Philadelphia University

Faculty of Information Technology

Department of Computer Science

Undergraduate Programme Handbook
(2009 – 2010)

Date: September 2009

Contents

This handbook, which is also available on the web, contains important general information for students undertaking Undergraduate Degree programme in the Department of Computer Science. It includes information about the Degree Programme in the Department but not descriptions of individual course units (modules). Details of the modules you may take are given in a separate document called Undergraduate Course Catalogue. An electronic version can be consulted on the Department Web site at
www.philadelphia.edu.jo/it-cs.asp.

Your degree program is subject to regulations contained in the **University Students Guide**. This departmental handbook interprets the regulations and your tutors may give advice, but the University Students Guide defines the regulations.

1. GENERAL INFORMATION

1.1 Mission Statement

The mission of Computer Science Department is derived from the over all IT Faculty and University missions. The Computer Science Department at Philadelphia University was founded in 2003 after closing the department of Computer and Computer Information Systems. The Department is committed to provide an opportunity to students with varied entry qualifications to obtain relevant and well rounded education, through the provision of a high quality Degree programme, which is well resourced and is supported by a good quality research. Its mission is to pursue outstanding teaching and to provide high quality learning in pure and applied computer science. The Department has one of the largest and most comprehensive computer science undergraduate programs in Jordan. It intends to produce its graduates as competent computer science practitioners who have a solid foundation of basic and fundamental knowledge and experience in applying the existing IT to contemporary problems. The Department aims to maintain an environment that promotes innovative thinking, values mutual respect and diversity, encourages and supports scholarship, instils ethical behaviour, and engenders life-long learning. The strategies of the Department are set to meet the demands of a rapidly evolving world, and to meet the needs of a developing job market in Information Technology. This program well addresses the analytic skills required by students to develop their abilities in research and to proceed for postgraduate study.
In addition, the Department creates opportunities for students to understand and gain competence as a computer science practitioner.

1.2 Key Academic Staff

Dean of the Faculty
Dr. khaldoun batiha kh_batiha@philadelphia.edu.jo

Vice Dean of the Faculty
Dr. Nameer El Emam nemam@philadelphia.edu.jo

Head of Department
Dr. Nameer El Emam nemam@philadelphia.edu.jo

1.3 Tutors

As soon as you are enrolled in the Department, a tutor will be assigned for you. This tutor is one of the academic staff members in the Department who will guide and help you throughout your stay in the Department.

1.4 Registration

Admission criteria are issued by the Higher Education Council, which governs all private universities (55% in the Tawjihi exam, the scientific branch). First year students must attend the University and they will be given a full timetable for the introductory activities. Departmental and University registration must be completed at the time specified in the introductory timetable. Returning students must also register in the times specified during introductory week. You may consult the University calendar at the web page www.philadelphia.edu.jo/arabic/event.asp.

1.5 Timetable

Lectures timetable is published separately from this book and is available on the University web site. Whilst every attempt is made to timetable reasonable combinations of course units (modules), various constraints make some combinations and outside options impossible. If you have a timetable problem, please consult your personal tutor in the first instance.

1.6 Use of Notice Boards

Official notices are posted on the Department notice board and on the Faculty general notice board on the third and fourth floors of the Faculty. Notices are often also posted on the University web site. Electronic mail is also used extensively for communication with the Department and University. Each lecturer provides the students with his/her e-mail at the beginning of the term. Most official information including copies of this handbook, the undergraduate course catalogue, and timetables are available on the Computer Science Web pages www.philadelphia.edu.jo/it-cs.asp. This includes directories of staff and students for internal use, completed with photographs.

1.7 Health and Safety in the University

The University has a Health and Safety Committee, which comprises representatives of all services within the University. It is the responsibility of this committee to investigate complaints and potential hazards, examine the cause of all accidents, and carry out periodic inspections of all areas of the University. At registration you will be required to assent to the University code of behaviour which relates to health and safety in the University buildings as well as the responsible use of Computer equipment as required by the Department of Computer Science.

1.7.1 Buildings

The Department comprises two kinds of buildings: Class Rooms and IT Laboratories. The buildings are generally open between 08.00 and 19.30 (Sunday – Thursday). In accordance with University policy, smoking is prohibited throughout all buildings.

1.7.2 Emergency Evacuation

It is the responsibility of every individual to familiarise themselves with the Faculty's buildings and be aware of the fire exits (which are clearly marked).

- After evacuation of any building please assemble well away from the building and do not block any exit.
- Do not return to any building until authorised to do so.

1.7.3 Fire Action

Fire Action notices and important telephone numbers are located at all floors of the Faculty and all staff and students should make themselves acquainted with this routine.

On hearing the continuous alarm you should evacuate the building immediately by the nearest exit.

1.7.4 Operating the Fire Alarm

The manual fire alarm system can be activated by breaking the glass in the red contact boxes sited at strategic points throughout the premises.

1.7.5 Use of Fire Appliances

Fire appliances are sited at strategic points throughout the Faculty to deal with fires. Fires should only be tackled provided there is no personal danger and after the alarm has been set off.

1.7.6 First Aid

If any thing happened to you, you can get first aid from the health center located near the Nursing Faculty.

1.7.7 Personal Difficulties

Please inform the head of Department or your tutor of any difficulties with which the Department can be of assistance.

2. PROGRAMME OVERVIEW

2.1 Aims and Learning Outcomes of the Programme

The Department offers the degree of BSc Computer Science (in 4 years). The Department, being the first among many Computer Science programs in Jordan, with its excellent teaching quality, provides a very rich learning environment for undergraduates. Sections 2.1.1 and 2.1.2 details the aims and learning outcomes of this programme respectively.

2.1.1 Aims

Computer Science program at Philadelphia University gives you the opportunity to:

- Enable you to develop your capacity to learn and participate in society as competent professionals;

- Prepare you for the world of work and develop self-confidence and problem solving abilities;
- Develop among students the awareness of the social, organizational, and professional context in which you will be working;
- Be a graduate who will be able to contribute to and take active part in a variety of industrial, commercial, and academic activities;
- Be a graduate who exhibits a range of broad based skills and activities related to Computer Science;
- Be a graduate who can adapt to changing technology and have the ability to recognize technological and human trends;
- Be a graduate who meets the industry standard in Computer Science and have experience in the use of general tools and technologies used in the design and implementation of software;
- Provide different study opportunities, which are comparable with national, and international academic qualifications;
- Engender among students the spirit of research and enquiry through suitable mechanism such as departmental research;
- To develop transferable skills such as verbal and written communication, teamwork leadership, etc.

2.1.2 Learning Outcomes

Learning outcomes describe what you should know and be able to do if you make full use of the opportunities for learning that we provide. All these skills are described in the following areas (A, B, C, and D). In the individual module syllabi, the categories of learning outcomes (A, B, C, and D) and the individual learning outcomes appropriate to the module are identified.

A- Knowledge and Understanding

A1) Know and understand the essential mathematics relevant to Computer Science.

A2) Understand and apply a wide range of principles and tools available to the software developer, such as design methodologies, choice of algorithm, language, software libraries and user interface techniques.

A3) Know and understand the principles of various current applications and research areas of the subject including artificial intelligence, databases, software engineering, net-centric, and distributed systems.

A4) Know and understand a wide range of software and hardware used in development of computer systems.

A5) Recognise the professional and ethical responsibilities of the practising computer professional including understanding the need for quality, security, and computer ethics.

B- Intellectual (thinking) skills - able to

B1) analyse a wide range of problems and provide solutions related to the design and construction of computer systems through suitable algorithms, structures, diagrams, and other appropriate methods.

B2) identify a range of solutions and critically evaluate them and justify proposed design solutions.

B3) design and implement practical software systems.

B4) practice self-learning by using the e-courses.

C- Practical skills - able to

C1) Plan and undertake a major individual / group project in the areas of computer science.

C2) Prepare and deliver coherent and structured verbal and written technical reports.

C3) Give technical presentations suitable for the time, place, and audience.

C4) Use the scientific literature effectively and make discriminating use of Web resources.

C5) Design, write, and debug computer programmes in appropriate languages.
C6) Use appropriate computer-based design support tools.

D- Transferable skills - able to

D1) Display an integrated approach to the deployment of communication skills.
D2) Use IT skills and display mature computer literacy.
D3) Work effectively with and for others.
D4) Strike the balance between self-reliance and seeking help when necessary in new situations.
D5) Display personal responsibility by working to multiple deadlines in complex activities.
D6) Employ discrete and continuous mathematical skills as appropriate.

In order to provide students with the "life long learning" attitude, the teaching method is essentially based on self learning (3 hours in class rooms and 6 hours out of class rooms: coursework, practical works, workshops, seminars, etc.)

2.2 Overview of the Programme Structure

The system of study at Philadelphia University is the courses system that depends on the credit hours. Each academic year consists of two semesters and an optional semester (the summer semester). An individual course of lectures is known as a "**course unit**" or a "**module**". Each module has one or more prerequisite modules. The curriculum contains modules that are from University Requirements, Faculty Requirements, Department Requirements, and Supportive Requirements. Each module has 3 credit hours per week. However, some modules are supported by tutorials and some continuous assessment, such as seminars or laboratory work, usually amounting to 1 hour per week.
You are required to successfully complete 44 modules (132 credit hours), summarised as follows:

-	9	modules (University requirements)	(27 credit hours)	(20.45 %)
-	8	modules (Faculty requirements)	(24 credit hours)	(18.18 %)
-	21	modules (Departmental Compulsories)	(63 credit hours)	(47.73 %)
-	2	modules (Departmental Electives)	(6 credit hours)	(4.55 %)
-	4	modules (Supportive modules)	(12 credit hours)	(9.09 %)

These modules are listed in the following sections. The information given here is extracted from the Programme Specifications for the degree programme. The specifications are published separately, they can be found on the Department web site at www.philadelphia.edu.jo/it-cs.asp
Also, the description of each module can be found in the Undergraduate Course Catalogue on the web site at www.philadelphia.edu.jo/it-cs.asp

2.3 Module Organisation

2.3.1 Credit Rating

In the courses system, there are no pass requirements from one year of study to another. However, the total number of your successfully completed credit hours is only used to classify you in the corresponding year of study as shown below:

First Year	less than 30 credit hours
Second Year	between 30 and 59 credit hours
Third Year	between 60 and 89 credit hours
Fourth Year	between 90 and 132 credit hours

When you register for modules, you should follow the academic guidance plan that the Department arranges for you. In fact, you can register on any module only if you have taken its prerequisite(s) with the exception that you can register on the module and its prerequisite only if you are in the graduation semester.

In each semester, you can register for at least 12 credit hours and at most 18 credit hours, except for the semester in which you are expected to graduate when you can register for 21 hours. The complete four years academic guidance plan is listed in **Appendix A** of this Handbook.

The Department covers the Computer Science programme from the following areas:

1. **Programming Fundamentals**
2. **Theory / Languages**
3. **Architecture / Operating Systems**
4. **Net-Centric Computing**
5. **Intelligent Systems**
6. **Information Management**
7. **Human Computer Interaction / Graphics / Applications**
8. **Professional Practice**
9. **Project / Training / Special Topics**

The taught modules in each area are shown in Table (1), where each module is identified by a module number that consists of six digits according to the University numbering scheme. For example, the number of the module "Concepts of Programming Languages" is 750321. The numbering scheme is described in Figure (1).

Figure (1) Module Coding and Numbering Scheme

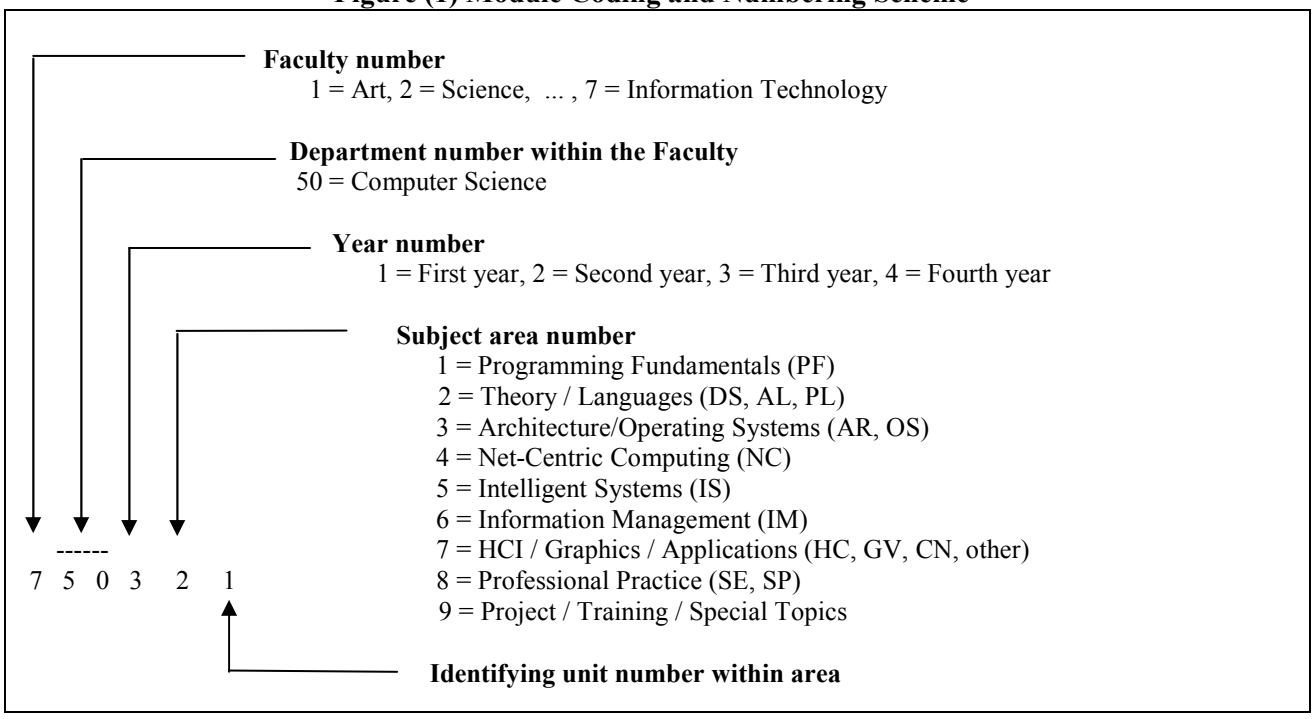

Table (1) Taught Modules in The Different Areas

A – The Compulsory Specialisation Modules		B- The Elective Specialisation Modules	
1. Programming Fundamentals		***1. Programming Fundamentals***	
750112	Programming Fundamentals	210101	Mathematics (1)
761211	Windows Programming	210103	Mathematics for Computing
721210	Introduction to Software Engineering	210104	Discrete Structures
750412	Advanced Programming		
2. Theory / Languages		***2. Theory / Languages***	
721120	Object-Oriented Paradigms	721321	Concurrent and Distributed Programming
721221	Object Oriented Data Structures		
750223	Theory of Computation		
750321	Concepts of Programming Languages		
750322	Design and Analysis of Algorithms		
750324	Compiler Construction		
750425	Advanced Data Structures and Algorithms		
3. Architecture / Operating Systems		***3. Architecture / Operating Systems***	
731332	Systems Analysis and Design	210231	Principles of Statistics and Probabilities
750231	Logic Circuits Design		
750232	Computer Architecture		
750333	Principles of Operating Systems		
750334	Advanced Operating Systems		
4. Net-Centric Computing		***4. Net-Centric Computing***	
721240	Computing Ethics	750441	Advanced Computer Networks
750444	Information and Computer Networks Security	761442	Advanced Web Programming
761340	Fundamentals of Computer Networks	761443	Wireless and Mobile Computing
5. Intelligent Systems		***5. Intelligent Systems***	
731150	Introduction to Information Systems and Technology	750452	Knowledge-Based Systems
750351	Fundamentals of Artificial Intelligence	750454	Neural Networks and Genetic Algorithms
6. Information Management		***6. Information Management***	
750461	Advanced Databases	761462	Information Retrieval
760261	Database Fundamentals		
7. HCI / Graphics / Applications		***7. HCI / Graphics / Applications***	
731270	Introduction to Web Programming	761373	E-Commerce Applications
761272	Multimedia Systems		
750474	Digital Image Processing		
8. Professional Practices		***8. Professional Practices***	
		750481	Software Engineering and Formal Specifications
9. Project / Training / Special Topics		***9. Project / Training / Special Topics***	
750398	Practical Training *	750491	Special Topics
750499	Research Project		

2.3.2 Modules Availability

The modules described here and in the Undergraduate Course Catalogue are those modules we expect to offer in the coming year. However modules may be cancelled if they are chosen by too few students or for other necessary reasons. The portfolio of modules is reviewed every year and the availability of a particular module in the coming year is not a guarantee of availability in subsequent years.

