal and written technical report. (C)
8- Be able to display an integrated approach to the deployment of communication skills, use IT skills and display mature computer literacy, strike the balance between self-reliance and seeking help when necessary in new situations, and display personal responsibility by working to multiple deadlines in complex activities.

***Synopsis:*** Software Process overview; Management concepts: General project management, Classic management models, Project management roles, Enterprise/Organizational management structure, Software management types (e.g. acquisition, project, development, maintenance, risk, etc.); Project planning: Evaluation and planning, Work breakdown structure, Task scheduling, Effort estimation, Resource allocation, Risk management; Project personnel and organization: Organizational structures, positions, responsibilities, and authority, Formal/informal communication, Project staffing, Personnel training, career development, and evaluation, Meeting management, Building and motivating teams, Conflict resolution; Project control: Change control, Monitoring and reporting, Measurement and analysis of results, Software Quality, Correction and recovery; Software configuration management: Revision control, Release management, Tool support, Software configuration management processes, Maintenance issues, Distribution and backup

***Textbooks and Supporting Material:***

1- R. S. Pressman, Software Engineering, A Practitioner's Approach, 6e, 2005

**Journals**

Boehm, B. (1987) A sprial model of software development and enhancement, *Software Engineering Project Management*, p.128–142.

**Websites**

1. http://www.engineering.wright.edu/~rrea/468.html
2. www.mhhe..com/pressman
3. http://highered.mcgraw-hill.com/sites/0077109899/information_center_view0/

***Modes of Assessment:*** Two 1-hour midterm exams (20% each); Assignments (10%); 2-hours Final Exam (50%)

……………………………………………………………………………………………………..

## 731331 Database Applications

3 hours per week, 3 credit hours, prerequisite: **760261**

***Teaching methods:*** 35 hours Lectures (2 per week) + 13 hours Optional Tutorials/Lectures (average 1 per week)

***Aims*:** This module tends to give the student knowledge in database systems programming using Oracle. This includes the different types of SQL statement, PL/SQL blocks, Cursors, Stored and user-defined Procedures, stored and user-defined Functions, Packages. The module also focuses on the Graphical user interface programming tools within Oracle especially Oracle Developer, and Oracle Report Writer.

***Learning Outcomes:***

On completion of this module, student will:

1. Understand the theoretical concepts in relational database design and analysis
2. Determine the limitations of relational databases.
3. Determine the limitations of relational databases.
4. Analyze, design, and implement an intermediate software system using Oracle tools
5. Be able to prepare and present seminars, lectures and others.
6. Work effectively with others.

***Synopsis:***

***Textbooks and Supporting Material:***

1- Kevin Owens, Programming Oracle, 3$^{d}$ edition, The Prentice Hall, 2004.

***Modes of Assessment:*** Two 1-hour midterm exams (20% each); Assignments (10%); 2-hours Final Exam (50%)

…………………………………………………………………………………………………………..

## 731332 Systems Analysis and Design

3 hours per week, 3 credit hours, prerequisite: **760261**

***Teaching Methods*:** 32 hours Lectures (2 per week) + 10 hours Tutorials (on average 1 per week) + 6 hours seminars (last 2 weeks)

***Aims:*** This module provides a methodical approach to developing computer systems including systems planning, analysis, design, testing, implementation and software maintenance. Emphasis is on the strategies and techniques of systems analysis and design for producing logical methodologies for dealing with complexity in the development of information systems. The module approaches the development of information systems from a problem-solving perspective. This course builds upon concepts to which the student has been exposed in previous classes.

***Learning Outcomes:***

At the end of this module, student will be able to:

1- Understand A wide range of principles and tools available to the software developer and information system developer in particular, such as data bases, Analysis and methodologies, and user interface techniques. (A)

2- Understand the professional and ethical responsibilities of the practicing computer professional including understanding the need for quality. (A)

3- Understand the application of computing in a business context (A)

4- Solve a wide range of problems related to the analysis, design and construction of information systems. (B)

5- Analysis and Design of system of small size. (B)

6- Plan and undertake a major individual project, and prepare and deliver coherent and structured verbal and written technical report. (C)

7- Be able to design, write and debug computer programs in appropriate languages. (C)

8- Be able to display an integrated approach to the deployment of communication skills, use IT skills and display mature computer literacy; strike the balance between self-reliance and seeking help when necessary in new situations, and display personal responsibility by working to multiple deadlines in complex activities. (C)

***Synopsis:*** System Analysis Fundamentals: Introducing SA&D, SA&D concepts, Roles of system analyst; The system development life cycle, Using Case tools; Depicting system graphically, determining feasibility, activity planning and control; Information requirement analysis: Sampling and investigating data, interviewing, Using questionnaires; Prototyping; The analysis process: Using data flow diagram, Using data dictionaries, Describing process specifications and structured decisions.

The system proposal; The essentials of design: designing output, designing input, Designing the file or database, Designing the user interface, Designing data-entry procedures, Documenting the design phase; Quality assurance through software engineering; Implementing the information system; Object oriented SA&D; Submitting and discussing the projects

***Textbooks and Supporting Material:***
**1-** Kenneth E. Kendall Julie E. Kendall, Systems Analysis and Design, 5th Edition, Prentice Hall, 2001

***Modes of Assessment*****:** Two 1 – hour midterm exams (20% each) + Assignments (10%) + Finals written Examination (50%)
…………………………………………………………………………………………………

## 731351 Information Systems Modelling

3 hours per week, 3 credit hours, prerequisite: **731332**

***Teaching Methods:*** 32 hours Lectures (2 per week) + 10 hours Tutorials (on average 1 per week) + 6 hours seminars (last 2 weeks)

***Teaching Methods:*** 30 hours Lectures (2 per week) + 10 hours Laboratory **+** 5 hours Tutorials

***Aims:*** The concepts of Information Systems Modelling. Topics covered include the principles of Information System Modelling, techniques for modelling, Object Oriented Modelling, and the diagrams used in modelling including Use Case, Class Diagram, Sequence Diagram, Activity diagram and others. Data Modelling is also discussed. Students will use the object oriented modelling in implementing a real world information systems. Different tools will be used in this course including SmartDraw and Rational

***Learning outcomes:***
On completion of this module, student will be able to:
1- provide a brief introduction to general issues of Information Systems Modelling
2- provide students with a clear understanding of the different modelling techniques.
3- introduce students to the role of information systems modelling in information systems development
4- explain the stages and process different modelling techniques.
5- understand the strengths and weaknesses of particular information systems modelling techniques
6- learn IS Modelling techniques through the use of different tools (e.g. SmartDraw and Rational)

***Synopsis:***

***Textbooks and Supporting Material:***
1- Grady Booch, Rumbaugh, Jacobson, The Unified Modelling Language User Guide, Addison Wesley, Last Edition.

***Modes of Assessment*****:** Two 1 – hour midterm exams (20% each) + Assignments (10%) + Finals written Examination (50%)
…………………………………………………………………………………………………

## 750232, Computer Architecture

3 hours per week, 3 credit hours, prerequisite: 731150

***Teaching Methods:*** 32 hours Lectures (2 per week) + 12 hours Tutorials (0-1 per week) + 4 hours Seminars/Presentations

***Aims:*** The module will emphasize on the following knowledge areas: assembly level machine organization, memory system organization and architecture, interfacing and communication, functional organization, and alternative architectures.

***Learning Outcomes:***
A student completing this module should:

1. Know what actions are taken at the machine level during the user's efforts for running a code written in high level language. (A)
2. Know what micro-actions are taken within a CPU during the machine's efforts for running a machine level code. (A)
3. Know the basic structure of a typical RISC and CISC processor. (A)
4. Understand how memory hierarchy and pipelining affect the performance of a processor. (A)
5. Understand the communication (input/output) issues.
6. Know the common blocks required in a typical computer system.
7. Be a knowledgeable consumer when it comes to the selection of appropriate computer hardware. (B)
8. Be able to prepare and deliver a written report. (C)

***Synopsis:*** Review of Basic Computer Architecture and Microprocessors; Von Neumann architecture: principles, instruction sets, instruction format, addressing modes, assembly/machine language programming, CISC versus RISC architectures, subroutine call and return mechanism; Control unit: hardwired, micro-programmed; Storage system and their technology: memory hierarchy, main memory organization and operations, cycle time, bandwidth and interleaving; cache memory: addressing mapping, block size, replacement and store policy; virtual memory: page table , TLB; I/O fundamentals: handshaking, buffering, programmed I/O, interrupts-driven I/O; Buses: types, bus protocols, arbitration, Direct Access Memory; Pipelining: principles, Instruction pipelines, Pipelines difficulties and solutions; Introduction to SIMD, MIMD.