2.4 Programme Structure

The BSc Computer Science programme offers the opportunity for students to choose a study pathway which reflects their own changing and developing interests. It aims to develop strengths in both the principles and practice of Computer Science, and gives the opportunity for extensive practical work.
A graduate of this degree programme should have a good understanding of the architecture of hardware and software systems and the process of system design and will meet all the general aims of programme listed in section 2.1.1.

2.4.1 Module Choices

You may choose a module if you have already taken all its prerequisite modules and your personal tutor must supervise this choice. An initial choice is made before or at Departmental Registration.
You can choose modules according to the level of the modules as follows:

- **First Year**

In the First Year, you are encouraged to take 12 compulsory modules, 6 modules (18 credit hours) in each semester (first and second, the summer term is not taken into account). During each 16 weeks semester, you will normally attend 6 modules. Thus, each teaching week contains 18 hours or more of scheduled work. In addition, each scheduled hour typically requires two extra hours of unscheduled work (e.g. writing up lecture notes, preparing for a tutorial, finishing off a laboratory exercise etc.).
Five of the 12 modules of the first year are from the University requirements (UR), **two** from the Faculty requirements (FR), **three** from the supportive requirements (SR), and **two** from the Department requirements (DR) as shown below:

First Semester		
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 Semester		
111101	National Education	(UR)
----	University Elective (2)	(UR)
130102	English Language Skills (2)	(FR)
721120	Object-Oriented Paradigms	(FR)
731150	Introduction to Information Systems and Technology	(FR)
210103	Mathematics for Computing	(SR)

- **Second Year**

In the **Second Year,** the number and size of modules is similar to that of the first year. **One** of the 12 compulsory modules of the second year are from the University requirements, **five** from the Faculty requirements, **one** from the supportive requirements, and **five** from the Department requirements as shown below:

First Semester		
721240	Computing Ethics	(FR)
731270	Introduction to Web Programming	(FR)
210231	Introduction to Statistics and Probabilities	(SR)
721210	Introduction to Software Engineering	(DR)
721211	Object Oriented Data Structures	(DR)
750231	Logic Circuit Design	(DR)
Second Semester		
----	University Elective (3)	(UR)
761211	Windows Programming	(FR)
761272	Multimedia Systems	(FR)
750223	Theory of Computation	(DR)
750231	Computer Architecture	(DR)
760261	Database Fundamental	(DR)

- **Third Year**

In the Third Year, you should take six modules in the first semester and five modules in the second semester. **Eight** modules are from the compulsory Department Requirements, **one** departmental elective module, **one** module from the University requirements and **one** module form the Faculty requirements. One of the compulsory modules is the **Practical Training module**, which consists of realizing a supervised training in an industrial organization, or using distance online training. You should take this module in the first semester.
Note that the elective modules offered by the Department that you could select during the third and fourth years would help you to choose a particular path of interest to you, e.g. Artificial Intelligence, Computer Networks, etc.

First Semester		
-----	University Elective (4)	(UR)
731332	Systems Analysis and Design	(DR)
750321	Concepts of Programming Languages	(DR)
750322	Design and Analysis of Algorithms	(DR)
750333	Principles of Operating Systems	(DR)
761340	Fundamentals of Computer Networks	(DR)

Second Semester		
------	University Elective (5)	(UR)
750324	Compiler Construction	(DR)
750334	Advanced Operating Systems	(DR)
750351	Artificial Intelligence	(DR)
750398	Practical Training	(DR)

- **Fourth Year**

In the **Fourth Year,** you should take nine modules. In the first semester, you must select **one** departmental elective module, the Graduation Project module, and **two** compulsory modules that are all from the Department requirements. In the second semester, you must take **one** University elective module and **four** modules from the compulsory Department Requirements as shown below. The selection of a University elective module (one module) depends upon your choice.

First Semester		
750412	Advanced Programming	(DR)
750474	Digital Image Processing	(DR)
750499	Research Project	(DR)
------	Department Elective 1	(DR)
Second Semester		
111100	Military Sciences	(UR)
750425	Advanced Data Structures and Algorithms	(DR)
750444	Information and Computer Networks Security	(DR)
750461	Advanced Databases	(DR)
------	Department Elective 1	(DR)

2.4.2 Modifying Module Choices

After setting your plan and register on modules as described in section 2.4.1, you can make changes on your choices as follows:

- In each semester, one week after lectures start (three days for summer semester), you can add or withdraw modules. Normally, no changes of modules will be permitted after these dates except for the withdrawal mentioned below.

- In the first instance, you should discuss any plan to change modules with your primary tutor. You must check that the new module you wish to take is a valid option for your degree program and find out if there are likely to be any timetable problems. If there are timetable clashes this will probably prevent you from changing module.

2.4.3 Programme Characteristics

The following are the main characteristics of the programme:

- ***Elaboration on Content and Emphasis of Practical Components of Modules.*** Most of the modules contain practical work that makes you use current software tools and computing technologies. Thus, the practical part of modules accounts for at least 25% of the total number of hours. Many laboratory assignments are given during the semester through which you can practice what you have learned from the theoretical part of the module, or develop your skills in using most recent software tools and programming languages. For example, the practical works in "Computer Skills (2) for Scientific Colleges", "Windows Programming", and "Object-Oriented Paradigms" modules emphasis on problem solving via Visual Basic, C++, and Java languages. However, the practical work in Operating System module is concerned with inter-process communication, while in Computer Networking it is concerned with client server applications and simulation of OSI protocols. Besides the necessary stress on practical components in various modules, you also undergo practical training and undertakes graduation project. These two combined help you to get the necessary professional exposure required in the industry domain.

- ***Supervised Work Experience (Practical Training Module).*** This attends to the Practical Training module in year 3. This module adds a new flavour to the coursework you have to go through before earning the degree. In order to ensure that practical training has rigorous implementation that complies with University Code of Practice, we have set up some important regulations to emphasize the educational value of the training. The Department and Faculty Councils approve these regulations**.** You are placed in industry and work two days per week at the work place. Your training is jointly supervised by industry and University supervisors. The supervision is through visits and liaison.

- ***Graduation Project Module:*** The Final Year Project is an important integrative module, which invites you to apply your knowledge, skills, and academic ability to a specific problem. The project demands skills in researching materials, verbal and written communications and encourages you to tackle problems, which simulate industrial situations. The time allocated to the project is one to two semesters.

3. TEACHING, LEARNING AND ASSESSMENT

3.1 Work and Attendance

The University regulations governing the Work and Attendance of students are given in the Student Guide 2006/2007. Full attendance is required at all lectures, laboratories, and any tutorials, which may be scheduled. Completed laboratory work should be handed in on time. Attendance at laboratories and at many lectures is monitored and attendance registers kept. Please note that the expectation is that you will be required to undertake approximately thirty six hours per week of study i.e. an average of two hours private study will be required for every scheduled hour of lectures, laboratories etc. and some of you may require much more time than this. Being a full time student means that your attendance is mandatory and absence for holidays is not permitted in term-time. The experience of the Department confirms that lack of attendance leads to study problems and if you have problems you should consult your subject tutors or personal tutor. In addition, failure to attend can result ultimately in refusal by the University to allow you to sit in the degree examinations. The duty of the lecturer is to keep continuous review of the work and attendance of

the students with whom he/she is concerned. If the rate of your absences in a module is greater than 15% (or 20% for student representing the University in sportive or cultural activities) of the completely accredited hours and you have no acceptable justification, then you are excluded from that module. If the Dean of the Faculty accepts the justifications of absence, then you are mentioned as ***withdrawn*** without refunding the registration fees. A formal process is defined to tackle the problem of any student whose work and attendance appear unsatisfactory. Direct approaches by lecturer to solve the problem are as follows: He/she may choose to issue an "informal" warning, which has a precisely defined format and permits recovery of the situation. If this is unsatisfactory, a "formal" warning is issued. This is again of a precisely defined format. Failure to recover the situation at this stage leads to an exclusion from the course. A copy of this correspondence is held in a student's file.

3.2 Assessment

3.2.1 Examinations

In each semester, there are two 1-hour mid term exams and one final 2-hours exam (at the end of the semester). For the mid term exams, the lecturer returns to you, after one week of the examination time, your corrected answer sheet marked with some feedback for you to check. Whereas the final exam is unseen exam and you can obtain your marks from the Admission and Registration Office or directly from the University web site at most after 72 hours of the examination time.

At the end of each semester, the timetable of the final exam of the next semester is set by the Admission and Registration Office to help and guide you in choosing your modules for the next semester. The two mid term exams are set by the Department and the syllabus of each module contains their timetable. The lecturer of the module will also inform you about this timetable in the first lecture of the semester.

For the graduation project module, you should submit your final project report to the Department in the fourteenth week of the semester. In the fifteenth week, a committee will assess your project work, report, and presentation.

3.2.2 Role of Internal and External Examiners

For each module, the Department assigns a module coordinator and an internal examiner who is one of the senior staff members. If many lecturers teach the same module concurrently, they should suggest exam questions (for the first, second and final exams) and run the same exam for all sections. The main coordinator of the module will collect these questions from lecturers and select some of them to be in the exam paper. The internal examiner moderates the exam paper.

On the other hand, external examiners validate the standard of degree program. The external examiners are expected to look at the question papers, inspect a selection of scripts and project reports (particularly those on borderlines). They supply an assessment report to the Department.

3.2.3 Criteria for Assessing Examination Work

First class (90 – 100 marks)*:* First class answers demonstrate depth of knowledge or problem solving skills, which is beyond that expected from a careful and conscientious understanding of the lecture material. Answers will show that you

- have a comprehensive knowledge of a topic (often beyond that covered directly in the program) with an absence of misunderstandings;
- are able to apply critical analysis and evaluation;
- can solve unfamiliar problems not drawn directly from lecture material and can adjust problem solving procedures as appropriate to the problem;
- can set out reasoning and explanation in a logical, incisive and literate style.

Upper Second Class (80 – 89 marks)*:* Upper second class answers provide a clear impression of competence and show that you

- have a good knowledge base and understanding of all the principal subject matter in the program;
- can solve familiar problems with ease and can make progress towards the solution of unfamiliar problems;
- can set out reasoning and explanation in a clear and coherent manner.

Lower Second Class (70 – 79 marks)*:* Lower second class answers will address a reasonable part of the question with reasonable competence but may be partially incomplete or incorrect. The answer will provide evidence that you

- have a satisfactory knowledge and understanding of the principal subject matter of the program but limited to lecture material and with some errors and omissions;
- can solve familiar problems through application of standard procedures;
- can set out reasoning and explanation which, whilst lacking in directness and clarity of presentation can nevertheless be followed and readily understood.

Third Class (60 – 69 marks)*:* Third class answers will demonstrate some relevant knowledge but may fail to answer the question directly and/or contain significant omissions or incorrect material. Nevertheless, the answer will provide evidence that you

- have some basic knowledge and a limited understanding of the key aspects of the lecture material;
- can attempt to solve familiar problems albeit inefficiently and with limited success.

Pass (50 – 59 marks). Answers in this category represent the very minimum acceptable standard. Such answers will contain very little appropriate material, major omissions and will be poorly presented lacking in any coherent argument or understanding. However the answer will suggest that you

- have some familiarity with the general subject area;
- whilst unable to solve problems, can at least formulate a problem from information given in a sensible manner.

3.2.4 Appeal Procedures

If you have good reason to question a mark you have been given (in midterm exams or in coursework), you should in the first instance approach the module lecturer. If the problem is not solved, you must submit it to your primary tutor. He/she will find the appropriate solution with administrative structures.

Problems with final examinations are resolved by submitting complaints or appeals in writing (within three days of the announcement of examination results) to the Department. Such requests are forwarded to the Examination Committee of the Faculty. The Department and the examination committee will consider these cases and checks if there is any mistake in the summation of the marks and so on.

3.2.5 Unfair Practices

The University treats attempting to cheat in examinations severely. The penalty is usually more severe than a zero in the paper concerned. More than one student were dismissed from the University because of this. Plagiarism, or copying of course or lab work, is also a serious academic offence as explained in the University guidelines. In Computer Science Department these guidelines apply also to laboratory exercises.

3.2.6 Department Guidelines on Plagiarism

1. Coursework, laboratory exercises, reports, and essays submitted for assessment must be your own work, unless in the case of group projects a joint effort is expected and is indicated as such.

2. Unacknowledged direct copying from the work of another person, or the close paraphrasing of somebody else's work, is called plagiarism and is a serious offence, equated with cheating in examinations. This applies to copying both from other students' work and from published sources such as books, reports or journal articles.
3. Use of quotations or data from the work of others is entirely acceptable, and is often very valuable provided that the source of the quotation or data is given. Failure to provide a source or put quotation marks around material that is taken from elsewhere gives the appearance that the comments are ostensibly your own. When quoting word-for-word from the work of another person quotation marks or indenting (setting the quotation in from the margin) must be used and the source of the quoted material must be acknowledged.
4. Paraphrasing, when the original statement is still identifiable and has no acknowledgement, is plagiarism. A close paraphrase of another person's work must have an acknowledgement to the source. It is not acceptable for you to put together unacknowledged passages from the same or from different sources linking these together with a few words or sentences of your own and changing a few words from the original text: this is regarded as over-dependence on other sources, which is a form of plagiarism.
5. Direct quotations from an earlier piece of your own work, if not attributed, suggest that your work is original, when in fact it is not. The direct copying of one's own writings qualifies as plagiarism if the fact that the work has been or is to be presented elsewhere is not acknowledged.
6. Sources of quotations used should be listed in full in a bibliography at the end of your piece of work.
7. Plagiarism is a serious offence and will always result in imposition of a penalty. In deciding upon the penalty the Department will take into account factors such as the year of study, the extent and proportion of the work that has been plagiarized, and the apparent intent of the student. The penalties that can be imposed range from a minimum of a zero mark for the work (without allowing resubmission) through caution to disciplinary measures (such as suspension or expulsion).

3.3 Assessment Regulations

Most modules have some continuous assessment, such as assignments, essays, tutorials, laboratory exercises, seminars, and examinations. Assignments and any coursework must be submitted by the due dates and any submission after these dates will not be assessed. The proportions of coursework and examination are set out in the detailed syllabus for each module.

The examination and continuous assessment marks are combined to form a single mark out of 100 for each module. This mark is divided as follows: 50% of the total mark is given for two 1-hour midterm exams, coursework and/or seminars, projects, or essays, and 50% for the final exam that may be a written exam only or a written exam plus final laboratory exam (if applicable), final small project, or seminar presentation. The 50% of the final exam is from the University regulations. The minimum pass mark is 50% for any module.

When you do not sit for the final exam without any excuse, you will either get the "University zero" (i.e. 35%) if your collected mark during the term was less than or equal 35%. Otherwise, you will retain your collected mark. In both cases, you have to reenroll in this module and study it again.
On the other hand, if you have a certified excuse approved by the lecturer, the Department Head, and the Faculty Dean, then you can submit a request for "incomplete" that lets you sit for the exam, which is normally held at the first two weeks of the semester that follows.

On the other hand, a committee of three staff members including the supervisor of the project assesses the graduation project module. The project's assessment includes the supervisor mark (35%) and the discussion committee mark (65% given as follows: 20% for project presentation, 25% for report writing, and 20% for defendant discussion).

3.4 Supervised Work Experience

This attends to the Practical Training module in year 3. The Department and Faculty Councils approve the regulations for training**.** The Practical Training Committee in the Department has responsibility for industrial placements and advertises any contacts from industry giving opportunities for vacation placements for training. You register for the practical training module as usual module but you have to arrange your timetable to include at least two free days to get your training. You should complete 160 hours in the trainee company. Students placed in industry are jointly supervised by industry and University supervisors. The supervision is through visits and liaison.

For the practical training module there is no 100% mark but only you will get "pass" or "fail" in this module according to the following rules. You should submit a technical report of your training, and a team of academic staff members makes several observations on the trainers' work in their place of training. Then according to the observations and the report, they assess you. For more information on the training, you can consult the Faculty web site www.phildelphi.edu.jo/it.sp

3.5 Awards

The Faculty prize for the graduate student with the first highest grade in the cohort (200 JD).
The Faculty prize for the graduate student with the second highest grade in the cohort (200 JD).
The Faculty prize for the best graduation project in the cohort (250 JD)

4. STUDENT PROGRESSION

4.1 Progression

To pass the degree, you need to successfully complete 44 modules of different requirements; University, Faculty, Department, and supportive. The pass mark of any module is 50%. Your progress in the programme is measured according to the number of credit hours that you have successfully completed. The level (year) in which you are in depends on that number of credit hours. Another thing which is vital for your assessment and progression is the accumulative average that should be at least 60% in each semester. Consequences of unsatisfactory progress may include:

- Failure to progress to the next year,
- Failing to graduate,
- dismissing from the programme

If you failed in some modules, you cannot be considered in the next level. However, this does not prevent you from taking modules of the next level as long as you have taken their prerequisites.

Failing in a compulsory module means that you have to register on this module in the next semester. This can be repeated three times until you pass the module. If you failed to pass the module in the third time, then you have a choice to take an alternative to it only if you are in the graduation semester. However, if the module that you failed to get 50% was an elective module, then either you register on the same module in the next semester or take another elective to substitute it.

You have to pay attention to your accumulative average that should be not less that 60%. You will be warned if you could not obtain the 60% in each semester. In this case, you are encouraged to repeat studying those modules with low marks in order to increase your accumulated averages. Note that, repeating modules may delay your graduation so you may graduate in more than four years. The maximum allowed period for you to stay in the University is seven years. However, you will be dismissed from the programme if this average is not achieved in the third attempt.

You can graduate and pass the degree if you have successfully completed all Degree requirements and your accumulated average is at lest 60%. Failing to get average of at least 60% in the graduation semester means that you could not be graduated and you have to register in the next semester to repeat some modules with low marks in order to achieve the required average.

The average is graded as follows:

84% - 100%	Excellent
76% - < 84%	Very good
68% - < 76%	Good
60% - < 68%	Fair

4.2 Change, Interrupt, Withdraw, and Transfer from the Programme

4.2.1 Changing Your Choice of Modules

You can change your choice of modules as described in section 2.4.2.

4.2.2 Interruption of Degree Programme

Any interruption (taking at most 2 years) of your degree programme requires special permission from the Faculty. Regulations state that a B.Sc. degree is a continuous 4-year period of study. Permission will only be granted if satisfactory reasons are given. A written case with supporting evidence must be presented to the Faculty. Reasons might include prolonged illness. Consult your tutor for advice.

4.2.3 Withdrawal from Modules

There is a late withdrawal from a module with loosing its fees. If you are contemplating withdrawing from a module, please discuss the situation with your personal tutor at the earliest opportunity. You should follow the following University regulations in this context:

- You can withdraw a module at most during the thirteenth week of the first or second semester and at most during the seventh week of the summer semester.
- The minimal number of modules (which is 9) required in each semester should be followed.

4.2.4 Transfer between Departments

- If you are contemplating any change of Faculty or Department, consult your primary tutor as soon as possible.
- You can change your Department by filling a special form at the beginning of the semester. It is only required that the Tawjihi average imposed in the new faculty or department must be less than or equal to your Tawjihi average. A specialized committee will decide what courses will be retained from your actual Department.

5. STUDENT SUPPORT AND GUIDANCE

5.1 Deputy Dean Office

The Deputy Dean Office (Room IT 604) is mainly for students advisory services. It deals also with all routine undergraduate enquiries. Problems, which cannot be dealt with by the Deputy Dean, will be referred to an appropriate person in the Department or University.

5.2 Academic Guidance

All new students should have academic (personal) tutors. The new students are grouped into 20 – 30 students groups and each group is assigned to an academic staff member who will be their academic tutor for the four years. The students remain with the same tutor till their graduation. The tutor deals with all routine undergraduate inquiries, advises for academic registration at the beginning of each semester, and any other raised problems. However, problems, which cannot be

dealt with by the tutor, will be referred to the Head of the Department, the Dean of the Faculty, or to an appropriate member of academic staff. The academic guidance is available on specified dates in the terms, and any advisory service offered by the Deputy Dean is available daily to all students in the Computer Science Department (including both Full- and Part-time students).

Time: 11:00 AM to 07:00 PM Sunday to Thursday during term,
Venue: Room IT 604 (for all students)

The advisory service offers advice on departmental and University matters and helps with anything that concerns you, whether in your studies, in the Department, in the University, or in your life outside the university. The advisor is available with knowledge of the Department and University and who is willing to listen and help with whatever you bring. Note that

- All visits to the advisory service offices are strictly confidential.
- If you have difficulties with material on particular course units you should normally first approach your tutors (or lecturers/project supervisors). You may also consult your tutors on matters that are more general but you can equally well call in at the Deputy Dean Offices.
- If you have health problems, you are welcome to consult an advisor in the Department but may prefer to go directly to your doctor or to the University Clinic.

Feel free to make use of these services at any time on any matter.

5.3 Students Affair Deanship

Confidential, individual counselling on any matter affecting personal well-being or effectiveness is available at the Philadelphia University Students Affair Deanship. The Deanship sees well over a hundred students a year and gives expert advice on problems such as low motivation, personal decision making, relationships, and anxiety and family difficulties. People there, are willing to help in finding fresh ways of coping with the emotional and personal aspects of problems and seeks to do so in a collaborative, straightforward and empowering way with the individual concerned. Advice is available concerning referral to other services, helping others and dealing with common student problems such as exam anxiety.

The Deanship is open from 8.00 AM to 4.00 PM, from Sunday to Thursday throughout the year and appointments can be made by calling into the office of the Dean of Students affairs. All inquiries will be treated confidentially.

5.4 Tutoring Arrangements

Some of your modules will have tutorials, where you can discuss topics on a module and run through exercises. Usually, the lecturer of the module runs the tutorial. There will be an opportunity for you to ask questions on matters you do not understand.

As you have a personal tutor from the beginning of your University life, your tutor is here to help you in your way through University life. He/she will watch your progress and offer help and advice wherever necessary. If you get into difficulties, you should contact your personal tutor or visit the Deputy Dean at the earliest possible opportunity. Do not let things slide until it is difficult to retrieve the situation, especially if you are getting behind with your work. Your personal tutor will also advise on your choice of modules, on departmental or University procedures and will provide references for jobs and other purposes.

Course lecturers are always available to discuss questions or problems with the module material. Each lecturer fixes at least six office hours on his timetable, which is fixed on his office door. You can call at these hours. For any reason, if these lecturers could not see you at these office hours, they may arrange an appointment at another time. It is important that any matter that affects your ability to work is notified to the Department - through your personal tutor, through the Deputy Dean or otherwise. The following are examples of matters that may affect your work: illness, personal or family difficulties (including illness in the family) or financial problems. In assessing your performance, the Department has a policy of trying to compensate for difficulties you have encountered whilst studying. We can only do this if we are notified of difficulties and have some idea of their extent.

5.5 Student Presentation and Feedback

5.5.1 Staff Student Liaison Committee

At each academic year, the Department forms a staff student liaison committee that is composed of student representatives who are elected from different levels and three staff members. The committee meets at least twice each semester and may discuss any matter of concern which cannot be resolved informally. The staff members of the committee are members of the Department and principally are the academic tutors.

Feedback from students on modules and teaching is important to us, particularly for the role it plays in ensuring and enhancing the overall quality of the programme. The objectives of this committee are:

- to provide a unique forum of staff and students for the discussion of new ideas and for solving problems;
- to form the basis for the representation of students' views within the department;
- to take students' opinion on academic matters including degree programme and syllabuses and form part of the Department's quality assurance and enhancement procedures;
- to provide an opportunity for students to learn about and contribute to the development of quality assurance and enhancement procedures in their Department

5.5.2 Module Coordination Committee

Sometimes the number of students enrolled in a module could be large, so this number is divided into more than one section (class) and theses classes could be run be more than one lecturer. Such modules need coordinators to coordinate between different classes. For each class, a student representative is elected by the class students to be a member of the module coordination committee that contains also the lecturer of each class of that module. At the beginning of each semester, the Department issues a list of module coordinators. The module coordination committee meets at least twice per semester to coordinate everything related to that module. The main objectives of this committee are:

- To ensure that all classes have the same syllabus
- To follow the same timetable in delivering the course material
- To unite the examination
- To get feedback from students' representatives and use it to improve the quality of teaching
- To use feedback in module monitoring

5.5.3 Departmental and Deanship Meetings

The meetings, held by the Head of Department and the Dean of the Faculty during term time, has mainly an advisory role, where students may raise their problems that need some concern from these authorized persons. These meetings are held separately for each year students.

5.5.4 Collecting and Analysing Feedback

The Faculty in general and the Department in particular attach great importance to the opinion of students on the quality of the teaching provided. At the thirteenth week of each semester, every student is asked to complete a Module Evaluation Questionnaire for each module. The questionnaires are anonymous. Final Year students are also given another questionnaire on which they can comment on their degree programme as a whole.
The Departmental Quality Assurance and Enhancement Committee which is responsible for the quality of teaching in the Department, usually makes the analysis of these questionnaires and uses the result to monitor the teaching process and the progremme as a whole.

6. FACULTY AND DEPARTMENTAL LEARNING RESOURCES

6.1 Learning Resources Centre

Photocopy facilities are available in the Learning Resource Centre, room 103, Tel. 2453. Reference copies of textbooks are available for consultation. Copies of previous weeks' tutorial solutions are also available. The resource centre holds non-loan copies of undergraduate textbooks. Lending copies of textbooks are available in the University Library.

6.2 Code of Practice for Computer Usage

At registration, you will be required to assent to the following departmental code of behaviour, which relates to the responsible use of Computer equipment. Misuse of the facilities is regarded as serious disciplinary offences.
This code of practice is supplementary to University regulations concerning the use of computing equipment to which you are required to assent at Registration.

1. Every student is allocated one PC in every laboratory session. But for UNIX laboratory, you have been allocated one or more usernames for your own personal use: you must not use other usernames or permit other people to use your username. You must not use computers to which you have not been granted access, or attempt to access information to which you have not been granted access.
2. You must not deliberately hinder or annoy other computer users.
3. You must not use machines belonging to the Department for commercial purposes without the prior written permission of the Head of Department. You must not sell the results of any work you do using Departmental facilities without the prior written permission of the Head of Department.
4. You must not write or knowingly store, on machines belonging to the Department, software that, if executed, could hinder or annoy other users, except with the prior written permission of the Head of Department.
5. You must not make an unauthorized copy, in any form, of copyright software or data.

6. You must not store personal information, except in a manner permitted by the Data Protection.
7. You must follow all rules, regulations and guidelines imposed by the Faculty of IT and the University in addition to the Department's Code of Practice.

Explanatory Notes

The following notes indicate ways in which the Code of Practice applies to undergraduates for use of computers. It is not intended to be a complete list of possible abuses of the equipment. Each note refers to the corresponding paragraph above.

1. Undergraduate students are not normally granted access to the computers in the network, or to other students' files. You should not attempt to use another student's account even if they have not set a password. Of course, it is still important to set a password for your own privacy and security.
2. This will be interpreted very broadlya as:
   - Tampering with another user's files.
   - Tampering with another user's screen.
   - Setting up processes which persist after you log out and annoy subsequent users of the machine.
   - Broadcasting of offensive messages.
   - Display or storage of offensive pictures.
   - Abuse of the mail system.
   - Occupying a machine to play games while other students need it to do their laboratory work.
3. Clearly, the Head of Department would have to be convinced that any such use of the machines would not conflict with their primary purpose.
4. Note carefully that this means you are not allowed to write or introduce a virus program, even if it is never executed.
5. Note that this does not prevent your taking copies of your laboratory work home, or making copies of non-copyright material, but does prevent your taking random pieces of software away on a floppy. You should assume that all material is copyright unless it specifically states otherwise. If in doubt, ask.
6. Personal information includes names, addresses, mailing lists, etc. You should contact the Data Protection Officer, Mr. Mustafa Abu Masoud, if you need to store such information.
7. In fact, you agreed to abide by the University and Faculty rules when you registered.

Please direct queries concerning the code of practice to Department Head.

Support for Computer Equipment

Students are encouraged to own their own machines. Please note, however, that you are NOT REQUIRED to own your own computer. The Department has excellent facilities and undergraduate students are allowed to use the facilities provided in the building of the Faculty of Information Technology. Whenever the buildings are open between 08 AM and 07 PM, access is also allowed in this range of time, from Sunday to Thursday during term.

6.3 Other Resources and Facilities

There are many different resources and facilities that you can utilize. These are:

- **Photocopying**
  Out of the library, photocopy may be done at different Bookshops, on an affordable cost.

- **Printing**
  You can take printout (free of charge) in any lab of the Department. Each lab contains at least two printers for this purpose.

- **Administrative Infrastructure**
  It is composed of six offices (Dean, 1 Advisory service, Dean Secretary, and Department's Chair, Department Secretary, and Meeting Room).

- **Academic Infrastructure**
  It is composed of
  - 16 Department classrooms plus some other classrooms shared with other faculties and one lecture theatre equipped with support facilities: computer, data show, overhead projector.
  - 4 laboratories (each contains 20 to 25 PCs or Monitors and 1 to 2 printers): Windows NT Laboratories, Internet Laboratories, SunRay1 UNIX Laboratories, and Sun Sparc UNIX Laboratory. The Department also shares some other laboratories with other departments.
  - 5 staff offices where each staff member is supplied with a PC.
  - 1 room for staff meeting
  - 1 office for the student's guidance and examination committee.
  - 1 Base room.
  - Two e-learning centres (Avicenna, Phoenix) shared with other departments in the Faculty and other Faculties.

- **Lecture Support Facilities**
  In the Department, there are 4 overhead projectors and 7 data shows used to support modules and seminars presentations.

- **University Computer Centre**
  This centre provides the Department with training and maintenance facilities.

- **Networking Facilities**
  Ethernet*:* The PCs in each laboratory are connected to an Ethernet platform 10/100 Mbps.
  Intranet: All computing facilities of the University are connected to a Gigabit Intranet backbone.
  Internet: The University is connected to the Internet by 4 Mbps lines.

- **Type and Level of Access**
  For communication, computing, or information searching, the Department provides free access to networking facilities at any time for the staff and the students.

- **Library Infrastructure**
  This structure includes the University Main Library, which provides students and staff members with the required recent text and references books, journals, and CD ROMs. According to its collaboration and co-ordination program, it has relations with more than 120 universities and scientific organisations. It opens from 08:00 AM to 07:00 PM. It includes:

- ***Conventional Library*****,** which contains books and journals. The books hall contains more than 2226 different English titles in computing, where more than 11% are edited in years 2005-2006 and few are published in 2007. The room of journals contains 30 computing journals that are useful for research and teaching.

- ***Electronic Library*****,** which contains 2000 CD ROMs and 220 floppy disks for the taught programming languages courses and module support tools, such as self-study packages. It has access to approximately 800 universities electronic libraries via the World University Library that is endorsed by the United Nation University. The World University Library has six databases that contain more than 4674 periodicals available online. The online resources in the electronic library include sites that list more than 50000 online books and access to online libraries and encyclopaedias and other databases on the Internet.

- ***Internet Access Service,*** available in a room containing 20 PCs.

- **Bookshops:** contain books, exercises with solutions, solutions to previous examinations and so on.

- **Self Study Facilities**

  The self study facilities include the following:
  - The Faculty Learning Resource Centre, as mentioned before.
  - The Electronic Library as mentioned before.
  - The Department Web/Intranet provides you with all relevant information such as:
    - Undergraduate Handbook (this handbook)
    - Programme Specifications
    - Lectures and course notes.
    - Bulletin board for messages and general use.

    This provides you with a rich "one stop" learning environment.
  - Distance learning has been implemented through agreement with Phoenix International and through a project financed by UNESCO.
  - Disabled students' facilities. The University has appointed an equal opportunity officer to help and assess the needs of any physically disabled student.

- **Training Facilities**
  - The University has signed an agreement with Phoenix International for distance learning, which is used as a support for the practical training module.
  - The University has signed a licensed grant with Microsoft allowing the University to use Microsoft software. In addition, the agreement allows one person to be trained on Microsoft products.
  - The University has signed an agreement with Cisco Systems, which authorizes our Department to have access to the Cisco Networking Academy programme. The following Cisco certifications will be provided: (Certified Network Associate, Cisco Certified Network Professional, and NetPlus+).
  - The University is an ICDL Accreditation Test Centre (UNESCO International Computer Driving License).
  - The University is in the process to sign an agreement under the program "SUN Academic Initiative (SAI)" to provide supporting educational and training materials.
  - The University is in the process of establishing a training centre for awarding Microsoft certifications.
  - The University is in the process of establishing virtual labs that can be used for training.

- **Incubator Lab**
  This lab is a result of feedback from students and staff. The main purpose of the lab is to encourage a focus for new ideas, industrial applications etc. so that the staff, students and Industry can have a common forum and facility. Two projects were commenced in this context.

- **Special Help Tutorial Room**
  Students having problems in some modules may meet specialist lecturers in this room. Specific and directed tutorials may help them.

- **Careers Advisory Service**
  This service provides information for students and graduates of the University.