***Textbooks and Supporting Material:***
1- Patterson, D. A. and Hennessy, J. L. Computer Organization and Design: The Hardware/ Software Interface. 2nd Edition, (ISBN 1-558-604-91X), Morgan Kaufmann 1998
2- William Stallings, Computer Architecture & Organization: Design for Performance, Prentice Hall, 2000
3- J. Van de Goor, Computer Architecture and Design , 1989.

***Modes of Assessment:*** Two midterm exams (15% each); Course work (10%); Seminars (5%); Tutorial Contribution (5%); Final Exam (50%)
……………………………………………………………………………………………………..

## 750322, Design and Analysis of Algorithms

3 hours per week, 3 credit hours, prerequisite: **210103 + 721221**

***Teaching Methods:*** 38 hours Lectures (2 per week) + 7 hours Tutorials (1 per 2 weeks) + 3 hours Seminars (project presentation)

***Aims:*** The aim of this module is to learn how to develop efficient algorithms for simple computational tasks and reasoning about the correctness of them. Through the complexity measures, different range of behaviours of algorithms and the notion of tractable and intractable problems will be understood. The module introduces formal techniques to support the design and analysis of algorithms, focusing on both the underlying mathematical theory and practical considerations of efficiency. Topics include asymptotic complexity bounds, techniques of analysis, and algorithmic strategies.

***Learning Outcomes:***
On completion of this module, students should be able to:

1. Understand basic ideas about algorithms

2. Understand the concepts of time and space complexity, worst case, average case and best case complexities and the big-O notation
3. Understand the range of behaviors of algorithms and the notion of tractable and intractable problems
4. Know and understanding a wide range of searching and sorting algorithms
5. Develop efficient algorithms for simple computational tasks (B)
6. Reason about the correctness of algorithms (B)
7. Compute complexity measures of algorithms, including recursive algorithms using recurrence relations (B)
8. Able to present projects

***Synopsis:*** Introduction, Algorithm definition, Algorithm Analysis; Mathematical Induction; Summation Techniques; Recurrence Relations; Design & Analysis of Algorithms: Divide and conquer, Greedy Algorithm, Dynamic Programming, Backtracking, Branch-Bound; Lower Bound Theory; Sorting and Searching; NP-Complete Problems: Basic Concepts, NP-Hard & NP-Complete Problem

***Textbooks and Supporting Material:***

1- Anany Levitin, Introduction to the Design and Analysis of Algorithms, Pearson Education, 2007
2- Sara Baase, Computer Algorithms: Introduction to Design and Analysis, Third Edition, Addison-Wesley, 2000.
3- E. Horowitz, et.al., Fundamentals of Computer Algorithms, Galgotia Publications, 1998.
4- Udi Manber, Introduction to Algorithms: a Creative Approach, Addison-Wesley, 1997.
5- T. Cormen, et.al., Introduction to Algorithms, 1999.
6- R. Sedgewick, Algorithms in C++, 2002.

***Modes of Assessment:*** Two 1-hour midterm exams (15% each) + Tutorial contributions (5%) + Coursework (15%) + Final written Examination (50%)

……………………………………………………………………………………………………

## 750333, Principles of Operating Systems

3 hours per week, 3 credit hours, prerequisite: **750232**

***Teaching Methods:*** 40 hours Lectures (2-3 per week) + 8 hours Tutorials (1 each fortnight)

***Aims:*** The aims of this module are to introduce the basic principles of computer systems organization and operation; to show how hardware is controlled by program at the hardware/software interface; to outline the basic OS resource management functions: memory, file, device (I/O), process management, and OS security/protection. Two concrete examples of operating systems are used to illustrate how principles and techniques are deployed in practice.

***Learning Outcomes:***

On completion of this module, students should:

1- Have knowledge and understanding of the overall structure and functionality of a modern operating system and of its interactions with the underlying computer hardware and overlying user-program. (A)
2- Have knowledge and understanding of the operation of the following major components of an operating system: the I/O device manager; the memory manager; the process manager; the file manager; OS security/protection manager (A)
3- Have the ability to design and implement (an emulation of) a prototypical process manager. (B, C)
4- Be aware of how fundamental techniques in (1) and (2) are applied in practice in two distinct modern operating systems. (A)

***Synopsis****:* Operating System overview; Operating System Structures: System components, Operating system services, System calls, System structures, Virtual machine; Processes: Process concept, Process scheduling, Operation on process, Cooperative process, Inter process communication; Threads: Thread overview, Benefits, User and kernel threads, Multithreading model, Solaris 2 threads; CPU Scheduling: Basic concept, Scheduling criteria, Scheduling algorithm, Thread scheduling, Algorithm evaluation; Process synchronization and mutual exclusion: Critical section problem, Two task solution, Synchronization hardware, Semaphore, Classical synchronization problem; Deadlock and starvation: System model, Deadlock characterization, Method for handling deadlock, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock; Memory management: Background, Swapping, Paging, Virtual memory, Background, Demand paging, Page replacement, Allocation of frame, Thrashing; File system implementation and management: File concept, Access method, Directory structure, Protection, File system structure, Allocation method, Free space management, Directory implementation, Efficiency and performance, I/O management and disk scheduling, Application I/O interface, Kernel I/O subsystem, I/O request handling, Disk structure, Disk scheduling, Disk management, Swap space management, Disk reliability, Stable storage implementation

***Textbooks and Supporting Material:***

1- A. Silberschatz and Peter Galvin, Applied Operating Systems Concepts, First edition, John Wiley & sons, Inc, 2000

2- J. Bacon, Concurrent Systems: Database and Distributed Systems, 2$^{nd}$ Edition, (ISBN 0-201-177-676), Addison Wesley, 1998.

3- A. S. Tanenbaum, Modern Operating Systems, Prentice Hall, 1992

***Modes of Assessment:*** Two 1-hour midterm exams (15% each); Assignments (10%); Lab work (5%); Tutorial contribution (5%); 2-hours Final Examination (50%)

……………………………………………………………………………………………………..

## 750351, Fundamentals of Artificial Intelligence

3 hours per week, 3 credit hours, prerequisite: **721221**

***Teaching Methods:*** 24 hours In-class Lectures (1-2 per week) + 16 hour E-learning Lectures (1 per week) + 8 hours Tutorials (1 per 2 weeks)

***Aims:*** The module is the primary introduction to artificial intelligence. Halve of the module material is delivered in-class and the other halve is distant learning using the e-learning module designed at faculty of IT in Philadelphia University. The module aims to present the basic representation and reasoning paradigms used in AI in both theory and practice with careful attention to the underlying principles of logic, search, and probability. It is also designed to show students practical examples of the use of AI in applications and to encourage further reading. The e-learning part enables students to practice self learning. The Assignments aim to give students a sound practical introduction to knowledge based systems and a basic introduction to modern paradigms of knowledge representation and belief networks. The examples classes aim to provide an introduction to the underlying issues in cognitive emulation and to provide an opportunity for practical exercises in logic and probability.

***Learning Outcomes:***

On completion of this module, student will

1- have an understanding of search, logic based knowledge representation, of issues in planning and learning. (A, D)

2- have an understanding of the limitations of current symbolic AI paradigm (A).

3- be able to select appropriate search paradigms for appropriate problems (A, B)

4- have knowledge of Bayes' Rule and its use in Belief Networks and be able to solve problems concerning updating of prior probabilities with evidence using it and to construct belief networks for simple problems. (A., B)

5- be able to design a simple agent system and associated ontology and justify the design (B)

6- be able to design and implement a forward chaining knowledge based system including rule base (C)

7- be able to study on-line (B, C).

***Synopsis***: Introduction to AI (what is AI? foundations of AI); Intelligent agents (What is an agent? structure of agents); Intelligent agents (types of agents, environments); Problem Solving (search algorithms, understand the search problems and their algorithms); Problem Solving (introduce search algorithms, uninformed search algorithms); Problem Solving (iterative deepening search, informed search algorithms); Problem Solving (best-first search, A* search algorithm); Problem Solving (admissibility and dominance, simulated annealing search); Knowledge representation (Introduction, history of knowledge representation, semantic networks); Knowledge representation (frames, scripts, conceptual graphs, and conceptual dependency); Knowledge representation (production rules, logic knowledge representation, propositional logic); Knowledge representation (first-order logic, inference rules in first-order logic, Prolog and Lisp); Expert System (Introduction, components of an expert system); Expert System (compare between human thinking and computer thinking, rules based systems); Expert System (programs required to develop an expert system, types of expert systems). Expert System (examples of well known expert systems, strategies in expert systems, develop an expert system).

***Textbooks and Supporting Material:***

1- S.J. Russell & P. Norvig, Artificial Intelligence: A Modern Approach, 2nd edition, Prentice Hall, 2002.
2- G.F. Luger & W.A. Stubblefield, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, 3rd edition, Addison Wesley, 1998.
3- N. J. Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann Publishers, 1998.
4- P.H. Winston, Artificial Intelligence, 3rd edition, Addison Wesley, 1992.
5- E. Rich, K. Knight, Artificial Intelligence, 2nd edition, McGraw Hill, 1991.
6- Charniak, D. McDermott, Introduction to Artificial Intelligence, Addison Wesley, 1985.

***Modes of Assessment:*** Two 1-hour midterm exams (20% each); Assignments (5%); Tutorial contributions (5%); Final Examination: 2-hours written exam (40%) + defended project (10%).

…………………………………………………………………………………………………..

## 760261, Database Fundamentals

3 hours per week, 3 credit hours, prerequisite: **721221**

***Teaching Methods*:** 26 hours Lectures (average 2 per week) + 16 hours Laboratory (1 per week) + 6 hours Tutorials (1 each fortnight)

***Aims:*** This module aims to give the students the main concepts of database, design the database, database models, normalization techniques, query languages, object oriented database, query optimization and database and the web. Further the students have to practice and write some applications regarding the database.

***Learning Outcomes:***

On completion of this module, student should be able to:
1- Define the general concepts, objectives and the database models
2- Understand the importance of data, and the difference between file management and databases. (A)
3- Understand the design of database management system architectures and environments. (A)
4- Know the principals of database design. (A)
5- Design a database as free-standing applications
6- Apply conceptual design methodologies, in particular conceptual design using Extended Entity Relationship modelling. (A, B, C, D)
7- Apply the relational model and mappings from conceptual designs, in particular normalizations. (A, B, C, D)
8- Explain physical and performance related design considerations. (A)
9- Understand transaction processing. (A)
10- Discuss and apply SQL and the Oracle DBMS. (A, C, D)
11- Work effectively with others.

12- Invoke the database applications with the World-Wide Web browser

***Assessment of Learning Outcomes:***
Learning outcomes (1) through (7) are assessed by examinations. Learning outcomes (3), (4), and (8) are assessed by projects design and implementation.

***Contribution to Programme Learning Outcomes:***
A2, A3, A4, A5, B1, B2, B3, C1, C2, C6, D1, D3

***Synopsis:*** Introduction to Data base and DBMS; Database Models; Database Design; Relational Algebra and Relational Calculus; Query Languages (SQL); DB normalization; Database Integrity and Security; Indexing Techniques; Query Optimization; Distributed Data Base; Object-Oriented Database

***Textbooks and Supporting Material:***
1- A. Silberschatz, H.F. Korth & S. Sudarshan, Database System Concepts, McGraw Hill, 2002
2- El-Masri & Navathe, Fundamentals of Database Systems, Prentice Hall, 2002.
3- Jeffry Ullman, Principles of Database Systems, SU Publishers, 1999
4- C.J. Date, An Introduction to Database Systems, Addison Wesley, 1995

***Modes of Assessment:*** Two 1-hour midterm exams (15% each); Lab work (10%); Assignments (10%); Final Examination: written exam (50%)
……………………………………………………………………………………………………..

## 760398, Practical Training

3 hours per week, 3 credit hours, prerequisite: **Department agreement**
(Students can take this module on completing 90 credit hours at least).

***Aims:*** The main aim of this module is that students will have practice in different industrial, commercial, administrative enterprises or companies. By this module, students may apply, in the real world, what they have learned during the first three years of their study in the University. The module also aims to teach students how to be self-confident when they face problems in their practical life.

***Duration:*** At least 9 weeks (18 training hours per week at least). This may be distributed onto two semesters at most.

***Regulations for Training:*** Students who register on practical training module should not register on modules with total credit hours more than 15 hours per week including the training module itself. Students must, therefore, be full-time trainees for at least 2 days per week. Students should arrange their timetable for other modules in a way that enables them to enroll in the pre-specified enterprise or company at least two days per week during the semester period.

***Modes of Assessment:*** A committee from the Department supervises the students along their training period, where one supervisor is assigned on one group of students. The student should submit a technical report to this committee in 2 weeks time after completing the training session. In addition, the trainer body presents a report to the committee. The grade "pass" is given to students who complete the training requirements successfully and discuss their reports with the supervision committee.
……………………………………………………………………………………………………..

## 761340 Fundamentals of Computer Networks

3 hours per week, 3 credit hours, prerequisite: **721221**

***Teaching Methods:*** 40 hours Lectures (2-3 per week) + 8 hours Tutorials

***Aims:*** The aims of this module are to introduce the basic principles of computer network and how the network is working. Type of the topologies and the technologies are also given. The module, however, does not focus on the detailed study of mathematical aspects. Bottom up is the approach used to teach this module.

***Learning Outcomes:***

On completion of this module, students will be able to

1. Discriminate and appraise different switching technologies
2. Enumerate, differentiate and appraise different WAN and LAN networks
3. Breakdown communication networks into their components
4. Explain the abstract concepts related to layered communication system architecture
5. Select an appropriate communication network architecture
6. Select and devise an appropriate network interconnection solution
7. Outline the communication software implementation issues
8. Configure the network devices like switch and router

***Contribution to Programme Learning Outcomes:***

A2, A3, A5, B1, B2, C1, C2, C6, D3, D5

***Synopsis**:* Networks Models, OSI Model, TCP/IP, Transmission Media and Transmission modes, Multiplexing, Switching Technology: Packet Switching, Virtual Circuit Switching, Cell Switching, Switch Technologies, LAN and WAN.

***Textbook and Supporting Material:***

***1***- Data Communications and Networking, ,Behrouz A. Forouzan, Mc GrawHill**,** 4th edition 2007

2- Computer Networks and Internets, Douglas E. Comer, Prentice Hall Pub., 2004, 4th  Edition

3- Business Data Communications,  W. Stallings, Prentice Hall, 5th edition, 2005

***Modes of Assessment:*** Two 1-hour midterm exams (15% each) + Lab work and Coursework (15%) + Tutorial Contribution (5%) + Final (unseen) exam (50%)

…………………………………………………………………………………………………..

## 761373 E-Commerce Applications

3 hours per week, 3 credit hours, prerequisite: 731270

***Teaching Methods:*** 32 hours Lectures (2 per week) + 8 hours Tutorial (1 per 2 weeks) + 8 hours Laboratory (1 per 2 weeks)

***Aims:*** The advent of e-commerce has posed many new issues in the development of business information systems (ISs), including Accounting Information Systems. In order to develop effectively and efficiently information systems for contemporary business, the IS specialists should understand new contexts, practices, and appropriate IT - specifically web-based technologies.  The purpose of this course is to introduce e-commerce, its impacts on business processes, and keys issues in the development of web-based business information systems and applications.