- **Extracurricular Activities**
  The University provides some entertainment for the students to enrich their talents in their free time. This includes

  - **A Deanship of Student Affairs** that organises the social, cultural, and sport activities for the students in the University. It has also an alumnae office that keeps track of the graduate's information and news.
  - Several spaces for different sports.
  - Several spaces for cultural activities.
  - Several common rooms for meetings, snacks, and cafeterias.
  - Four Internet cafes each one containing 10 PCs.
  - One Students Club.

6.4 Communications

- **Electronic Mail**
  Electronic mail is used widely for administrative purposes within the Department. It is frequently useful for communicating between individuals and small groups (e.g. between a tutor and his/her tutorial group), and occasionally for broadcasting important messages to wider groups. It is important that you know how to use email. It will be covered in the introductory laboratory sessions. The code of practice for computer usage covers electronic mail, please note the points below.

- **Obscene or Offensive Mail**
  DO NOT SEND OBSCENE OR OFFENSIVE MAIL. If you receive mail, which you regard as offensive or obscene, you may wish to complain to a member of staff so that appropriate disciplinary action can be taken against the offender.

- **Group Mailing**
  You are strongly discouraged from sending email to groups of people. The newsgroups should be used for this purpose.

Appendix A

The Academic Guidance Plan

Philadilphia University **Facalty of Information Technology**

Guidance Plan for the Computer Science

2009 / 2010

2010

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)	721240	Computing Ethics	731150	(FR)
		731270	Introduction to Web Programming	750112	(FR)
		210231	Introduction to Statistics and Probabilities	------	(SR)
		721210	Introduction to Software Engineering	731150	(DR)
		721211	Object Oriented Data Structures	721120+210104	(DR)
		750231	Logic Circuit Design	-----	(DR)
	Second (18 Credit Hours)	----	University Elective (3)	----	(UR)
		761211	Windows Programming	721120	(FR)
		761272	Multimedia Systems	-----	(FR)
		750223	Theory of Computation	210104	(DR)
		750231	Computer Architecture	750231+731150	(DR)
		760261	Database Fundamental	721211	(DR)
Third	First (18 Credit Hours)	-----	University Elective (4)	-----	(UR)
		731332	Systems Analysis and Design	760261 (or concurrently)	(DR)
		750321	Concepts of Programming Languages	721211	(DR)
		750322	Design and Analysis of Algorithms	210103+721211	(DR)
		750333	Principles of Operating Systems	750232	(DR)
		761340	Fundamentals of Computer Networks	721211	(DR)
	Second (15 Credit Hours)	------	University Elective (5)	------	(UR)
		750324	Compiler Construction	750223	(DR)
		750334	Advanced Operating Systems	750333	(DR)
		750351	Artificial Intelligence	721211	(DR)
		750398	Practical Training	721210	(DR)

Fourth	**First (12 Credit Hours)**	750412	Advanced Programming	761211	(DR)
		750474	Digital Image Processing	750322	(DR)
		750499	Research Project	750398	(DR)
		------	Department Elective 1	------	(DR)
	Second (15 Credit Hours)	111100	Military Sciences	------	(UR)
		750425	Advanced Data Structures and Algorithms	750322	(DR)
		750444	Information and Computer Networks Security	761340	(DR)
		750461	Advanced Databases	731332	(DR)
		------	Department Elective 1	----	(DR)

(UR) University Req. **(FR) Faculty Req.** **(DR) Dept. Req.**
(SR) Supporting Req.

Appendix B

Modules outlines

2009/2010

731270, Introduction to Web Programming

Providing Department: Management Information Systems, Faculty of IT
***Module Coordinator(s)*:**
Year: 2
***Credit*:** 3 credit hours
Prerequisite: 750112

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.

Teaching Methods: 32 hours Lectures (2 per week) + 8 hours Tutorials (1 per 2 weeks) + 24 hours Laboratory (1-2 per week)

Learning Outcomes:
On completing this module you should be able to:

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

Assessment of Learning Outcomes
Learning outcome (1) is assessed by examinations and tutorials. Learning outcomes (2 -3) are assessed by

examinations, tutorial, and assignments. Learning outcome (3) is assessed by examinations, tutorial, and assignments.

Contribution to Programme Learning Outcomes:
A2, A3, A4, A5, B1, C1, C4, C5, C6.

Synopsis*:* Internet and Intranet Technology: Concepts, protocols, Services, and architecture, TCP/IP Architecture and Protocols (Client & Server), DNS, Internet Service Providers (ISP), Internet Services: USENET News, E-Mail, FTP, and Telnet; The Web: Basic Concepts, WWW and Web Servers, Links: Hyperlinks & Hypermedia, Web pages and home pages, Browsers & Search Engines; Introduction to Markup Languages; Editing HTML, HTML Tags: Headers, HTML Tags: Text Styling and Formatting, and linking; HTML Tags: Images and Image maps; Basic HTML Lists and Tables; Basic HTML Forms and Frames; Frames and Cascading Style Sheets; Cascading Style Sheets and Introduction to Client Scripting; Simple JavaScript Programs; JavaScript: Control Structures, if, if/else, While, for, and switch. JavaScript: Break and Continue statements; JavaScript: Functions, Arrays.

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

Textbooks and reference books:

1- Deitel & Deitel, Internet and World Wide Web How to Program, Prentice Hall, 3rd Edition, 2004.
2- Douglas Comer, Computer Networks & Internets, Prentice Hall, 2003
3- Brian Salter, A simple guide to HTML, Prentice Hall, 2002.
4- Ellenn Behoriam, "HTML and XHTML: Creating Web Pages", Prentice Hall, 2002
5- Gary Rebholz, "How to Use HTML & XHTML", Sams, 2001
6- Ellie Quigley, "JavaScript by Examples", Prentice Hall, 2004
7- Tom Negrino, "JavaScript for the World Wide Web: Visual QuickStart Guide", Student Edition, 5/E, Peachpit Press, 2004
8- Susan Anderson-Freed, "Weaving a Website: Programming in HTML, Java Script, Perl and Java", Prentice Hall, 2002

Website(s):

1. www.w3schools.com
2. www.webteacher.org.
3. www.microsoft.com.
4. www.whatis.com.
5. www.idocs.org.
6. www.w3.org.
7. www.webdeveloper.com
8. www.javascriptmall.com
9. www.javascripts.com/toc.cfm
10. www.Deitel.com

750231, Logical Circuit Design

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 1
Credit: 3 credit hours
Prerequisite: 750101

Aims: This module introduces you to the design and implementation of digital circuits. Topics include: combinational and sequential circuit analysis and design, digital circuit design optimization methods using random logic gates, multiplexers, decoders, registers, counters and programmable logic arrays. Laboratory experiments will be used to reinforce the theoretical concepts discussed in lectures. The lab experiments will involve the design and implementation of digital circuits. Emphasis is on the use computer aided tools in the design, simulation, and testing of digital circuits.

Teaching Methods: 41 hours Lectures (2-3 per week) + 4 hours Tutorials (1 per 3 weeks) + 3 hours Laboratory (1 per 4 weeks)

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

1. Define the problem (Inputs and Outputs), write its functions. (A, B, C)
2. Minimize functions using any type of minimizing algorithms (Boolean Algebra, Karnaugh map or Tabulation Method). (A, B)
3. Implement functions using digital circuit (Combinational or Sequential). (A, B)
4. Have knowledge in analyzing and designing procedures of Combinational and Sequential circuits. (B, C)
5. Have knowledge in designing and analyzing circuits with Flip-Flops, Counters and Registers. (B, C)
6. Work effectively with others. (D)
7. Use simulation software, for testing the designed circuit. (C, D)

Assessment of Learning Outcomes

Learning outcomes (1), (2), and (3) are assessed by examinations, tutorial and in the laboratory. Learning outcomes (4), (5), and (6) is assessed by course work/workshops. Learning outcomes (7) is assessed in the laboratory.

Contribution to Programme Learning Outcomes:

A1, A3, A5, B1, B3, C6, D3, D6

Synopsis: Introduction to Digital logic Design; Binary Systems and Codes: Binary Numbers, Octal and Hexadecimal Numbers, Number Base Conversions, Arithmetic Operation with different Bases, Complements, Signed Binary Numbers, Binary Codes: BCD, Gray, ASCII and EBCDIC; Binary Logic and Logic Gates: AND, OR and NOT; Boolean Algebra and Logic Gates: Basic Definition, Basic Theorems, Boolean Functions; Standard Forms: Minterm and Maxterm, Simplification of Boolean Functions using SOP and POS; Logic Operations: NAND, NOR, Exclusive-OR and Equivalence, Integrated Circuits; Gate-Level Minimization: The Map Method, Two- and Three-Variable Map, Four-Variable Map, Product of Sums Simplification, Don't-Care Conditions, NAND and NOR Implementation, The Tabulation Method, Simplification of Boolean Functions using Tabulation Method; Analysis and Synthesis of Combinational Circuits: Combinational Circuits, Analysis and Design Procedure, Binary Adders-Subtractor, Decoders and Multiplexers; Analysis and Synthesis of Sequential Circuits: Sequential Circuits, Latches, Flip-Flops: RS, D, JK and T, Analysis of Clocked Sequential Circuits, Design Procedure; Registers and Counters: Registers, Shift Registers, Synchronous Counters, Ripple Counters; Sequential Circuits with Programmable Logic Devices: Introduction, Random-Access Memory, Memory Decoding, Read-Only Memory, Programmable Logic Array.

Modes of Assessment:
Two 1-hour midterm exams (15% each); coursework (10%); Lab work (10%); Final (unseen) exam (50%)

Textbook and supporting material:
1- Mortis Mano, Digital Design, Prentice-Hall, 2002
2- Morris Mano, Charles R. Kime, Logic and computer design fundamentals, Pearson Prentice Hall, 2004
3- Basavaraj,B., Digital fundamentals, New Delhi: Vikas Publishing House, 1999.
4- Kandel Langholz, Digital Logic Design, Prentice Hall, 1988.
5- Rafiquzzaman & Chandra, Modern Computer Architecture, West Pub. Comp., 1988.

731332, Systems Analysis and Design

Providing Department: Management Information Systems, Faculty of IT
Module Coordinator(s):
Year: 3
Credit: 3 credit hours
Prerequisite: 760261 or Concurrently
Aims:

This module introduces the students to the concepts and skill of system analysis and design. It includes expanded coverage of data flow diagrams, data dictionary, and process specifications, as it introduces examples of new software used by analysts, designers to manage projects, analyze and document systems, design new systems and implement their plans. It introduces also a recent coverage of UML, wireless technologies and ERP; web based systems for e-commerce and expanded coverage on RAD and GUI design.

Teaching Methods: **32 hours Lectures (2 per week) + 8 hours Tutorials (1 per fortnight) + 8 hours Seminars (in the last 3 weeks)**

Learning Outcomes:

At the end of this module, student will be able to:
1- Understand the principles and tools of systems analysis and design (A).
2- Solve a wide range of problems related to the analysis, design and construction of information systems (A, B, C).
3- Understand the application of computing in different context (A).
4- Understand the professional and ethical responsibilities of practicing the computer professional including understanding the need for quality. (A)
5- Plan and undertake a major individual project, prepare and deliver coherent and structured verbal and written technical reports (B, C, D).
6- Analysis and Design of systems of small sizes. (B, C)

Assessment of Learning Outcomes

Learning outcomes (1) – (4) are assessed by examinations, tutorial and in the assignments. Learning outcomes (5) and (6) are assessed by seminars and projects

Contribution to Programme Learning Outcomes:
A2, A3, A4, A5, B1, B3, C1, C2, C5, D2, D4, D5.

Synopsis: Systems Analysis Fundamentals: Introducing Systems Analysis and Design Concepts, roles of systems analysts, system development life cycle, using CASE Tools, depicting system graphically, determine feasibility, activity planning and control; Information requirements 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 essential of design: Designing output, designing the files or database, designing the user interface, designing data entry forms, documenting the design phase; Software engineering and implementation: Quality assurance through software engineering, implementing the information system, Object oriented analysis and design; Different software tools will be used in this course.

Modes of Assessment:
Two 1-hour midterm exams (15% each); Assignments (15%); Tutorial Contribution (5%); Final Examination: written (unseen) exam (40%) + final project (10%)

Textbook and Supporting Material:
1. Kenneth E. Kendall and Julie E. Kendall, Systems Analysis and Design 5th Edition, Prentice Hall PTR, 2001
2- Silver and Silver, System Analysis and Design, Addison Wesley, Last Edition

750232, Computer Architecture

Providing Department: Computer Science, Faculty of IT
***Module Coordinator*:**
***Year:* 2**
Credit: 3 credit hours
Prerequisite: 731150+751231

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.

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

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)

Assessment of Learning Outcomes:
Learning outcomes (1-7) are assessed by examinations, tutorials. Learning outcome (8) is assessed by assignments and seminars.

Contribution to Programme Learning Outcomes
A2, A3, A4, B2

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.

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

Textbook 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.

750322, Design and Analysis of Algorithms

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 3
Credit: 3 credit hours
Prerequisite: 210103+721221

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.

Teaching Methods: 38 hours Lectures (2 per week (including two 1-hour midterm exams)) + 10 hours Tutorials (average 1 hour per week)

Learning Outcomes:

When completing this module, you should be able to:
1. understand basic ideas about algorithms (A)
2. develop efficient algorithms for simple computational tasks (B)
3. reason about the correctness of algorithms (B)
4. understand the concepts of time and space complexity, worst case, average case and best case complexities and the big-O notation (A)
5. compute complexity measures of algorithms, including recursive algorithms using recurrence relations (B)
6. understand the range of behaviours of algorithms and the notion of tractable and intractable problems (A, B)
7. know and understand a wide range of searching and sorting algorithms (A, B)

Assessment of Learning Outcomes:
All learning outcomes are assessed by examinations and tutorials. Learning outcomes (4), (5), and (6) are assessed by examinations and coursework.

Contribution to Programme Learning Outcomes:

A1, A2, B1, B2, B3

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

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

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

750444, Information and Computer Networks Security

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 4
Credit: 3 credit hours
Prerequisite: 751443

Aims:
Upon successful completion of the course, the student will be knowledgeable of network security principles and implementation, including the technologies used and principles involved in creating a secure computer networking environment; authentication, types of attacks and malicious code that may be used against a network; threats and countermeasures for e-mail, Web applications, remote access, and file and print services; security topologies; technologies and concepts used for providing secure communications channels, secure internetworking devices, and network medium; intrusion detection systems, firewalls, and physical security concepts; security policies, disaster recovery, and computer forensics; and daily tasks involved with managing and troubleshooting security technologies.

Teaching Methods: 40 hours Lectures (2-3 hours per week) + 4 hours Tutorials (1 per 3 weeks) + 4 hours Lab (1 per 3 weeks)

Learning Outcomes:
Students completing this module should be able to*:*

1. To provide the student with basic knowledge of general security concepts, including authentication methods, common network attacks and how to safeguard against them. (A)
2. To provide the student with basic knowledge of communication security, including remote access, e-mail, the Web, directory and file transfer, and wireless data. (A)
3. To provide the student with basic knowledge of infrastructure security, including various network devices and media, and the proper use of perimeter topologies such as DMZs, Extranets, and Intranets to establish network security. (B)
4. To provide the student with basic knowledge of cryptography basics, including the differences between asymmetric and symmetric algorithms, and the different types of PKI certificates and their usage. (B,C)
5. To provide the student with basic knowledge of operational/organizational security, including its relationship to physical security, disaster recovery, and business continuity, as well as computer forensics and how it relates to further avenues. (C,D)

Assessments of Learning Outcomes:

Learning outcomes (1) and (2) are assessed by examinations. Learning outcomes (3) and (4) are assessed by assignments and research.

Contribution to Programme Learning Outcomes
A3, A5, B2, C2, C4, C5, D1, D4, D5.

Synopsis: Layered communication architecture: layers, services, protocols, layer entities, service access points, information Security Fundamentals, Attackers and Their Attacks, Security Basics, Security Baselines, Securing the Network Infrastructure, Web Security, Protecting Advanced, Communications, Scrambling Through Cryptography,

Using and Managing Keys, Operational Security, Policies and Procedures, Security Management, Advanced Security and Beyond

Modes of Assessment:
Two 1-hour midterm exams (15% each); Assignments (20%); Final Examination: 2-hours written exam (35%) + a research project (15%).

Textbooks and Supporting Material:
1-Security+ Guide to Network Security Fundamentals, Second Edition, by Mark Chiampa Course Technology, 2004

2- William Stallings, Wireless Communications & Networks, 2nd edition, Prentice-Hall Pearson, 2005

750425, Advanced Data Structure and Algorithms

Providing Department: Computer Science, Faculty of IT
***Module Coordinator(s)*:**
Year: 4
Credit: 3 credit hours
Prerequisite: 750332

Aims:
The course covers advance topics in Data Structures and Algorithms such as spatial data and its representations which is mainly based on the principle of recursive decomposition (Quadtree); Balanced Search trees (AVL trees, 2-3 trees, splay trees); Algorithms for generating combinatorial objects; Approximation Algorithms for NP-hard problems; Parallel Algorithms for machines that have more than one processor working on one problem at the same time; Correctness of Algorithms (loop invariants, proof by contradiction).