The module will focus on electronic commerce applications, technologies, and tools which are used to conduct business on the World Wide Web. It reviews foundations of e-commerce, its infrastructure, current business models in business-to-customers (B2C) and business-to-business (B2B) transactions, security and quality assurance, web site design strategies, payment systems, and various issues--Internet marketing, legal, regulatory, technological, social, and ethical--which relate to electronic business, systems development issues, electronic data interchange, web-based marketing, e-supply chains, e-procurement, e-marketplace, customer relationship management, and web-enabling mobile. A major part of the course will be devoted to hands-on practices covering client-side (front-end) and server-side (back-end) applications in web-based business information systems.  Essentials of contemporary programming tools for e-commerce development such as HTML, XML, ASP (C#, VB/JavaScript) ... will be explored. E-Business case studies are used to demonstrate the advantages and the challenges related to integrating e-commerce applications.

***Learning Outcomes:***

On completion of this module, student should be able to:

1. Understand the process of setting up an interactive web site, displaying product catalogue, deploying shopping carts, handling credit card transaction. (A, B, C, D)
2- Identify e-business models
3- Describe issues of concern in the design and development of an e-commerce
4- Discuss the techniques and technologies used to process online payments
5- Understand the process of maintaining security on the E-commerce site. (C)
6- Have knowledge in XML technology related to Business-to-Business E-commerce. (A)
7- Discuss the issues facing businesses that are considering worldwide marketing of their products and services
8- Discuss how the "digital divide" is impacting on our society and impact of the Internet on education
9- Evaluate the information needs and requirements of a business entity wishing to adhere to e-commerce paradigm (B)
10- Participate in the development of an appropriate business information system to support the organizational needs. (B, C)
11- Differentiate between business-to-business marketing and business-to-consumer marketing
12- Build an online store. (B, D)
13- Enhance customer relationship management
14- Be aware of security issues and of technologies designed to ensure secure transactions
15- Work as a team
15- Write reports and make presentation
16- Work with an online store and modify it as necessary

***Assessment of Learning Outcomes:***

Learning outcomes (1), (2), and (4) are assessed by examinations and assignments. Learning outcome (3) is assessed by exams and in the laboratory.

***Contribution to Programme Learning Outcomes:***

A2, A3, A5, B1, B2, C1, C2, C6, D3, D5

***Synopsis*****:** Introduction to e-commerce; E-commerce Business Models and Concepts; E-Commerce Infrastructure: The Internet and World Wide Web, Web design, JavaScript; Internet Information Server (IIS); Personal Web Server (PWS); Introduction to Active Server Pages (ASP), Building an E-Commerce Web Site; E-Commerce Payment Systems; E-Commerce Marketing Techniques; Building product catalogue; search product catalogue, Web Spider and search agent; Ethical, Social and Political Issues in E-Commerce; Internet Communication**;** Transaction Systems; Shopping Carts, XML; -Commerce Applications: Business-to-Consumer (B2C), Consumer-to-Consumer (C2C), Business-to-Business( B2B); Digital Government, Marketplaces, and Communities; Security and Encryption, Web Security

***Textbook and Supporting Material:***

1. Turban, Rainer, and Potter, Introduction to E-Commerce, second edition, 2003
2. H. M. Deitel, P. J. Deitel and T. R. Nieto, E-Business and E-Commerce, Prentice hall, 2001
3. Darrel Ince, Developing Distributed and E-Commerce Applications, Addison Wesley, 2002
4. Walther & Levine, Sams, E-Commerce Programming with ASP, Walther & Levine, Sams, 2000
5. E-Business and E-Commerce Management: Strategy, Implementation and Practice, 3rd Ed., Prentice Hall, 2007
6. Russell Jones, Mastering Active Server Pages 3, Sybex inc., 2000
7. J. E. Rayport and B. J. Jaworski, Case in E-Commerce, McGraw Hill, 2002
8. Turban, Efraim; King, David; Lee, Jae; and Viehland, Dennis. Electronic Commerce 2006: A Managerial Perspective. Upper Saddle River, NJ: 2006.
   The Companion Website for the previous edition of this text is accessible at:
   http://myphlip.pearsoncmg.com/cw/mpbookhome.cfm?vbookid=593

**Web Sites:**
Suggested Supplemental Readings and Web Links:
Rappa, Michael. *Managing the Digital Enterprise*. URL: http://digitalenterprise.org/

***Modes of Assessment:*** Two 1-hour midterm exams (15% each) + Lab work and Coursework (20%) + Final (unseen) exam (50%)

……………………………………………………………………………………………………..

## 3.4 Advanced Modules

In this sub-section, the full descriptions of Level 3 modules are presented. Table (3-3) shows these modules and their descriptions are given below.

**Table (3-3) Advanced Modules in Computers and Information Systems Department**

| Module Number | Module Title | Prerequisite |
|---|---|---|
| 760421 | Information Security | 750333 |
| 760463 | Data Warehousing and Data Mining | 731331 |
| 760499 | Research Project | 760398 |
| 761442 | Advanced Web Programming | 731270 |
| 761443 | Wireless and Mobile Computing | 761340 |
| 761462 | Information Retrieval | 760261 |
| 761467 | Information Systems Testing | 731351 |
| ---- | Department Elective (2) | ----- |

## 760421, Information Security

3 hours per week, 3 credit hours, prerequisite: **750333**

***Teaching Methods:*** 38 hours lectures (2-3 hours per week) + 4 hours Tutorials + 3 hours Laboratory + 3 hours Seminars

***Aims:*** The aim of this module is to provide basic knowledge and skills on principles, mechanisms, and implementations of computer security. It covers the following topics: basic computing security concepts and principles, elementary cryptography and its applications, malicious codes, intrusion detection, security protocols, authentication and authorization mechanisms, secure programming, operating system security, network security, database security, security models and policies, and security administration.

***Learning Outcomes:***
On completion of this module, student should be able to
1. Explain common vulnerabilities in computer programs.
2. Explain concepts related to applied cryptography.
3. Explain the concepts of malicious code.
4. Outline the requirements and mechanisms for identification and authentication.
5. State the basic concepts in information security, including security policies, security models, and security mechanisms.
6. Describe security requirements for database security, and describe techniques for ensuring database reliability and integrity, secrecy, inference control, and multi-level databases.
7. Describe threats to networks, and explain techniques for ensuring network security, authentication, firewalls, and intrusion detection.
8. Explain the requirements and techniques for security management, including security policies, and risk analysis.
9. Conduct a security risk analysis for simple cases.

10. Identify and solve security breaches in a computer environment
11. Prepare and deliver coherent and structured verbal and written technical report.
12. Display an integrated approach to the deployment of communication skills, use IT skills and display mature computer literacy; strike the balance between self-reliance and seeking help when necessary in new situations, and display personal responsibility by working to multiple deadlines in complex activities**.**
13. Use standard security tools to locate and fix security leaks in a computer network.

***Synopsis:*** Basic and support material to be covered; Introduction to Computing security; Elementary Cryptography (1): traditional techniques; Elementary Cryptography (2): Modern techniques; Program Security: Secure programs, Non malicious program errors; Program Security: Malicious code; Access Control in General purpose Operating Systems; Trusted Operating systems: security policies, models of security; Public Key Infrastructure; Security Protocols; Database security (1): Security requirements, reliability and integrity, sensitive data; Database security (2): Inference, Multilevel databases, proposals for multilevel; Security tools; Network Security: Threats in Networks; Network Security: Firewalls; Security tools; Network Security: Intrusion detection systems; Security tools; Administrating Security: security planning, Risk Analysis; Legacy, privacy, and Ethical issues in computer security

***Textbook and Supporting Material:***

1. Charles P. Pfleeger, Shari Lawrence Pfleeger, Security in Computing, Third edition, Prentice Hall International, 2003
2. Introduction to Cryptography, Johannes A. Buchmann, Springer, 2000.
3. Cryptography and Network Security: Principles and Practices, William Stallings, prentice Hall, 2006.
4. Public Key Infrastructure: Building trusted Applications and Web services, John R. Vacca, Auerbach publications, 2004.
5. Information Security Risk Analysis, R. Peltier, Auerbach publications, 2001.
6. Practical Firewalls, Terry W. Ogletree, QUE corporation, 2000.
7. Unix Security Reference Manual
8. Windows Security Reference Manual
9. Java Security Reference Manual

***Modes of Assessment*:** Two 1- hour midterm exams (15% each) + Project and Assignments (20%) + 2-hours Final Exam (50%)

…………………………………………………………………………………………………..

## 760463, Data Warehousing and Data Mining

3 hours per week (48 hours in total), 3 credit hours, Fourth year, any semester, prerequisite: **731331**

***Teaching Methods:*** *3*0 hours lectures (2 hours per week) + 10 hours seminars (1-2 hours per 2 weeks) + 5 hours tutorials (1 per 2 weeks).

***Aims:*** The module equips students with the knowledge and skills necessary to design, implement a data warehouse/ a data mining algorithm using Oracle or any other appropriate programming language. Students are expected to become familiar with the common data mining tasks and techniques, principles of dimensional data modelling, techniques for extraction of data from source systems, data transformation methods, data staging, data warehouse architecture and infrastructure. Issues such as preprocessing the data, discretisation, rule pruning, cross validation, inductive bias, and prediction are included. Students will design and develop a simple data mining prototype using Oracle data mining package or any appropriate tools.

***Learning Outcomes:***

On completion of this module, student should be able to

1. Provide a brief introduction to general issues of Data Warehouse and Data Mining.
2. Provide students with a clear understanding of the different architectures and mining techniques
3. Introduce students to the role and function of Data Warehouse and Data Mining.
4. Understand the theoretical background of data mining tasks and techniques
5. Explain the stages and process different data mining techniques.
6. Work effectively with others, and to carry out projects in groups
7. Learn mining and warehouse techniques through the use of different tools (e.g. ORACLE, WEKA)
8. Learn the evaluation techniques of data mining and data warehouse.

***Synopsis:*** Basic and support material to be covered; Course Overview: Introduction; Knowledge discovery process; Why data warehouse & data mining; Data Warehouse: Why data warehouse?; OLTP and OLAP; Data Cube; Data Warehouse modelling; Warehouse views; Data Warehouse Architectures; Data Warehouse implementation; Data preprocessing: Why preprocess the data? Data cleaning; Data integration and transformation; Data reduction; Dimensionality reduction; Data compression; Feature extraction; Discretization and concept hierarchy generation; Applications on Data Warehouse; Case Study; What is data mining; Motivation and challenges of data mining; Data mining tasks; Types of Data; Data set types; Data mining applications; Data quality; Data preprocessing: Aggression, sampling, dimensionality reduction, feature selection, feature creation, discretisation, transformation; Measuring the similarity and dissimilarity between: Simple attributes, data objects; WEKA; Proximity measures; Issues in proximity calculation; Exploring the IRIS data set; Data Mining Techniques; Mining association rules: Association rule mining; Apriori algorithm; Frequent Pattern Growth algorithm; Rule based Classification; What is classification: Decision trees: ID3, C4.5; Rule induction: RIPPER algorithm; Data Mining Techniques: Rule based Classification: Associative classification (CBA, MMAC); Rule Pruning : REP, database coverage; Data Mining Techniques: Statistical classification: Naïve bayes; Issues in Classification: Overfitting and cross-validation; Evaluation methods in Classification; Data Mining Techniques: Other classification approaches: Regression; Neural networks; Genetic algorithms; Cluster analysis: Partitioning methods (K-means); Hierarchical methods (BIRCH and CURE); Outlier analysis: Preliminaries; Statistical approaches; Density-based methods; Case Study: Text Categorization; Using associative classification for text categorization

***Textbook and Supporting Material:***

1. Tan, P-N, Steinbach, M., Kumar, Introduction to Data Mining, Addison Wesley, 2005
2. Han, J. and Kamber, M**,** Data Mining: Concepts and Techniques, Morgan Kaufmann, 2006
3. Connoly, T., and Begg, C. Data Base Systems: A Practical Approach to Design, Implementation, and Management, Addison Wesley, fourth edition, 1999.
4. Witten, I., and Frank, E. Data mining: practical machine learning tools and techniques with Java implementations. San Francisco: Morgan Kaufmann, 2001.
5. Modern Data Warehousing, Mining, and Visualization, by George M. Marakas, 2003, Prentice-Hall.

**Software:**

1. WEKA
2. Oracle data warehousing package

***Modes of Assessment*:** Two 1- hour midterm exams (15% each) + Project and Assignments (20%) + 2-hours Final Exam (50%)

……………………………………………………………………………………………………

## 760499, Research Project

3 hours per week (48 hours in total), 3 credit hours, Fourth year, any semester, prerequisite: **760398**

***Teaching Methods:*** Duration: 16 weeks; 6 or 7 hours Lectures + none scheduled Laboratories 120 hours expected through semester

***Description:*** The research project consists of a single project on which the student works over a period of 16 weeks that can be extended to 32 weeks (2 semesters). It is assumed that the student spends a nominal 192 hours (or 384 hours), the equivalent of 12 hours per week, working on this. There are three deliverables: demonstration, discussion, and a written report.
A student works under the supervision of a member of staff, the Supervisor. Most of the projects involve three students working together on the same project; apart from these, all students do different projects.

***Aims:*** The aims for the project work done in the fourth year are:

1- To manage and execute a substantial project in a limited time.
2- To identify and learn whatever new skills are needed to complete the project.
3- To apply design and engineering skills in the accomplishment of a single task. In this context the skills mentioned may be in the general area of design and engineering in its broadest sense, or may be very specifically related to particular tools.

***Learning Outcomes:***

On completion of this module, a student should

1. Know how to use the project supervisor appropriately as project consultant or customer. (D)
2. Plan, execute and complete a significant design and, as appropriate, implementation within the time budget available. (B, C)
3. Give a demonstration showing practical competence and demonstrating the results of the project. (C).
4. Document the project in a final report. (C)

***Modes of Assessment:***

Project Examination Committee mark: 1005% (demonstration and discussion 30%, Report 30%, and Technical aspects 40%)

***Contribution to Programme Learning Outcomes***

B1, B2, B3, C1, C2, C3, D1, D2, D3, D4, D5

***Syllabus***

The occasional lectures are on topics of particular interest to students doing a project in their final year: Overview of projects and project assessment; Career advice; How to give a seminar; Writing English; How to give a demonstration; How to write a project report;

***Reading List and Supporting Material:***

1- C. W. Dawson. The Essence of Computing Projects, A Student's Guide. ISBN 0-13-021972-X. Prentice Hall 2000.

The project notes for guidance in carrying out a project are available in the Research Project Committee.

**How to Choose a Project?**

The list of projects for each semester will be available at the beginning of the semester. This list will contain the projects title and names of supervisors. The main selection and allocation of students to projects was made at the beginning of the semester. It is possible for students to propose their own projects, in which case, they should prepare a proposal and give it to the Research Project Committee (RPC).
Usually each project is suitable for more than one student (normally 3 students). Therefore, groups of three students should be arranged by students themselves. Each group of three students should make three choices of projects on the selection form obtained from the RPC.
Students are strongly encouraged to see the associated members of staff for projects they are interested in, to find out more about the projects.

### OWN Projects

If a student has successfully negotiated a project - outside the list of projects given by the department - with the project committee, and possibly a prospective supervisor, he/she still gives another 2 choices, and code choice number 1 as "OWN"; this is likely to be the student's first choice, but it does not have to be.

### Project Timetable

Students are expected to be in regular attendance working on their projects. They must co-operate in maintaining regular contact with their supervisors. It is an attendance requirement that students see their supervisors every week during term time. The formal project deliverables are a demonstration with discussion, and a written report.

The project lifecycle should follow a sensible methodology and include the various stages identified in any Software Engineering course.

Work on the project itself, in particular use of equipment and computing facilities, must finish at the end of the 12th week of the semester. In some cases, this can be extended to another semester.

The project report and the Auxiliary Appendix together with any relevant discs, logic circuit and wiring diagrams etc., must be handed in to the Graduation Project Committee after being signed by the supervisor by the end of the 15th week of the second semester. The GPC will announce a timetable for all project discussions. It forms a number of discussion committees, where each consists of two staff members and discusses one project.

The formal demonstrations and project discussion take place within one week after the submission of the report. The demonstration and discussion will contribute to the assessment of the "Quality of the project work".