Teaching Methods: 31 hours Lectures (1-2 per week) + 7 hours Tutorials (1 each fortnight) + 10 hour Laboratory (1-2 per week)

***Learning Outcomes*:**
By the end of the module, the student should be able to:

1. Learn algorithms for ubiquitous graph problems, that can be solved very efficiently; Students should be able to solve open problems by modifying these algorithms (A).
2. Learn complex data structures used in these algorithms; Students should be able to modify and use these data structures for solutions of open problems when necessary (B).
3. Learn how to analyze behavior of relatively complex algorithms; Students should be able to evaluate running time of the algorithms that they have designed as solutions for open problems.(B,C)
4. Be able to implement these algorithms and data structures using STL.(B,C,D)

Assessment of Learning Outcomes:
Learning outcomes (1) - (4) are assessed by coursework and examinations

Contribution to Programme Learning Outcomes:
A2, B1, B3, C1, C5

Synopsis: Trees, Graph representations, Growth of functions Breadth first search, Flow Networks , Generic Ford-Falkerson algorithm, Edmonds-Karp modification, Bipartite matching, MST-Prim, Potential method. Fibonacci Heaps, Fibonacci heap, Implementing MST-Kruskal efficiently; Disjoint set lists and forests, SSP and its properties, Bellman-Ford algorithm, Difference constraints, Linear programming, including difference constraints

Modes of Assessment:
Two 1-hour midterm exams (20% each); Lab work (10%); Final (unseen) examination: written exam (40%) + Project (10%)

Textbooks and Supporting Material:
ntroduction to Algorithms. By T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, MIT Press, 2001, 0-07-013151- 1

750412, Advanced Programming

Providing Department: Computer Science, Faculty of IT
***Module Coordinator(s)*:**
Year: 4
Credit: 3 credit hours
Prerequisite: 750321

Aims:
This course is constructed of real-world programs -- programs that robustly interact with their users and with their computing environment. The course is roughly organized into two sections: Unix programming and Java programming. The first section covers Unix system calls, shells, and tools; the second section introduces and uses the distributed programming language Java. In the context of Java, we will study concurrency, graphics, user interfaces, and network programming. The course emphasizes large-scale programming and program correctness. Throughout the term, guests will augment the core material with lectures describing various software systems.

Teaching Methods: 32 hours Lectures (2 per week) + 16 hours Tutorial (1 per week) + 16 hours Laboratory (1 per week)

Learning Outcomes:
Students who successfully complete this course will be able to
1. Apply and develop object oriented code.
2. Develop software for a variety of architectures (e.g. Windows, Unix, and Linux).
3. Choose an appropriate computer language for a given project.
4. Demonstrate basic knowledge of parallel and network programming
5. Demonstrate basic knowledge of software engineering concepts.

Assessment of Learning Outcomes:
Learning outcomes (1), (3), (5), and (7) are assessed by assignments and Seminars. Learning outcomes (2) (4), (8), and (9) are assessed by examination and assignments.

Contribution to Programme Learning Outcomes:
A2, A3, B1, B2, B3, C1, C3, C5, D1, D3

Synopsis: processes, files, file systems, shell programming, shell graphics, java , java, top down , java, bottom up; unix timers; advanced java , java class libraries, concurrency, threads, synchronization , graphics, java graphics, user interfaces , java interaction , network programming , networked java , java security.

Modes of Assessment:
Two 1-hour midterm exams (15% each); Seminar preparation and presentation (10%); Assignments (10%); Final (unseen) Exam (40%) + Final Project (10%)

Textbooks and Supporting Materials:
1- JAVA for Scientists and Engineers," 2nd Edition, by S. Chapman, Prentice-Hall.
2- David j. Barnes And Michael Kolling, Objects First with Java: A Practical Introduction using BlueJ, Prentice Hall, Pearson Education, 2nd Edition, 2005

750411, Applications of WWW Programming

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year:4
Credit: 3 credit hours
Prerequisite: 731270

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

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)

Learning Outcomes:
Students completing this module 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)

Assessment of Learning Outcome:
Learning outcomes (1) and (2) are assessed by examinations and assignments. Learning outcome (3) is assessed by projects and seminars.

Contribution to Programme Learning Outcomes:
A3, A4, B2, B3, C3, C5, C6

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.

Modes of Assessment:
Two 1-hour midterm exams (15% each); Assignments (15%); Seminars (5%); Final Examination: 2-hours written exam (30%) + defended project (20%)

Textbooks 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

761443, **wireless and Mobile Computing**

Providing Department: Computer Information Systems, Faculty of IT
Module Coordinator(s):
Year: 4
Credit: 3 credit hours
Prerequisite: 761340

Aims:
To impart an understanding of fundamental concepts underlying current developments in mobile communication systems and wireless computer networks.

Teaching Methods: 32 hours Lectures (2 per week) + 8 hours Tutorials (1 per 2 weeks) + 8 hours Projects/Seminars (1 per 2 weeks)

Learning Outcomes:
At the end of the course, students will have acquired the following knowledge and skills.
1. Understanding of characteristics of radio propagation and interference in multipath propagation and channel model description (A)

2. Understanding of a range of digital transmission systems as used for applications in mobile telephony and wireless computer networks, pulse shaping and equalisation techniques (A)

3. Understanding of the issues and techniques used in the design of Medium Access Control protocols for wireless Networks (A)

4. Understanding of the systems, protocols and mechanisms to support mobility for mobile internet users (A)

5. The ability to investigate fundamental aspects of transmission and modulation by writing MATLAB programs. The experience of using an industrial standard network simulation package(A,B,C,D)

Assessment of Learning Outcomes:

Learning outcomes (1-6) are assessed by examinations, tutorials, seminars and projects.

Contribution to Programme Learning Outcomes:
A1 – A5, B1, B3, C3, C4, C5, D6

Synopsis: *Introduction to wireless networking, Advantages and disadvantages of wireless networking, Characteristics of radio propagation, Fading, Multipath propagation, Introduction to digital transmission, Definition of bit-rate and signalling rate, Introduction to synchronous transmission, The need for pulse shaping, synchronisation and line-coding, Calculation of bit-error probabilities when the channel is affected by the addition of Gaussian noise, Narrowband digital modulation, The need for modulation, Binary and multi-level (M-ary) amplitude-shift keying (ASK), frequency-shift keying (FSK) and phase-shift keying (PSK), Wideband modulation techniques to cope with intersymbol interference , Direct sequence spread spectrum Adaptive Equalization Orthogonal frequency division multiplex, Medium Access Control (MAC), MAC protocols for digital cellular systems such as GSM, MAC protocols for wireless LANs such as Hidden and exposed terminals, Collision Avoidance (RTS-CTS) protocols, Protocols supporting mobility, Mobile network layer protocols such as mobile-IP, Dynamic Host Configuration Protocol (DHCP), Mobile transport layer protocols such as mobile-TCP, indirect-TCP, Wireless Application Protocol (WAP).*

Modes of Assessment:
Two 1-hour midterm exams (15% each); Assignments (10%); Seminars (10%); Final Examination: 2-hours written exam (35%) + Project (15%)

Textbooks and Supporting Material:
1. Schiller, Mobile communications, ISBN: 0-321-12381-6, Addison-Wesley, 2003
2. T.S. Rappaport, Wireless communications; Principle and Practice, ISBN: 0-13-375536-3
3. A S. Tanenbaum, Computer Networks (Fourth Edition), Publisher: Prentice Hall PTR; ISBN: 0130661023; August, 2002.

761462 Information Retrieval.

Providing Department: Computer Information Systems, Faculty of IT
Module Coordinator(s):
Year: 4
Credit: 3 credit hours
Prerequisite: 760261

Aims:
Information Retrieval (IR) is a really HOT subject these days. All thanks to the World Wide Web, Web Search, and our friends at Google, Yahoo!, MSN, and all the other search engines that have come and gone!!! But, there's a lot that happens between the typing of 2-3 keywords in a small box at the User Interface, and receiving the results. In the next ten weeks we'll look at issues surrounding information retrieval systems. We will examine information system design and evaluation issues, and look under the hood of the search engines to pick at what's going on and why.

Teaching Methods: 36 hours Lectures (2-3 hours per week) + 8 hours Seminars (1 per 2 weeks) + 4 hours Laboratory (1 per 3 week)

Learning Outcomes:
A student completing this module unit should be able to:
1- become familiar with basic issues and current practice in IR (A)
2- familiarize student's selves with IR tools (B)
3- review important research in IR. (C,D)

Assessment of Learning Outcomes

Learning outcomes (1-5) are assessed by examinations, tutorial and in the laboratory. Learning outcomes (6-8) and (10)are assessed by tutorials and in laboratory. Learning outcomes (9) and (11) are assessed by seminars and/or workshop

Contribution to Programme Learning Outcomes

A1, A2, A3, A4, A5, A6, C1, C2, C3, C4, D1, D2, D4, D5.

Synopsis: Introduction to IR. Overview of the components of an IRS, Queries, Documents, Indexing, Theories & Models in IR (Retrieval Techniques) for text, hypermedia, web, Exploring Google search, Queries and Information Needs, Document Analysis, Structure of documents, parsing, stemming, morphological analysis, tokenization, Retrieval Techniques, Exact Match vs. Partial Match Weighted Ranked Retrieval, Vector Space & Probabilistic retrieval models, Relevance Feedback, Retrieval Techniques, Retrieval Techniques, Exact Match vs. Partial Match Weighted Ranked Retrieval, Vector Space & Probabilistic retrieval models, Relevance Feedback, Query Processing for IR ,Query Formulation, Expansion, Refinement, Web search. Hypertext, Internet Search Engines, Link Analysis, Evaluation of IRS, Performance Measures. Relevance, User centered evaluation of IR systems, Evaluation of IRS, Evaluating Exploratory Search Systems, Web search Crawlers, Web graph. Log Analysis, evaluation revisited, Faceted Search.

*Mode of Assessment***:**

Two 1-hour midterm exams (15% each); Coursework (15%); Tutorial contribution (5%); Final Examination: written (unseen) exam (35%) + Lab Exam (15%)

Textbook and Supporting Material:

Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.

210104, Discrete Structures

Providing Department: Basic Sciences, Faculty of Science
Module Coordinator(s):
Year: 1
Credit: 3 credit hours
Pre-requisites: none
Pre-requisite for: 751323, 760261

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.

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

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.

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

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.

……..

761211, Windows Programming

Providing Department: Applied Computer Science, Faculty of IT
***Module Coordinator(s)*:**
Year: 1
Credit: 3 credit hours
Prerequisite: 710104

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.

Teaching Methods: 32 hours Lectures (2 per week) + 12 hours Tutorials (on average 1 per week) + 16 hours Laboratory (1 per week) + 4 hours Seminar

Learning Outcomes:

On completing this module you should:
1. Have a clear understanding of what comprises a correct program in C# through .Net frame components (A)
2. Have a clear understanding of the object-oriented terminology used to describe features of C# and VC# project with their visual components. (A, C)
3. Have an informal understanding of the operational semantics of object-oriented programs in terms of creation of objects and messages passing between difference interfaces. (A)
4. Be able to design, code, and test C# project, which meet requirements expressed in English. (B, C, D)

5. Be able to understand the documentation for, and make use of, the MSDN library. (A, C)
6. Have a good understanding of the different focus at various stages of the development process. (A, C, D)
7. Have knowledge of design GUI with visual components guidelines. (A, B)
8. Be able to apply the guidelines in learning outcome (7) to a real design problem and justify how they have been used. (A, B)
9. Be able to write a project in C# and VC#, which implements the design in learning outcome (8). (C).
10. Be able to work effectively alone or as a member of a small group working on some programming tasks. (C, D)

Assessment of Learning Outcomes:
Learning outcomes (1), (6), and (8) are assessed by examination and laboratory; learning outcomes (2), (3), and (7) are assessed by tutorial and examination; learning outcomes (4), (5), (9) and (10) are assessed in the laboratory.

Contribution to Programme Learning Outcomes:

A2, A3, A4, B3, C5, C6, D1, D2, D4, D5

Synopsis: **Introducing the Microsoft .NET Platform:** .NET Platform, .NET and Windows DNA, .NET Architecture Hierarchy, .NET Platform features, Multilanguage Development, Platform and Processor Independence, .NET Components, Common Type System CTS, Common Language Specification CLS , .NET Base Class Library (BCL); **Visual Studio.NET IDE:** Visual Studio.NET, Components of VS.NET, Design Window, Code Window, Server Explorer, Toolbox, Docking Windows, Properties Explorer, Solution Explorer, Object Browser, Dynamic Help, Task List Explorer, Features of VS.NET, XML Editor, Creating a Project, Add Reference, Build the Project, Debugging a Project; **Introducing C# Programming:** Data Types, Value Types, Reference Types, Control Structures (if, if-else, switch, for, while, do while, break, continue, return, goto), Understanding Properties and Indexers Accessing Lists (Array) with Indexers, Events, Exception Handling, Using OOP (Object, Class, Constructor/destructor, Inheritance, Polymorphism, Encapsulation); **Windows Forms:** Windows Forms, Adding Controls, Adding an Event Handler, Adding Controls at Runtime, Attaching an Event Handler at Runtime, Writing a Simple Text Editor, Creating a Menu, Adding a New Form, Creating a Multiple Document Interface, Creating a Dialog Form, Using Form Inheritance, Adding a *TabControl,,* Anchoring Controls, Changing the Startup Form, Connecting the Dialog, Using the ListView and TreeView, Controls, Building an *ImageList,* Adding a *ListView,* Using the Details View, Attaching a Context Menu, Adding a *TreeView,* Implementing Drag and Drop, Creating Controls, Creating a User Control, Adding a Property, Adding Functionality, Writing a Custom Control, Testing the Control, Enhancing the Control, Sub classing Controls; **Graphics and Multimedia:** Graphics Contexts and Graphics Objects, Color Control, Font Control, Drawing Lines, Rectangles and Ovals, Drawing Arcs, Drawing Polygons and Polylines, Advanced Graphics Capabilities, Introduction to Multimedia, Loading Displaying and Scaling Images, Animating a Series of Images, Windows Media Player, Microsoft Agent; **ADO.NET:** ADO.NET Architecture, Understanding the *Connection*Object, Building the *Connection* String, Understanding the *Command*Object, Understanding *DataReaders,* Understanding *DataSets* and *DataAdapters,* DataTable, *DataColumn, DataRow,* Differences between *DataReader* Model and *DataSet* Model, Understanding the *DataView*Object ,Working with System.Data.OleDb, Using *DataReaders,* Using *DataSets,* Working with SQL.NET, Using Stored Procedures, Working with Odbc.NET, Using DSN Connection; **Multithreading:** Thread States: Life Cycle of a Thread, Thread Priorities and Thread Scheduling, Thread Synchronization and Class Monitor, Producer/Consumer Relationship without Thread, Synchronization, Producer/Consumer Relationship with Thread Synchronization, Producer/Consumer Relationship: Circular Buffer; **Networking:** Introduction, Establishing a Simple Server (Using Stream Sockets), Establishing a Simple Client (Using Stream Sockets), Client/Server Interaction with Stream-Socket Connections, Connectionless Client/Server Interaction with Datagrams, one Server multi-Clients system; **ASP.NET:** Introducing the ASP.NET Architecture, ASP.NET Server Controls, Working with User, Controls, Custom Controls, Understanding the Web.config File, Using the Global.asax Page,

Modes of Assessment: Two 1-hour midterm exams (15% each); Assignment 15%; Tutorial Contribution (5%); 2-hours Final Exam (50%: 35% Written Exam + 15% Practical Exam)

Textbooks and reference books:

1- H. M. Deitel & J. Deitel, " C# How to Program", Prentice Hall, 2001
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.

……..

761272, Multimedia Systems

Providing Department: Applied Computer Science, Faculty of IT
***Module Coordinator(s)*:**
Year: 2
Credit: 3 credit hours
Prerequisite: 710104

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.

Teaching Methods: 40 hours Lectures (2-3 per week) + 8 hours Tutorials (1 per 2 weeks) + 16 hours Laboratory (1 per week)

Learning Outcomes:
On completing this module you will:
1. Understand basic multimedia concepts. (A)
2. Acquire basic knowledge on Multimedia devices. (A, D)
3. Understand current trends in multimedia by experiencing a variety of applications and development packages. (A, C)
4. Understand the preproduction process including content acquisition and development, process flow, team management and integration, and legal issues surrounding multimedia (A)
5. Demonstrate technical knowledge and limited proficiency in designing production elements in each of the multimedia disciplines. (A, C)
6. Create a multimedia project for the desktop or Internet. (B)

Assessment of Learning Outcomes:
Learning outcomes (1 - 4) are assessed by examination, tutorials and laboratory; learning outcomes (5) and (6) are assessed by laboratory and examination.

Contribution to Programme Learning Outcomes:

A2, A4, A5, B3, C4, C6, D2

Synopsis*:* Introduction to Multimedia: definition, classification (discrete, continuous, passive, interactive), properties; Medium perception, representation, presentation, storage, and transmissionl; Lab: An overview of macromedia flash; MM hardware, application areas, stages of MM project, design issues (speed, simplicity, clarity, consistency, ease of use, and navigation); Lab: Flash drawing tools; Media and data stream, transmission modes, authoring tools (types, features, card/page-based, time-based, and icon-based); Lab: Flash panels; Text: text importance, encoding, fonts (type, size, style, leading, and kerning), text in MM (font design, menus, buttons, fields, portrait, landscape), editing design tools, hypertext vs. hypermedia; Lab: Animation and motion tween; Sound terminology (acoustic, electromagnetic wave, cycle, frequency, amplitude, decibel); Digital audio (sampling, quantization, file size, size vs. quality, formats); Lab: Guide layer and symbols in flash; MIDI files (creation, size, advantages, disadvantages). MIDI vs. digital audio; Speech: generation (TTS), recognition (STT), applications, difficulties, program learning); Lab: Shape tweening; Sound summary; Lab: Demos on MIDI maker, TTS and STT; Digital image (bitmap, vector graphic); Bitmap (pixels, color encoding, palette, and models, resolution); Lab: Mask layers and text animation in flash; Image scanning, capturing, editing, morphing, dithering, file size, format (BMP, GIF, PNG, JPEG, …etc); Vector graphics (types, properties, drawing, advantages, disadvantages, file size); Lab: Design and create buttons in Flash; Bitmap image vs. vector graphic; Image processing and programming skills; Lab: Image processing (write code); Animation : transition, cel animation (key frames, tweening, layers, morphing, formats); Lab: Image processing (write code); Video: concepts, standards, capturing, analog vs. digital, TV vs, computer video, compression and streaming; Lab: Flash action scripts; Encoding requirements (entropy, source, and adaptive), fixed length vs. variable length encoding, compression (HW vs. SW, lossy vs. lossless); Lab: Flash action scripts; Compression (symmetric vs. asymmetric, dialogue mode vs. retrieval mode, RLE, Huffman); Lab: Presentations of Flash and programming assignments; Compression techniques (JPEG and MPEG); Lab: Presentations of Flash and programming assignments;

Modes of Assessment: Two 1-hour midterm exams (15% each); Assignments (10%); Lab work (10%) + 2-hours Final Exam (50%)

Textbooks and reference books:

1. Vaughan Tay, Multimedia: Making it work, Berkeley Osborne McGraw-Hill, 6th Edition 2004.
2. Ralf Steinmetz & Klara Nahrstedt, Multimedia fundamentals Volume 1: Media coding and content processing, Prentice-Hall, 2002.
3. Stephen McLoughlin, Multimedia: Concepts and Practice, Prentice hall, 2001.
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:

Flash, Macromedia, Photoshop

……

721120, Object-Oriented Paradigms

Providing Department: Software Engineering, Faculty of IT
***Module Coordinator(s)*:**
Year: 1
Credit: 3 credit hours
Prerequisite: 710104

Aims:
This module introduces the concepts of object-oriented programming for students after having a background in the procedural paradigm. 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. The module emphasizes modern software engineering principles and developing fundamental programming skills in the context of a language that supports the object-oriented paradigm (Java for instance).

Teaching Methods: 32 hours Lectures (2 per week) + 16 hours Tutorial (1 per week) + 16 hours Laboratory (1 per week)

Learning Outcomes:
A student completing this module should:
1. Acquire a full Object Oriented Thinking (A)
2. Have a clear understanding of the object-oriented concepts such as objects, classes, inheritance, and polymorphism. (A)
3. Have an informal understanding of the operational semantics of object-oriented programs in terms of creation of objects and messages passing between them. (A)
4. Be able to design small object oriented programs which meet requirements expressed in English, with a strong software engineering foundation (B)
5. Have knowledge of Object Oriented Design guidelines. (A, B)
6. Be able to code small software systems in Java language. (C).
7. Be able to maintain large, high-quality software systems (C)

Assessment of Learning Outcomes:
Learning outcomes (1), (6), and (7) are assessed by examination and laboratory. Learning outcomes (2), (3), and (7) are assessed by tutorial and examination. Learning outcomes (4) and (5) are assessed in the laboratory.

Contribution to Programme Learning Outcomes:

A2, A3, B1, B2, C5, D2, D4

Synopsis: Introduction to Object Oriented Thinking: Object Modeling; Objects and Classes; Understanding Class Definition; Object Interaction (1): Overloading; Object Interaction (2): Composition; Grouping Objects; Using Library Classes; More Sophisticated Behavior: Information Hiding; Inheritance (1): Reuse, Inheritance (2): Sub-typing; Inheritance (3): Polymorphism, Overriding; Abstract Classes, Abstract Methods, Interfaces, Multiple inheritance; Exception Handling; Designing Applications

Modes of Assessment:
Two 1-hour midterm exams (15% each); Lab work (15%); Tutorial contribution (5%); Final Exam: written (unseen) Exam (40%) and lab Exam (10%)

Textbooks and Supporting Material:
1- David j. Barnes And Michael Kolling, Objects First with Java: A Practical Introduction using BlueJ, Prentice Hall, Pearson Education, 2nd Edition, 2005

……

721221, Object-Oriented Data Structures

Providing Department: Software Engineering, Faculty of IT
***Module Coordinator(s)*:**
***Year:* 2**
Credit: 3 credit hours
Prerequisite: 721112

Aims:
This is a **programming-intensive** module where students learn the fundamentals of designing data structures for use in complex programs. Data structures course is an essential area of study for computer scientists and for anyone who will ever undertake any serious programming task. This course 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 course 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.

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

Learning Outcomes:
On successful completion of this module, student will:
1. build on understanding of basic ideas about data structures given in the prerequisite module (A)
2. understand the basic concepts of time and space complexity (A)
3. be able to manipulate recursive algorithms (B)
4. be able to develop efficient algorithms for manipulating data structures (B)
5. know a range of algorithm structures and how to implement them (A, B, C)
6. know and understand a wide range of searching and sorting algorithms (A, B)
7. understand how the Abstract Data Type (ADT) is used (A)
8. understand several representations of trees, and their applications (A, C)
9. understand several representations of graphs, and their applications, together with a selection of important algorithms on graphs (A, C)
10. be able to construct and use the data structures mentioned above. (A, B, C)

Assessment of Learning Outcomes:
Outcomes 1 to 10 are assessed by coursework and examinations

Contribution to Programme Learning Outcomes:
A1, B1, B2, C5, D6

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

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

Textbooks and Supporting Material:

1- Nell Dale, Daniel T. Joyce and Chip Weems, Object-Oriented Data Structures using Java, Jones and Bartlett Publishers, 2001
2- Goodrich and Tamassia, Data Structures and Algorithms in Java, 2nd edition, John Wiley and Sons, 2000, ISBN 0471383678.
3- Arnold, Gosling, and Holmes, The Java Programming Language, 3rd edition, Addison-Wesley, 2000, ISBN 0201704331.
4- Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, Addison-Wesley, 1999

……

731150, Introduction to Information Systems
// Introduction to Systems Information Technology//

Providing Department: Management Information Systems, Faculty of IT
***Module Coordinator(s)*:**
Year: 1
***Credit*:** 3 credit hours
Prerequisite: 750112

Aims: This module aims to provide students with some concepts of information systems and some applications in business and management systems. This is a major introductory course presents problems in business environment and solutions with computer-based tools. It focuses on systems and information systems concepts and techniques. Students will learn the most effective ways to use information systems. Case studies are examined to highlight new technology and applications like multimedia.

Teaching Methods: 20 hours Lectures (1-2 hours per week) + 25 hours Class workshop and labs/E-Learning (1-2 per week) + 3 hours Workshops

Learning Outcomes:
On completing this module you should be able to:
1. Understand the relevant concepts from the General Systems Theory.
2. Implement the concepts of dynamic systems into the realm of Information Systems.
3. Understand the meaning of "Information System" distinguished from "Working Systems.
4. Discuss various types of information systems acting in business organizations.
5. Be aware of the techniques and technologies used for managing information in working organizations.
6. Introduce security and ethical issues related to information systems.
7. Understand the main issues and trends in computer-based Information Systems.

Assessment of Learning Outcomes:
Outcomes (1), (3), and (7) are assessed by examinations. Outcomes (2), (5), and (6) are assessed by workshops.

Contribution to Programme Learning Outcomes:
A2, A4, A5, B1, B4, C4, D1, D2.

Synopsis: Introducing General Systems Theory, Modelling, Dynamic Systems, Systems Theory in Business, Information Theory, Information Systems (ISs), Integrated ISs, Strategic IS Planning, Self reading(fundamental type of M.I.S, References (5) page 29-35, Alternative ISs Development Approaches and tools, Electronic business systems, Electronic commerce systems, ISs Evaluation, ISs Security and Control, Implications of Information Age, Moral and ethics in ISs, Information Technology: Computer hardware, Computer software: Application software; Information Technology: Computer software: System software, Telecommunications and Networks

Modes of Assessment: Two midterm exams (15% each); Homework (10%); Workshop Contribution (10%); 2-hours Final Exam (50%).

Textbooks and reference books:
1- Ralph M. Stair, George W. Reynolds, Fundamentals of Information Systems. Course Technology; 4th edition, 2007

2- Gerald M. Weinberg, an Introduction to General Systems Thinking, Silver Anniversary Edition, 2001.
3- The Analysis, Design, and Implementation of Information Systems, Henry C. Lucas, Jr, 4th ed. McGraw-Hill, 1992.
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- David Kroenke, Management Information System, 1999.

……

750333, Principles of Operating Systems

Providing Department: Applied Computer Science, Faculty of IT
Module Coordinator:
Year: 2
Credit: 3 credit hours
Prerequisite: 711231
***Prerequisite for*:** 750331, 750341

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.

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

Learning Outcomes:
On completing the 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)

Assessment of Learning Outcomes:
Learning outcomes (1) and (2) are assessed by examination. Learning outcome (3) is assessed via course project. Learning outcome (4) is not formally assessed.

Contribution to Programme Learning Outcomes
A3, B3, C5.

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

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

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, 2nd Edition, (ISBN 0-201-177-676), Addison Wesley, 1998.
3-A. S. Tanenbaum, Modern Operating Systems, Prentice Hall, 1992

……..

721210, Introduction to Software Engineering

Providing Department: Software Engineering, Faculty of IT
Module Coordinator:
Year: 2
Credit: 3 credit hours
Prerequisite: 710104
***Prerequisite for*:** 750361, 750398

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. This module gives an introduction to methods for analysis, design, testing, and implementation of medium size software systems. Simple and realistic case studies will be used along all the software process steps.

Teaching Methods: 38 hours Lectures (2-3 per week) + 10 hours Tutorial

Learning Outcomes:

A student completing this module should be able to:
1- 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. (A)
2- Design software systems of small size through academic and realistic case studies (tutorials). (B, C, D)

Contribution to Programme Learning Outcomes:

A2, B2, C5, D4

Synopsis: Basic Concepts: Software product, Software crisis, software engineering, software process, software process model, methodologies, methods, tools, artefacts; **Software Process (I):** process models, iterative process; **Software Process (II):** software process activities (specification, design and implementation, validation/verification, evolution); **Software Requirement Engineering (I):** Functional/Non Functional requirements, user requirements, system requirement, requirement document; **Software Requirement Engineering (II**): Software requirement, elicitation and analysis, basics on Use cases, UML notation; **Software Prototyping; System Models (I):**Context models, Behavioural models**; System Models (II):** Data Models, Objects Models**; Architectural Design:** system structuring, control models, modular decomposition; **Object Oriented Design, UML notation; User interface design:** user interface design principles, user interaction, information presentation**; Verification and Validation:** planning, software inspections, automated static analysis; **Software Testing :** defect testing, integration testing; **Software Change:** program evolution dynamics, software maintenance; **Software Cost estimation**

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

Textbook and supporting material:
1- Ian Sommerville, Software Engineering 7/e, Addison Wesley, 2004
2- R. S. Pressman Software Engineering: A Practitioner's Approach, 5th Edition, McGraw Hill; 2001
Website(s): www.software-engin.com

……..

721240, Professional Issues in Computing and Information Technology // computing ethics// أخلاقيات الحوسبة

Providing Department: Software Engineering, Faculty of IT
***Module Coordinator(s)*:**
Year: 2
Credit: 3 credit hours

Prerequisite: None

Aims:
This module aims to give students an informed awareness of the principal issues of professional ethics and responsibility (ergonomics and ethics) in the analysis, design, implementation and use of computers, information systems and Information Technology (IT) products. This will help students in recognition of ethical problems when they occur. Also it will enable students to deal effectively with ethical, social and professional issues now and in their future careers.

Teaching Methods: 36 hours Lectures (2-3 per week) + 9 hours Projects (class work) (average 1 per week) + 3 hours Seminars (1 per month)

Learning Outcomes:
On completing this module, students will:
1. Understand the basic concepts of ethics, moral, law, ergonomics and profession.
2. Be aware of the requirements for accreditation in respect of Professional Issues.
3. Have a basic knowledge of Intellectual Property Rights (IPR) in relation to Copyright and Patents.
4. Be aware of some of the potential problems of managing large IT projects in accordance with professional and ethical issues.
5. Be aware of the requirements for professionalism in respect of the work of the professional societies and their codes of conduct and practice.
6. Have acquired basic knowledge of the Data Protection Act and its implications.
7. Be able to assess and evaluate the legal aspect of workplace practices.
8. Be able to asses and evaluate the impacts of IT technology on society and culture.
9. Be aware of Jordanian Professional Issues.

Assessment of Learning Outcomes:
Learning outcomes are assessed through examination and individual and group case studies, which require demonstration of the use of a combination of the learning outcomes to be employed in producing the essays and presentations.

Contribution to Programme Learning Outcomes:
A2, A5, B1, C1, C3, C4, C6, D3, D4.

Synopsis*:* Introduction to Ethics; Professional and Professionalism; Code of Ethics and Social Issues; Computer/IT professionals; Computer Security; Privacy and Internet Issues; Information Systems and Ethics; Associations of IT professionals; Ethics and the Internet; Ethical Challenges of e-Business; Ethical Challenges of e-Business; Continuous Professional Development; Intellectual Property Rights; Jordanian Codes for Intellectual Property Rights; Seminars and Project Discussion.

Modes of Assessment:
Two 1-hour midterm exams (15% each); Assessment by individual essay (10%); Workshop Assessment (10%); Final examination: written Exam (35%) + Final (case study) presentation (15%)

Textbooks and Supporting Material:
1. Deborah G. Johnson, Computer Ethics. 3ed Edition, Englewood Cliffs, N.J., Prentice Hall, 2001.
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. مجموعة تشريعات الملكية الفكرية الأردنية

Website(s):
ACM, IEEE and BCS Web Sites.
www.cyberethics.cbi.msstste.edu
www.aitp.org
www.acm.org
www.prenhall.com
www.jcs.rg.jo
……..

761340, Introduction to Computer Networking
أساسيات شبكات الحاسوب

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 3
Credit: 3 credit hours
Prerequisite: 711232
Prerequisite for: 750441, 750443, 750444

Aims:
This module is the first module of the curriculum related to the computer network field. Its aim is to provide students with a broad coverage of the basic computer networking concepts of the four layers of ISO, circuit switch, packet switch, etc.
The module, however, does not focus on a detailed study or cover the technologies. The concepts given in this module will be deeply handled in the next level module (750441).

Teaching Methods: 32 hours Lectures (2 per week (including two 1-hour midterm exams)) + 16 hours Tutorial (1 per week) + 16 hours Laboratory

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

1. Discuss important network standards in their historical context (A)
2. Describe the responsibilities of the first four layers of the ISO reference model. (A)
3. Discuss the differences between circuit switching and packet switching along with the advantages and disadvantages of each. (A, B)
4. Explain how a network can detect and correct transmission errors. (A, B)
5. Illustrate how a packet is routed over the Internet. (C)
6. Install a simple network with two clients and a single server using standard host-configuration software tools such as DHCP. (C, D)

Assessment of Learning Outcomes:
Learning outcomes (1) - (3) are assessed by examination and tutorials. Learning outcomes (4) – (6) are assessed by assignments and seminars.