### Demonstration

The demonstration is an informal presentation of the results of the project to one of the project discussion committee. The students will say briefly, what the aims of the project are, and will then demonstrate the results for example by running the program or using the equipment constructed. The duration is about 20 minutes. See **Guidance on demonstrations** below for more information.

### Report

The report is a formal written report on the project. This must be word processed. The report must follow a **set of standards**, given below, to facilitate its inclusion in the library and its usefulness for subsequent readers. Besides these, student will find it useful to read the slides of the talk given on writing, which is given in the lecture.

Copies of previous graduation project reports are available for reference in the Department.

Project documentation may be prepared on the PCs and printed on a laser printer. Students should hand in three soft cover copies of the report. After the discussion with the discussion committee, students should make all the correction that are suggested by the committee within the specified period of time under the supervision of their supervisors, then they should handed in three blue colour hard cover copies of the project. The title of the project, the University, Faculty, Department names, and students' names are all written in golden colour.

### Overhead Projector

Students are expected to make reasonable use of the overhead projector or power point presentation on the day of their demonstrations.

### Guidance on Demonstrations

A demonstration lasts about 20 minutes.

The group of students should aim to spend no more than 10 minutes summarising what their project is designed to achieve and showing what it currently does achieve. The rest of the time is spent in answering questions.

**Note**: Students should not attempt to demonstrate on the computer every last thing their program can do. A demonstration of its basic operation plus one or two highlights should suffice.

The mark given for the demonstration is based on the quality and quantity of the work attempted and the final state of achievement.

Students should have their working documents to hand and appropriate reference material, design workings, reasonably up-to-date listings, examples, tests, etc. They are not giving a 20-minute seminar; at least half the time must be available for questions.

Obviously, the kinds of things that are sensibly shown in a demonstration vary from project to project. If students are in doubt as to what to show, they should ask their Supervisors.

The discussion committee consists of two staff members plus the supervisor.

In general, students should be available and ready to start their demonstrations at least within one week of their submission of the project.

**Report Standards**

1. The report is a formal written account of the project, satisfying certain standards for inclusion in a library. Students must hand in all relevant work on the project by the end of the 11th week of the second semester. In addition to the report, this includes program listings, discs, detailed logic and wiring lists, etc. It is important to meet this deadline. When students hand this to their supervisors it must be accompanied by a signed version of a form supplied by the GPC. In the case of programming projects, program listings must be submitted in some bound form in an ``Auxiliary Appendix" that does not need to satisfy any particular standard apart from being neat and tidy. It is suggested however that an economical listing would be double-sided on A4.

   Here is a suggested structure for a report. Some projects may be rather different from others, and therefore have good reasons for not following these suggestions exactly. Supervisor guidance should anyway be sought!

   - Introduction (1st chapter). What is the overall aim of the project. Why is it worth doing? Who will benefit from it? If the overall aim can be split into a number of subgoals, this is a possible place to do it. Finish with a chapter by chapter overview of the rest of the report.
   - Background (2nd chapter). Analyse the background to the project. This should mention any previous work, here or elsewhere, and explain its relevance to the project. This could be an appropriate place to justify the choice of platform/software etc. used in the project.
   - Description of the student's own work: Design and Implementation (a chapter each). The structure of these chapters may reflect the project lifecycle, but do not write a diary of progress. The design should be clearly described and justified. Supporting diagrams should be used where appropriate and helpful. Keep your design description fairly high level. When describing implementation, confine yourself to the important, difficult, or interesting bits. Do not include large chunks of code. Figures may well be useful.
   - Results (1 chapter). What is the resulting system like to use. Include screen shots as appropriate.
   - Testing and Evaluation (1 chapter). What testing was done? How confident are student that everything works correctly, and what evidence can they produce to support this claim? Have students evaluated the system against its aims? How did they make this evaluation?
   - Conclusions (last chapter). What conclusions can students draw from the whole project? This should include a clear statement of what has been achieved overall, and will normally continue by suggesting areas of further related work, which could be done.

2. The report itself (apart from technical considerations) is worth 25% of the project mark. However, it forms the basis of an independent assessment of the project and therefore has greater effect than 25% in practice.

3. The report must be on paper of A4 size (210 x 297 mm). Only one side of paper should be used except in the Auxiliary Appendix.

4. The report must be produced using word processing facilities. The body of the report should be suitably divided into chapters and sections. Chapters, sections, pages, figures and appendices should all be numbered. Chapters, sections and appendices should have a heading. Each chapter should start on a new page. The body of the report should be preceded by a temporary title page, an abstract and a list of contents, and it should be followed by the references and then any appendices. References to other published work should follow the conventions used in giving references in published work. e.g.: [1] P.J. Denning. Human error and the search for blame. *Communications of the ACM* 33(1): pp 6-7, January 1990. The abstract page must give the title, author, and supervisor, as well as an abstract of the project.

5. Straightforward and peripheral aspects of the work done should be mentioned only briefly, and description and explanation concentrated on important and interesting aspects. No extra credit is gained by writing a long report and excessive length is detrimental. More detailed description should be placed in appendices to the report. The appendices and/or the Auxiliary Appendix should contain any further documentation. Only the report itself will be held in the Department. Therefore, where important material is not included in it, e.g. because it is not convenient to produce it in A4 format, or it would be too bulky, it may sometimes be appropriate to include extracts in the report.

**Copyright**
In general, it is an infringement of copyright to reproduce any material, except short acknowledged quotations, from a published book or journal without the written permission of the publisher.
Except for the copying of material that is clearly from internal documents of the Department, any copying of books, journals, or documents required for the report should be checked with the supervisor before it is carried out.
Any material that is copied **must** be acknowledged as such. Attempting to present material written by others as your own is plagiarism and a serious disciplinary offence, as described in the University guidelines in the Undergraduate Handbook.

**Marking Scheme for Reports**
The report, as a document, is worth 30% of the project mark. These marks are divided among the following headings:

- Organisation (10%): balance of content, clarity, flow, relevance.
- Context (5%): discussion of background, aims, and significance of achievements.
- Literacy (5%): English, style, report manner.
- Presentation (10%): tidy layout, headings, references, diagrams.

…………………………………………………………………………………………………………..

## 761442, Advanced Web Programming

3 hours per week (48 hours in total), 3 credit hours, Fourth year, any semester, prerequisite: **731270**

***Teaching Methods:*** 32 hours Lectures (2 per week) **+** 8 hours Laboratory on project assignment (1 per week) + 8 hours Seminars presentations (in last 3 weeks)

**Aims:** This module aims to present advanced techniques of Web designing and programming. It introduces networks, and the paradigm of client / server.

***Learning Outcomes:***
On completing this module, students should be able to:
1. Understand how to design Web pages and use HTML. (A)
2. Be familiar with the concepts of client / server. (A)

3. Design any large-scale Web sites. (B, C)

***Synopsis:*** Introduction to networks; Informal retrieval; The client server paradigm; Legal and ethical considerations of web-based applications; Designing large-scale web sites; Dynamic page design with scripting; Object oriented scripting; Scripting language structure and syntax; Scripting events and event handlers; Objects and navigation; Applications for scripting in animation; data validation, data persistence, and user interaction; Synchronized and embedded multimedia with text, images, video and audio; Bandwidth consideration; XML: AML markup, well-formedness, valid documents, DTDs, XML objects, styling XML with CSS, XSL.

***Textbook and Supporting Material:***

1- Marty Hall and Larry Brown, Core Servlets and JavaServer Pages (ISBN 0-13-009229-0), 2/e

2- Marty Hall and Larry Brown, Core Web Programming (ISBN 0-13-089793-0), 2/e

3- Deepak Alur el al., Core J2EE Patterns (ISBN 0-13-142246-4), 2/e

4- Steve Graham et al., Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI (ISBN 0-672-32641-8), 2/e

5- Ian F. Darwin, Java Cookbook (ISBN 0-596-00701-9), 2/e

***Modes of Assessment:*** Two 1-hour midterm exams (15% each); Assignments (15%); Seminars (5%); Final Examination: 2-hours written exam (40%) + defended project (10%)

…………………………………………………………………………………………………….