Contribution to Programme Learning Outcomes
A3, A4, B2, C6, D2, D5, D6.

Synopsis: Introduction; Network Model; Data and Signal**;** Digital signal; Analog signal; Switching; Error Detection and Control; Error Detection and Control; Data Link Control; Multiple Access; Network Layer: Logical Addressing; Network Layer: Delivery, Forwarding, and Routing; Network Layer: Delivery, Forwarding, and Routing; Process-to process Delivery; Congestion Control and Quality of service.

Modes of Assessment:
Two 1-hour midterm exams (15% each); Course work (10%) + Seminar presentation and essays (10%); Final Examination: 2-hours written exam (40%) + Final Project (10%)

Textbooks and Supporting Material:
1- Behrouz A. Forouzan, Data Communications and Networking, McGraw Hill Higher Education, Fourth Edition, 2007
2- Andrew S. Tanenbaum, Computer Networks, Prentice Hall, Last Edition.

Website(s):
www.mhhe.com/forouzan
……..

750351, Fundamentals of Artificial Intelligence

Providing Department: Computer Science, Faculty of IT
Module Coordinators:
Year: 2

Credit: 3 credit hours
Prerequisite: 710104
***Prerequisite for*:** 750352, 750354, 750424

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.

Teaching Methods: 22 hours In-class Lectures (1-2 per week) + 16 hour E-learning Lectures (1 per week) + 8 hours Tutorials (1 per 2 weeks)

Learning Outcomes
A student completing this module should
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).

Assessment of Learning Outcomes
Learning outcomes (1) to (3) are assessed by examination. Learning outcomes (4, 5) are assessed by coursework of the unsupervised laboratory and by examination. Learning outcome (6) is assessed by coursework. Learning outcome (7) is assessed by on-line course.

Contribution to Programme Learning Outcomes:
A2, A3, A4, B1, B2, B3, B4, C4, C5, C6, D6

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).

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%).

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.
……..

750223, Theory of Computation

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 3
Credit: 3 credit hours
Prerequisite: 210104 + 721211
***Prerequisite for*:** 750421

Aims:
This module introduces the theory of computation through a set of abstract machines that serve as models for computation - finite automata, pushdown automata, and Turing machines - and examines the relationship between these automata and formal languages. Additional topics beyond the automata classes themselves include deterministic and nondeterministic machines, regular expressions, context-free grammars, undecidability, and the P = NP question.
Finite automata are a useful model for many important kinds of hardware and software. Here are the most important kinds: Software for designing and checking the behaviour of digital circuits; The "lexical analyzer" of a typical complier, that is, the compiler component that breaks the input text into logical units, such as identifiers, keywords, and punctuation; Software for scanning large bodies of text, such as collections of Web pages, to find occurrences of words, phrases, or other patterns; Software for verifying systems of all types that have a finite number of distinct states, such as communication protocols or protocols for secure exchange of information.

Teaching Methods: 38 hours Lectures (2-3 hours per week) + 10 hours Tutorials (average 1 per week)

Learning Outcomes:
A student completing this module should be able to:
1. Acquire a full understanding and mentality of Automata Theory as the basis of all computer science languages design (A)
2. Have a clear understanding of the Automata theory concepts such as RE's, DFA's, NFA's, Stack's, Turing machines, and Grammars (A, B).
3. Design FAs, NFAs, Grammars, languages modelling, small compilers basics (B).
4. Minimize FA's and Grammars of Context Free Languages (C).
5. Design sample automata (B)

Assessment of Learning Outcomes
Learning outcome (1) and (2) are assessed by tutorials and examinations. Learning outcomes (4) is assessed by tutorials, homework, and examinations. Learning outcomes (3) and (5) are assessed by tutorials.

Contribution to Programme Learning Outcomes:

A1, A2, B1, B2, C2, C5

Synopsis: Basic concepts and definitions; Set operations; partition of a set; Equivalence relations; Properties on relation on set; Proving Equivalences about Sets; Central concepts of Automata Theory; Regular Expressions; Operations on Regular expressions; Finite Automata and Regular Expressions; Recursive definitions; Conversion from FA and regular expressions; Kleen's Theory; Mealy Moore Machines; Conversion from Mealy to Moore and vice versa; Deterministic Finite Automata (DFA); Equivalence Classes; Minimization of DFA; Non-Deterministic Finite Automata (NDFA); Equivalence of Deterministic and Non- Deterministic Finite Automata; Finite Automata with Epsilon-Transition; Equivalence between DFA, NFA, NFA-Λ; Pumping Lemma for Regular Languages; Closure Properties of Regular Languages; Context Free languages; Context-Free Grammars; Regular Grammars; Parse Trees; Ambiguity in Grammars and Languages; Simplified Forms; Standard Forms; Chomsky Normal Forms; Greibach normal Forms; Pumping Lemma for Context-Free Languages; Closure Properties of Context-Free Languages; Minimization of CFGs; Pushdown Automata (PDA); Deterministic and Non-Deterministic (PDA); Formal definition of NPDA; Transition functions of NPDA; NPDA Execution; Accepting Strings with NPDA; Equivalence of PDAs and CFG; The Turing Machine; Programming Techniques for Turing Machines; Formal definition of TMs; TMs as acceptors; TMs as transducers; Recognizing Languages with TMs; Sorting with TMs; Programming in TMs; Multiple Tracks, Subroutines; Complexity issues and analysis.

Modes of Assessment:
Two 1-hour midterm exams (15% each); Assignments (15%); Tutorial contributions (5%); 2-hour Final exam (50%)

Textbooks and Supporting Material:

1. Daniel I. A. Cohen, "Introduction to computer theory", Second Edition, Prentice-Hall, 1997.
2. Papadimitriou, Elements of the Theory of Computation, Prentice-Hall, 1998
3. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, "Introduction to Automata Theory, Languages, and Computation", Second Edition, Prentice-Hall, 2001
4. Peter Dehning, Jack B. Dennis, "Machines, Languages and Computation", Second Edition, Prentice-Hall, 1978
5. Harry R. Lewis, Christos H. Papadimitriou, "Elements of the theory of computation", Second Edition, Prentice-Hall, 1998

Simulators:
In order to improve the pedagogy of this course, interactive animations of the various automata using available simulators are recommended.

……..

760261, Database Fundamentals

Providing Department: Computer Information Systems, Faculty of IT
Module Coordinator(s):
Year: 2
Credit: 3 credit hours
Prerequisite: 721211 or Concurrent + 210104

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.

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

Learning Outcomes:

When completing this module, a student should be able to:

1. Discuss/explain the importance of data, and the difference between file management and databases. (A)
2. Discuss/explain the design of database management system architectures and environments. (A)
3. Discuss/explain the principals of database design. (A)
4. Discuss, explain and apply conceptual design methodologies, in particular conceptual design using Extended Entity Relationship modelling. (A, B, C, D)
5. Discuss, explain and apply the relational model and mappings from conceptual designs, in particular normalizations. (A, B, C, D)
6. Discuss/explain physical and performance related design considerations. (A)
7. Discuss/explain transaction processing. (A)
8. Discuss, explain and apply SQL and the Oracle DBMS. (A, C, D)

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

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

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
……..

750321, Concepts of Programming Languages

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 3
Credit: 3 credit hours
Prerequisite: 721211
Prerequisite for: 750421

Aims:
This module aims to provide the student with a framework for thinking about programming languages. There are always new languages being devised, of which a very few actually become widely used; typically, these are specialized languages for particular applications (e.g., Java). As a computer scientist, the student must be able to learn new languages as necessary, and the background he/she gets from this module should make this easier. Finally, students will almost certainly have to choose which programming language to use for a particular project. A final goal of this module is to give students enough background in the study of programming languages that they can argue persuasively why a particular language is appropriate (or inappropriate) for a particular problem.

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

Learning Outcomes:
A student completing this module should be able to:
1. understand different programming paradigms. (A, D)
2. understand the syntax and semantic of programming languages. (A, B)
3. develop different projects using different programming languages. (B, C, D)
4. design a new programming language. (B)

Assessment of Learning Outcome:
Learning outcome (1) and (2) are assessed by homework, assignments, and examinations. Learning outcome (3) is assessed by project assignment and examinations. Learning outcome (4) could be assessed by project.

Contribution to Programme Learning Outcomes:
A1, B1, B3, C5, D4, D6

Synopsis: Introduction; A survey of Programming Paradigms; Imperative Programming: Names, Bindings, and Type Checking; Scopes; Data Types: Primitive Data Types, Character String Type, User-Defined Ordinal Types; Data Types : Array Types, Record Types, Union Types, Set Types, and Pointer Types; Statement-Level Control; Subprograms; Abstract Data Types; Support for Object-Oriented Programming; Functional Programming; Logic Programming; Scripting Languages

Modes of Assessment:
Two 1-hour midterm exams (15% each); Assignments (15%); Tutorial contribution (5%) + 2-hours Final Unseen Exam (40%) + Project (10%)

Textbooks and Supporting Material:
1- Robert W. Sebesta, Concept of Programming Languages, Addison Wesley, 5th Edition, 2002
2- Terrence W. Pratt, Programming Languages: Design and Implementation, Prentice-Hall, 2002
3- Ravi Sethi, Programming Languages Concepts and Constructions, Pearson Education, 1996
4- Allen B. Tucker, Programming Languages, McGraw Hill, 1988
5- C. Ghezzi and M. Jazayeri, Programming Language Concepts, John Wiley and Sons, 1982

……..

750334, Advanced Operating Systems

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 3
Credit: 3 credit hours
Prerequisite: 711232
Prerequisite for: 750432

Aims:
The aim of this module is to study, learn, and understand the main concepts of advanced operating systems (parallel processing systems, distributed systems, real time systems, network operating systems, and open source operating systems); Hardware and software features that support these systems.

Teaching Method: 45 hours Lectures (2-3 per week) + 3 hours Seminars

Learning Outcomes:
A student completing this module should be able to:
1. Outline the potential benefits of distributed systems. (A).
2. Apply standard design principles in the construction of these systems. (A, B).
3. Select appropriate approaches for building a range of distributed systems, including some that employ middleware. (B, C).
4. Summarize the major security issues associated with distributed systems along with the range of techniques available for increasing system security. (A)

Assessment of Learning Outcome:
Learning outcome (1) is assessed by tutorials and examinations.
Learning outcome (2) is assessed by tutorials, homework, seminars, and examinations.
Learning outcome (3) is assessed by tutorials.

Contribution to Programme Learning Outcomes:
A2, A3, A4, A5, B2, B3, C5.

Synopsis: Review of Operating Systems concepts; Hardware concepts of distributed systems
Software concepts and design issues; Communication in distributed systems; Threads and thread usage; Multithreading operating system; Client – server model; Implementation of Client-server model; Remote procedure call; Implementation of remote procedure call; Synchronization in distributed systems; Clock synchronization; Mutual exclusion; Election algorithms; Transaction and concurrent control; Deadlock in distributed systems; Processor Allocation; Real – time distributed systems; Distributed file systems

Modes of Assessment:
Two 1-hour midterm exams (15% each); Seminars (10%); Report (10%); 2-hours Final Unseen Exam (40%) + Final Project (10%)

Textbooks and Supporting Material:
1- Andrew S. Tanenbaum; Distributed operating systems; Prentice Hall; 1995
2- Jean Bacon; Concurrent Systems; Addison – Wesley; 1998
3- William Stallings; operating systems; Prentice Hall; 1995
…….

750398, Practical Training

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 3
Credit: 3 credit hours
Prerequisite: 721281; It is implanted according to the Faculty regulations

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.

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

Regulations for Training:
Students have to register on at most 15 credit hours in the semester in which they register on the practical training module.
2- Students must be full-time trainees for at least 2 days per week.
3- Students who take this module should arrange their timetable for other modules in a way that enables them to enrol in the pre-specified enterprise or company at least two days per week during the semester period.
4- The student has to get an official letter from the Faculty requesting a placement, and the Faculty provides a standard document that the placement provider could use to confirm that appropriate opportunities would be available to the student.
5- There is an academic supervisor for any trainee from the Department in addition to the supervisor from the placement provider.
6- Student should submit a report at the end of the training period.
7- At the end of the training period, the student and the placement provider fill some forms that will be used in assessing the student.
8- More information about training can be found in the Practical Training Handbook.

Learning Outcomes:
A student completing this module should:
1- be able to prepare and write any technical report. (C)
2- be prepared for any practical work (C)
3- be able to use IT skills (D)
4- learn how to work with and for others. (D)

Assessment of Learning Outcomes:

Learning outcome (1) is assessed by report evaluation, learning outcomes (3) – (4) are assessed by the observation of the training committee.

Contribution to Programme Learning Outcomes
C2, C3, D1, D2, D3

Synopsis: This module requires no syllabus, but any previously taught module will be valuable and can be applied in the practice.

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.

……..

750324, Compiler Construction

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 4
Credit: 3 credit hours
Prerequisite: 751323 + 750321

Aims:
This module aims to show how to apply the theory of language translation introduced in the prerequisite courses to build compilers and interpreters. It covers the building of translators both from scratch and using compiler generators. In the process, the module also identifies and explores the main issues of the design of translators. Topics include compiler design, lexical analysis, parsing, symbol tables, declaration and storage management, code generation, and optimization techniques. The construction of a compiler/interpreter for a small language is a necessary component of this module, so students can obtain the necessary skills.

Teaching Methods: 32 hours Lectures (2 per week) + 14 hours Tutorials (1 per week) (except the last two) + 2 hours Seminars

Learning Outcomes:
A student completing this module should be able to:
1. Understand the structure of compilers. (A, B)
2. Understand the basic techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, and intermediate code generation. (B, C)
3. Understand the basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines. (A, D)
4. Design and implement a compiler using a software engineering approach. (A, B)
5. Use generators (e.g. Lex and Yacc) (A)

Assessment of Learning Outcomes:
Learning outcomes (1), (2), and (3) are assessed by examinations, tutorials and coursework. Learning outcomes (4) and (5) are assessed by projects and seminars.

Contribution to Programme Learning Outcomes:
A2, A3, B2, B3, C1, C5, C6, D5

Synopsis: Introduction to Compilers: The role of language translation in the programming process; Comparison of interpreters and compilers, language translation phases, machine-dependent and machine-independent aspects of translation, language translation as a software engineering activity; Lexical Analysis: Application of regular expressions in lexical scanners, hand coded scanner vs. automatically generated scanners, formal definition of tokens, implementation of finite state automata; Syntax Analysis: Revision of formal definition of grammars, BNF and EBNF; bottom-up vs. top-down parsing, tabular vs. recursive-descent parsers, error handling; Parsers Implementation: automatic generation of tabular parsers, symbol table management, the use of tools in support of the translation process; Semantic Analysis: Data type as set of values with set of operations, data types, type- checking models, semantic models of user-defined types, parametric polymorphism, subtype polymorphism, type-checking algorithms; Intermediate Representation, code generation: Intermediate and object code, intermediate representations, implementation of code generators; Code generation: code generation by tree walking; context sensitive translation, register use; Code optimization: Machine-independent optimization; data-flow analysis; loop optimizations; machine-dependent optimization; Error Detection and Recovery; Error Repair, Compiler Implementation; Compiler design options and examples: C Compilers, C++, Java, and YACC Compilers

Modes of Assessment:
Two 1-hour midterm exams (15% each); Seminars (5%); Assignments (15%); 2-hours Final Exam (50%)

Textbooks and Supporting Material:
1- A. Aho, R. Sethi, J. D. Ullman, Compilers: Principles, Techniques, and Tools, Addison-Wesley, 1986
2- W. Appel, Modern Compiler Implementation in Java, Prentice Hall, 2002
3- D. Watt, Brown, Programming Language Processors in Java: Compilers and Interpreters, Prentice hall, 2000

……..

750441, Advanced Computer Networks

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 4
Credit: 3 credit hours
Prerequisite: 750341

Aims:
This module is the second level module of the curricula related to the computer network field. Its aim is to provide:
(a) an in depth coverage of some basic topics taught in the first level course (750341): Layered communication architecture, Routing algorithms, Congestion control algorithms,
(b) a broad coverage of some new advanced topics in the field of computer networks (wireless networks, mobile networks, VPN networks, Mobile IP, …)

Teaching Methods: 40 hours Lectures (2-3 hours per week) + 4 hours Tutorials (1 per 3 weeks) + 4 hours Lab (1 per 3 weeks)

Learning Outcomes:
Students completing this module should be able to*:*
1. Understand the main abstract concepts related to the layered communication architecture (A)
2. Analyze and implement some of the most advanced routing and congestion control algorithms. (B, C, D)
3. Evaluate the performances of computer networks (through mathematical modelling and simulation) (A, B, D)
4. Practice network simulators (B, C)
5. Understand basics and principles of new generation of computer networks (VPN, wireless networks, mobile networks, etc). (A)

Assessments of Learning Outcomes:

Learning outcomes (1) and (2) are assessed by examinations. Learning outcomes (3) and (4) are assessed by assignments and research.