## 761443, Wireless and Mobile Computing

3 hours per week, 3 credit hours, prerequisite: **761341**

***Teaching Method***: 40 hours Lectures (2-3 hours per week) + 8 hours Laboratory for project assignment (1 per fortnight)

***Aims:*** Mobile computing and wireless networks is a young and dynamic field. The rabid advance in miniaturization of computing machinery and " unuttered " communication technology, together with the visionary demands for ubiquitous access to information, have introduce new constraints in many traditional areas of computer science. This module will cover a broad selection of topics in data communications, resource management, network protocols, distributed computing, information management, user interfaces, application /services, and security. Students will learn the principles of Mobile Computing and its enabling technologies, and explore a young but rich body of exciting ideas, solutions, and paradigm shifts.

***Synopsis***: Wireless Technology, Mobile IP, Mobile TCP, Mobile & Wireless Security, Mobile OS, Mobile Computing Models, Mobile Application Development, Mobile Data Management, User Interface Issues.

***Textbooks and Supporting Material:***

1- Any Time, Anywhere Computing: Mobile Concepts and Technology, Kluwer Academic Publisher

***Modes of Assessment***: Two 1-hour midterm exams (15% each); Assignments (20%); 2-hours written final exam (50%)

……………………………………………………………………………………………………..

## 761462, Information Retrieval

3 hours per week, 3 credit hours, prerequisite: **760261**

***Teaching Methods:*** 42 hours Lectures (2-3 per week) + 6 hour Tutorials (1 per 2 weeks)

***Aims:*** This module aims to give students an understanding of the fundamental techniques for hypermedia architectures, design and usability, document management and retrieval, metadata management, and searching the web.

***Learning Outcomes:***

On completion of this module, student should:

1- Be familiar with the fundamentals of hypermedia systems, and hypermedia design and usability methodologies.

2- Understand the difficulty of representing and retrieving documents.

3- Understand the latest technologies for linking, describing and searching the Web.

4- Understand the relationship between IR, hypermedia, and semantic models.

5- Be familiar with classical techniques of Information Retrieval, and the additional techniques employed by Web search engines sufficient to understand how Web search engines work and how they could be improved.

6- Be familiar with techniques for conveying the meaning of documents or hypermedia content, for example, metadata, thesauri, and classification taxonomies – sufficient to understand their application to the "semantic Web".

7- Be familiar with the fundamentals of hypermedia systems sufficient to know how to develop a good Web hypermedia and why a Web site is good or bad.

8- Be able to implement techniques for the preprocessing needed for information retrieval systems.

9- Be able to develop a small information retrieval system.

***Contribution of Programme Learning Outcomes:***
***A2, A3, A5, B1, B3, C2, D***

***Synopsis:*** Introduction to Information Retrieval (IR) systems: Goals and history of IR; the impact of the web on IR; Related areas to IR; Basic IR Models: Boolean and vector-space retrieval models; ranked retrieval; text-similarity metrics; Basic IR Models: text-similarity metrics; TF-IDF (term frequency/inverse document frequency) weighting; cosine similarity; Basic Searching and Indexing: Simple tokenizing, stop-word removal, and stemming; Basic Searching and Indexing: inverted indices and files; efficient processing with sparse vectors; Experimental Evaluation of IR: Performance metrics: recall, precision, and F-measure; Evaluations on benchmark text collections; Query Operations and Languages: Relevance feedback and query expansion; Query Languages**;** Text representation and properties: Word statistics; Zipf's law; Porter stemmer; morphology; index term selection; using thesauri; Metadata and markup languages (SGML, HTML, XML, DTD) and schema Web linking technologies; Hypermedia: Introduction; Hypermedia architectures and models: closed hypermedia (HyperWave), open hypermedia (DLS, Microcosm), the Dexter model, AHM, HAM Using Hypermedia: browsing, navigation and orientation, paths, trails; Hypermedia design: modeling methodologies (OOHDM, RMM), link consistency, link patterns, rhetoric and context, Usability and evaluation

***Textbooks and Supporting Material:***

1- Ricardo Baeza-Yates and Berthier Ribeiro-Neto, Modern Information Retrieval, Addison Wesley, 1999; **http://www.sims.berkeley.edu/~hearst/irbook/**

2. David Lowe and Wendy hall, Hypermedia and the Web: an Engineering Approach, John Wiley, 1999, ISBN: 0-417-98312-8

3. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, An Introduction to Information Retrieval, Cambridge University Press, Cambridge, England, 2007

4. R.K. Belew, Finding out about--A cognitive perspective on search engine technology and the www, Cambridge University Press, 2001

**Web Sites:**

http://www.cs.utexas.edu/users/mooney/ir-course
http://www.ischool.washington.edu/efthimis/courses/lis544
http://www.informationretrieval.org/

***Modes of Assessment:*** Two 1- hour midterm exams (15% each); Project work (15%); Assignment (5%); 2-hours Final Exam (50%)
……………………………………………………………………………………………………….

## 761467, Information Systems Testing

3 hours per week, 3 credit hours, prerequisite: **731351**

***Teaching Methods:*** 40 hours Lectures (2-3 per week) + 6 hour Tutorials (1 per 2 weeks) + 2 hours Laboratory

***Aims:*** The aims of this module are to introduce the role of business analysts in the testing process of information systems; explain how to test a computerized information system using business requirements; explain how to plan, prepare and conduct user acceptance testing; and explain how to find and generate test cases from business requirements

***Learning Outcomes:***
On completion of this module, student should be able to
1. Explain the role of business analysts in the testing process of computerized information systems.
2. Distinguish among different types of testing.
3. Understand the concepts of test strategies, test plans and test cases.
4. Differentiate between validation and verification activities.
5. Create and document a test Plan.
6. Create and apply test cases.
7. Conduct reviews and inspections for verification and validation
8. Execute the test strategy and test plan
9. Evaluate and interpret test results
10. Identify and manage risks in relation with testing
11. Design and specify black box tests and understand the business analyst role in white box and system testing.
12. Ensure complete test coverage with effective use of decision tables and decision trees
13. Prepare and deliver coherent and structured verbal and written technical report.
14. Display an integrated approach to the deployment of communication skills, using IT skills and display mature computer literacy; strike the balance between self-reliance and seeking help when necessary in new situations, and display personal responsibility by working to multiple deadlines in complex activities.
15. Use available software testing tools.

*Synopsis:* **Quality Assurance and Testing:** purposes, processes, verification/validation concepts, deliverables, teams; **Project Management for Testing:** System approaches to project management, Effective software Test Management Practices; Test Management Phases; **Role of the business analysts in the testing process; Test strategy development; Overview on Testing levels and types:** -System testing (unit, integration), User Acceptance testing, Performance testing, Load/stress testing, Security testing, User Interface testing: **User Acceptance Testing:** test case specifications, **User Acceptance Testing:** Test plans; **Testing techniques:** Static/dynamic testing, black box, gray box; **Testing techniques:** input/output testing, negative versus positive tests**; Engineering Test Data (1); Engineering Test Data (2); Verification:** Reviews and Inspections**; Test execution (1); Test execution (2); Test results evaluation and interpretations; Managing Risks related to testing; Testing tools Laboratory**

***Textbooks and Supporting Material:***
1- Andreas Spillner, Tilo Linz, Thomas Rossner Mario Winter, Software Testing Practice: Test Management: A Study Guide for the Certified Tester Exam ISTQB Advanced Level, Rocky Nook Publisher, 2007

2. A Practitioner's Guide to Software Test Design, Lee Copeland, Edition Artech House, , 2004
3. Software Testing, Ron Patton, Edition, Sams Publishing,
4. The Art of Software Testing, Glenford J. Myers, et al, ed. John Wiley & Sons Inc, 2004
5. Pragmatic Software Testing: Becoming an Effective and Efficient Test Professional Rex Black, Edition. John Wiley & So ns, 2007

***Modes of Assessment:*** Two 1- hour midterm exams (15% each); Assignment (20%); 2-hours Final Exam (50%)

……………………………………………………………………………………………………..

## 3.5 Elective Modules

Each student should select 2 modules out of a list of 14 modules according to his/her interest. The Department has a list of elective modules, which can be updated according to the staff expertise and the most recent trends in the field of Computer Science. The current list of such modules is shown in Table (3-4), where some modules are marked with (R) to indicate that these modules are research-oriented according to the staff expertise.

**Table (3-4) Elective Modules in Computer Information Systems Department**

| Module Number | Module Title | Prerequisites |
|---|---|---|
| 761444 | Computer Networks Design | 761340 |
| 761464 | Information Systems Evaluation | 731351 |
| 761474 | E-Learning | 761373 |
| 761491 | Special Topics | Dept. Agmt. |

**761444 Computer Network Design**

**Course/module description:**

The course covers first the study of traditional and latest networks technologies, Internet technologies, and backbone technologies in terms of their features (services, cost, quality, performance.) and the ways to make them working and operational (Frame Relays, ATM, VSAT, VPN, Internet…), then the study in details of the process of network design (from definition of requirements to design validation). The course content will be illustrated with some simple case studies.

**Course/module objectives:**

This course aims to:

- provide a depth knowledge of network technologies in terms of their features and operational use.
- provide an overview of the network design process and detail each step of this process.

**Books (title , author (s), publisher, year of publication)**

1. Business Data Communications and Networking, FitzGerald & Dennis, 8th edition, 2008, John Wiley and Sons.
2. Top Down Network Design, Priscilla Oppenheimer, CISCO Press, second edition, 2004.
3. Designing Wide Area Networks and Internetworks: A practical Guide, J. Scott Marcus, Addison Wesley, 1999.

**Support Material:** Slides

**Teaching methods**

**Duration:** 16 weeks, 48 hours in total.
**Lectures:** 38 hours (2-3 hours per week)
**Laboratory**: 3 hours
**Tutorial:** 4 hours
**Seminar:** 3 hours

**Learning outcomes**

- Knowledge and understanding

1. Understand a wide range of traditional and latest network technologies: their features, and operational use.
2. Understand the professional and ethical responsibilities of the practicing computer professional including understanding the need for quality.

- Cognitive skills (thinking and analysis).

3. Conduct with success the whole network design process.
4. Build campus networks, campus backbones, enterprise networks.

- Communication skills (personal and academic).

5. Plan and undertake a major individual project, and prepare and deliver coherent and structured verbal and written technical report.
6. Be able to display an integrated approach to the deployment of communication skills, use IT skills and display mature computer literacy; strike the balance between self-reliance and seeking help when necessary in new situations, and display personal responsibility by working to multiple deadlines in complex activities.

- Practical and subject specific skills (Transferable Skills).

7. Configure usual network (backbones) devices such as LAN switches, Routers.

**Assessment of Learning Outcomes**

Learning outcomes (1-3) are assessed by examinations, tutorials. Learning outcomes (4-7) are assessed by assignments and laboratory.

## 761491, Special Topics

3 hours per week, 3 credit hours, prerequisite: **Department Agreement**

***Teaching Methods:*** 48 Lectures or it depends on the chosen topic that might include seminars hours as well.

***Aims:*** This module aims to offer any recent topic in computer science. The chosen topic may be different from semester to another.

***Learning Outcome:***
It depends on the chosen topic.
***Synopsis:*** For this module, the department can choose any recent topic to cover it within one semester.
***Textbooks and Supporting Material:***
According to the selected topic

***Modes of Assessment:*** It depends on the chosen topic.

……………………………………………………………………………………………………..

# APPENDIX B

# STUDY PLAN

# OF

# COMPUTER INFORMATION SYSTEM PROGRAMME

## Philadelphia University
## (Private Accredited University)

## Study Plan for Bachelor Degree in Computer Information Systems
## (132 Credit Hours)

## Faculty of Infor
### Department of
### (200

**First: University Requirements (27 Credit Hours)**

| | | | | |
|---|---|---|---|---|
| 130101 | English Langauge Skills (1) | 3 | ---- | |

**2- University Electives : ( 15 credit hours )**

**(The student studies (15) credit hours from the following**

**areas with a minimum of one module from each area and a maximum of two modules)**

| **a. Sciences of Humanity Area (3 - 6) Credit Hours** | | | | |
|---|---|---|---|---|
| **Module No.** | **Module Name** | **Credit hours** | **Prereq.** | **Mark** |
| 110102 | Arabic Language Skills (2) | 3 | 110101 | |
| 130103 | English Lanagauge Skills (3) | 3 | 130102 | |
| 140101 | French Language Skills (1) | 3 | --- | |
| 140104 | Foreign Language (Italian 1) | 3 | --- | |
| 140106 | Foreign Language (Hebrew 1) | 3 | --- | |
| **b. Sciences of Social and Economic Area (3 - 6) credit hours** | | | | |
| 111111 | Introduction to Sociology | 3 | --- | |

**Second: Faculty Requirements (24 Credit Hours)**

| Module No. | Module Name | Credit hours | Prereq. | Mark |
|---|---|---|---|---|
| 761211 | Windows Programming | 3 | 721120 | |
| 761272 | Multimedia Systems | 3 | ---- | |

**Third: Major Requirements (81 Credit Hours)**

**a- Compulsory Modules (63 Credit Hours)**

| Module No. | Module Name | Credit hours | Prereq. | Mark |
|---|---|---|---|---|
| 721210 | Introduction to Software Engineering | 3 | 731150 | |
| 721221 | Object Oriented Data Structures | 3 | 721120+<br>210104 | |
| 721331 | Software Project Management | 3 | 731332 | |
| 731331 | Database Applications | 3 | 760261 | |
| 731332 | Systems Analysis and Design | 3 | 760261 (or concurrently) | |
| 731351 | Information Systems Modelling | 3 | 731332 | |
| 750232 | Computer Architecture | 3 | 731150 | |
| 750322 | Design and Analysis of Algorithms | 3 | 210103+<br>721221 | |
| 750333 | Principles of Operating Systems | 3 | 750232 | |
| 750351 | Fundamentals of Artificial Intelligence | 3 | 721221 | |

**b- Elective Modules**

| Module No. | Module Name |
|---|---|
| 761444 | Computer Networks Design |
| 761464 | Information Systems Evaluation |
| 761474 | E-Learning |
| 761491 | Special Topics |

**c- Supplementary Requirements (1**

| Module No. | Module Name |
|---|---|
| 210101 | Mathematics (1) |
| 210103 | Mathematics for Computing |
| 210104 | Discrete Structures |
| 210231 | Introduction to Statistics and Probabilities |

| | | | | |
|---|---|---|---|---|
| 111112 | Introduction to Psychology | 3 | --- | |
| 111133 | Culture and Civilization (1) | 3 | ---- | |
| 111142 | Communication and Society | 3 | --- | |
| 330101 | Introduction to Management | 3 | --- | |
| | | | | |

**c. The Area of Science, Technology, Agriculture, and Health**

**(3-6) Credit Hours**

| | | | | |
|---|---|---|---|---|
| 210121 | History of Science | 3 | --- | |
| 240141 | Principles of Nutrition | 3 | --- | |
| 240151 | Man and Environment | 3 | --- | |
| 620105 | Automobile Essentials | 3 | --- | |

| | | | | |
|---|---|---|---|---|
| 760261 | Database Fundamentals | 3 | 721221 | |
| 760398 | Practical Training * | 3 | Dept. Agrt. | |
| 760421 | Information Security | 3 | 750333 | |
| 760463 | Data Warehousing and Data Mining | 3 | 731331 | |
| 760499 | Research Project | 3 | 760398 | |
| 761340 | Fundamentals of Computer Networks | 3 | 721221 | |
| 761373 | E-Commerce Applications | 3 | 731270 | |
| 761442 | Advanced Web Programming | 3 | 731270 | |
| 761443 | Wireless and Mobile Computing | 3 | 761340 | |
| 761462 | Information Retrieval | 3 | 760261 | |
| 761467 | Information Systems Testing | 3 | 731351 | |

**Major modules include 25% practical wor**

*** Registration for practical training modu**

**Faculty regulations.**

**All students must apply for the level exam i**
**The student who failed in any of these exam**
**corresponding preliminary module.**