Contribution to Programme Learning Outcomes
A3, A5, B2, C2, C4, C5, D1, D4, D5.

Synopsis: Layered communication architecture: layers, services, protocols, layer entities, service access points, protocol functions; Advanced Routing algorithms; Advanced Network Congestion Control algorithms; Quality of service; Real Time Transport Protocol; Internetworking; Performance Issues; Overview on VPN networks; Overview on Wireless Networks and Mobile Networks: LAN, PAN, Sensor Networks, Ad_hoc Networks; Mobile IP; Mobile TCP; IP Security

Modes of Assessment:
Two 1-hour midterm exams (15% each); Assignments (20%); Final Examination: 2-hours written exam (35%) + a research project (15%).

Textbooks and Supporting Material:
1- Andrew S. Tanenbaum, Computer Networks, (Fourth or Latest edition), Prentice Hall
2- William Stallings, Wireless Communications & Networks*,* 2nd edition, Prentice-Hall Pearson, 2005
3- Jochen Schiller, Mobile Communication, (Latest edition), Addison Wesley
4- G. Wright and W. Stevens, TCP/IP Illustrated, Volume 2, Addison-Wesley, 1996.
……..

750461, Advanced Databases

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 4
Credit: 3 credit hours
Prerequisite: 750361

Aims:

This module aims to give students information about system implementation techniques, introduction to DBMS implementation, data storage, representing data elements, database system architecture, the system catalog, query processing and optimization, transaction processing concepts, concurrency control techniques, database recovery techniques, database security and authorization, enhanced data models for advanced applications, temporal databases, deductive databases, database technology for decision support systems, distributed databases and client server architecture, advanced database concepts, and emerging technologies and applications.

Teaching Methods: 32 hours Lectures (2 per week) + 10 hours Laboratory (0-1 per week, on project assignment) + 6 hours Seminars (in last 3 weeks)

Learning Outcomes:
Students completing this module should be able to:

1- Apply normalization techniques. (A, B)
2- Understand how transactions are processed in a database. (A, B)
3- Discuss/explain the different techniques in Concurrency Control. (A, B)
4- Discuss/explain the concepts of Distributed Databases and Data Warehousing. (A)
5- Discuss/explain some database security issues.(A)
6- Tune and Optimize some Database Applications. (B, C, D)
7- Discuss/explain the concepts of Object-Oriented database. (A, D)

Assessment of Learning Outcome:
Learning outcomes (1) through (5) are assessed by examinations tutorials and assignments. Learning outcomes (6) and (7) are assessed by projects design and implementation.

Contribution to Programme Learning Outcomes:
A2, A3, A4, A5, B1, B2, B3, C2, C4, C6, D2, D3, D5

Synopsis*:* Introduction, Concepts and Definitions; Normalize Techniques; Data Warehousing and Data Mining; Transaction Processing; Concurrency Control; Distributed Databases; Database Security; Database Tuning and Query Optimization; Object-Oriented Database; Different tools will be used in this course including Oracle.

Modes of Assessment:
Two 1-hour midterm exams (15% each); Assignments (10%); Seminar Presentation (10%); Final Examination: 2-hour written (unseen) exam (40%) + Project (10%)

Textbooks and Supporting Material:
1- El Masri, Fundamentals of Database Systems, Fifth Edition, 2006
2- Date, Database Systems, Eighth Edition, 2004
3- Patrick Valduriez M. TamerOzsu, Principles of Distributed Database Systems, 2nd Edition, Prentice Hall, 1999.
……..

750499, Project

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 4
Credit: 3 credit hours
Prerequisite: 750398

General Descriptions:
The graduation 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.

- How to choose a project
- Organisation for projects
- Demonstrations
- Report Standards

- Staff List

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.

Teaching methods: Duration: 32 weeks (2 semesters) starts in first semester: Lectures: 6 or 7 in total, spread through the 2 semesters + Laboratories: none scheduled, 120 hours expected through semester

Learning Outcomes:
On completion of this module, a student should have
1. Used the project supervisor appropriately as project consultant or customer. (D)
2. Planned, executed and completed a significant design and, as appropriate, implementation within the time budget available. (B, C)
3. Given a demonstration showing practical competence and demonstrating the results of the project. (C).
4. Documented the project in a final report. (C)

Assessment of Learning Outcomes:
There is no examination.
Learning outcome (1) is assessed by the supervisor. Learning outcomes (3) is assessed by the project examination committee and by judging the demonstration. Learning outcome (4) is assessed by judging the report. Learning outcome (2) is assessed by all of these mechanisms.

Modes of Assessment:
Supervisor mark: 35% + Project Examination Committee mark: 65% (demonstration 20%, Report 25%, discussion 20%)

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 list and notes for guidance in carrying out a project are available in the Graduation Project Committee.

Project Classifications:
Parallel Processing
Artificial Intelligence
Systems Programming
Distributed Systems
Information Systems
Communications
Formal Specifications
Web Programming
General Applications
Simulation
Communications
Computer Aided Design

Microprocessor Applications and Peripherals
Graphical Systems and Applications

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 Graduation Project Committee (GPC).
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 GPC.
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".

The subdivision of marks within the project is:
The supervisor mark: 35%
The Project committee mark: 65% divided into
Demonstration, 20%
Discussion, 20%
Report, 25%

Note that the 25% is awarded for the report judged as a report only, independent of the quality of the work being described.

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. In general the supervisor of the project is not present.

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 25% 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 (5%): tidy layout, headings, references, diagrams.

……..

761373, E-Commerce Applications

Providing Department: Applied Computer Science, Faculty of IT
***Module Coordinator(s)*:**
Year: 3
Credit: 3 credit hours
Prerequisite: 712272

Aims:
The ramification of electronic commerce has already been felt in many functional areas: organizational design, marketing, finance and operations. The computerizing infrastructure that is necessary for implementing electronic commerce is becoming crucial in shaping the future of business. Relevant technology-management issues include Internet and intranet application design and deployment, business applications that leverage off the World Wide Web, firewalls and transactional security, intelligent agents, and electronic payment systems. To show how these concepts are implemented in practice, students are given hands-on experience of building an online store, through in-depth study of Active Server Pages (ASP) and database language SQL.

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

Learning Outcomes:
A student completing this module should:
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. Understand the process of maintaining security on the E-commerce site. (C)

3. Have knowledge in XML technology related to Business-to-Business E-commerce. (A)
4. Be able to build an online store. (B, D)

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 technology; Web design, JavaScript, & Internet; Information Server; Active Server Pages; Cookies and Applications; Online Database, building product catalogue, search product catalogue; Web Spider, search agent, and Internet Communication; Transaction Systems and Shopping Carts; XML, business-to-business E-commerce; Web Security.

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

Textbook and Supporting Material:
1. H. M. Deitel, P. J. Deitel and T. R. Nieto, E-Business and E-Commerce, Prentice hall, 2001
2. Turban, Rainer, and Potter, Introduction to E-Commerce, second edition, 2003
3. Walther & Levine, Sams, E-Commerce Programming with ASP, 2000
4. Russell Jones, Mastering Active Server Pages 3, Sybex inc., 2000
5. Darrel Inc, Developing Distributed and E-commerce Applications, Addison Wesley, 2002

……..…..

721321, Concurrent and Distributed Systems

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 4
Credit: 3 credit hours
Prerequisite: 750331

Aims:
The aim of this module is to study, learn, and understand the main concepts of concurrency. Hardware and software features to support concurrency, language features for concurrent and distributed systems, and concurrent and distributed algorithms and middleware.

Teaching Methods: 35 hours Lectures (2-3 hours per week) + 5 hours Laboratory (1 per 3 weeks) + 8 hours Seminars (1 each fortnight)

Learning Outcomes:
A student completing this module should be able to:
1. Outline the potential benefits of concurrent and distributed systems. (A).
2. Apply standard design principles in the construction of these systems. (A, B)
3. Select appropriate approaches for building a range of distributed systems, including some that employ middleware. (B)
4. Summarize the major security issues associated with distributed systems along with the range of techniques available for increasing system security. (A)

Assessment of Learning Outcomes:
Learning outcomes (1) – (4) are assessed by examinations, assignments, and seminars.

Contribution to Programme Learning Outcomes
A2, A5, B2, B3.

Synopsis: Concurrent model of execution; interleaving; atomic operation; critical sections and mutual exclusion; deadlock; starvation; invariants; Concurrent and distributed algorithms: producer-consumer; reader-writer problems; dining philosophers; Architectural features to support concurrent and distributed systems; Language features for concurrent and distributed systems; Performance evaluation; Middleware.

Modes of Assessment:
Two 1-hour midterm exams (15% each); Assignments (10%); Seminars (10%); Final Examination: 2-hours written exam (30%) + defended project (20%)

Textbooks and Supporting Material:
1 Jean Bacon; Concurrent Systems; Addison – Wesley; 1998
……..…..

750452, Knowledge-Based Systems

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 3
Credit: 3 credit hours
Prerequisite: 751252

Aims:
This module aims to give students the main concepts of knowledge-based systems, reasoning, knowledge representation, acquisition, implementation and verification.

Teaching Methods: 39 hours Lectures (2-3 per week) + 9 hours Seminars (3 per month)

Learning Outcomes:
A student completing this module should:
1. Be able to understand the knowledge-based systems representation (A, B).
2. Be able to understand automatic reasoning (A).
3. Be able to understand inductive and deductive learning (A).
4. Be able to implement a small knowledge- based system (C).

Assessment of Learning Outcomes:
Learning outcome (1) is assessed by examinations. Learning outcome (2) is assessed by examinations. Learning outcome (3) is assessed by examinations and seminars. Learning outcome (4) is assessed by assignments.

Contribution to Programme Learning Outcomes:
A 1, A2, A5, B1, B3, C1, C3

Synopsis: Introduction to knowledge-based systems; Logic and automatic reasoning; Forward and backward reasoning; Introduction to knowledge representation; knowledge representation models; Knowledge representation methods; Rules, frames; Semantic network for knowledge- based system; The software life cycle in knowledge-based system; Feasibility analysis; Analysis of knowledge acquisition; Knowledge acquisition methods; Knowledge elicitation; Instruments for Knowledge elicitation; Knowledge-based systems implementation; Knowledge-based systems verification and validation.

Modes of Assessment:
Two 1 hour midterm exams (15% each); assignments (10%), seminars (10%); Final exam: (2 hours) written exam (50%)

Textbooks and Supporting Material:
1- A.J. Gonzalez & D.D.Dankel, The Engineering of knowledge-based systems, Prentice Hall
2- D.Partridge, K.M. Hussain, Knowledge-based information systems, McGraw Hill
3- Guus Schreiber, Hans Akkermans,... Knowledge engineering and management
4- Peter Jakson, introduction to expert systems, Addison-Wesley
5- Georae F Luger, William Astubblefield, Artificial intelligence, structures and strategies for complex problem solving, Addison-Wesley.
……

750453, Genetic Algorithms

الشبكات العصبونية والخورزميات الجينية

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 4
Credit: 3 credit hours
Prerequisite: 750322 + 751252

Aims:
This module introduces genetic algorithms and their terminology and describes two provocative applications in detail; then it looks at the use of genetic algorithms in machine learning (computer programs, data analysis and prediction, neural networks) and in scientific models (interactions among learning, evolution, and culture; ecosystems; evolutionary activity).

Teaching Methods: 42 hours Lectures (2-3 hours per week) + 6 hours Seminars (last 2 weeks)

Learning Outcomes:
A student completing this module should:
1. be able to understand the theoretical foundation of Genetic Algorithms (A)
2. be able to solve problems using Genetic Algorithms. (B)
3. be able to implement a Genetic Algorithm (C)
4. be able to represent a seminar (C)

Assessment of Learning Outcomes:
Learning outcomes (1, 2, and 3) are assessed by assignments, homework, and examinations. Learning outcome (4) is assessed by seminars

Contribution to Programme Learning Outcomes:
A1, B3, C1, C2, C3

Synopsis: Genetic Algorithms: An Overview; Genetic Algorithms in Problem Solving; Genetic Algorithms in Scientific Models; Theoretical Foundations of Genetic Algorithms; Implementing a Genetic Algorithm.

Modes of Assessment:
Two 1-hour midterm exams (20% each); Assignments (5%); Seminars (5%); Final Examination: 2-hours written exam (35%) + Project (15%)

Textbooks and Supporting Material:

1- Melanie Mitchell, An Introduction to Genetic Algorithms, MIT Press, 1998.

……..…..

750481, Software Engineering and Formal Specifications

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 4
Credit: 3 credit hours
Prerequisite: 721281

Aims:
The module aims at introducing the students to the application of formal methods to the practice of software engineering. Formal Methods refer to a variety of mathematical modelling techniques, which are used both to model the behaviour of a computer system and to verify that the system satisfies design, safety and functional properties. This is a course in formal mechanisms for specifying, validating and verifying, and constructing correct software systems.

Teaching Methods: 36 hours Lectures (2-3 hours per week) + 8 hours Seminars (1 per 2 weeks) + 4 hours Laboratory (1 per 3 week)

Learning Outcomes:
At the end of this module, students will be able to:
1- Apply a range of formal specification techniques (Algebraic, Z, Statecharts, Petri nets). (A, B)
2- Apply a range of formal validation and verification techniques. (B)
3- Apply refinement techniques to generate implementation. (B, C)
4- Identify the benefits and limits of formal methods in software engineering. (B)
5- Practice formal methods tools. (C, D)

Assessment of Learning Outcomes:
Learning outcomes (1 – 3) are assessed by examinations and assignments. Learning outcomes (4 – 5) are assessed by assignments and seminars

Contribution to Programme Learning Outcomes
A3, B3, C5, C6, D2.

Synopsis: Introduction to formal methods: What do we mean by "formal method (FM)?", What are they for?, Distinguishing features of FMs (property oriented versus model oriented), FMs in the context of software life cycle, Myths of FMs; Algebraic method (1) : Foundations; Algebraic method (2) : Basic concepts; Algebraic method (3) : Advanced features; Algebraic method (4) : Validation & Verification technique; Algebraic method (5) : Refinement techniques; Algebraic method (6): languages, tools, environments; Z method (1): Foundations; Z method (2): Basic concepts of Z language; Z method (3): Advanced features of Z language; Z method (4): Z Validation & Verification techniques; method (5): Z refinement techniques; Z method (6): Z tools; Specification of concurrent reactive systems: Statecharts; Specification of concurrent reactive systems : Petri nets

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

Textbooks and Supporting Material:
1- Specification of Software Systems (Texts in Computer Science) (Hardcover) by V.S. Alagar, K. Periyasamy,, Springer Verlag, 1998
2- I. Van Horebeek & J. Lewi, Algebraic Specifications in Software Engineering, Springer Verlag, 1989
3- Jim Woodcock, Jim Davis. *Using Z: Specification, Refinement and Proof*. Prentice-Hall International 1996
4- V Heitmeyer and Mandrioli (eds), Formal methods for real-time computing, John Wiley, 1996
……..…..

750491, Special Topics

Providing Department: Computer Science, Faculty of IT
Module Coordinator(s):
Year: 4
Credit: 3 credit hours
Prerequisite: Department Agreement

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

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

Learning Outcome:
It depends on the chosen topic.

Assessment of Learning Outcome:
It depends on the chosen topic.

Modes of Assessment:
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

Digital Image Processing 750474
Course Description:

In many business and industry application domains handling and processing of digital images is a basic and important component. This spans an extremely wide range of applications from digital content creation and management for web-based applications over medical imaging, digital photography, digital print and publishing to real-time 3D graphics in simulations and computer games.

Course Objectives:

The main objective of this module is to make it possible to effectively communicate visual results. This course prepares students in the fundamentals of digital image processing as used in various applications as outlined above and illustrates the various effects one can achieve with digital images and how to extract fundamental information.

Course Components

Image Sensing and Representation, Image Analysis, CVIPlab, Human Visual Perception, Image Enhancement, Image Transforms, Image Restoration, Image Compression, Edge Detection, Correlation and Feature Detection, Discrete Fourier Transform and its Properties, Digital Filtering and Noise Cleaning, Image Segmentation.

Textbooks:
Title: Computer Imaging: Digital Image Analysis and Processing , SE Umbaugh,
Publisher: CRC Press, 2005

Course/Module Description:
This module focuses on problem solving strategies and the use of algorithmic language to describe such problem solving. It introduces the principles of procedural programming, data types, control structures, data structures and functions, data representation on the machine level. Various problems are considered to be solved using C-like procedural programming language.

Programming Fundamentals 750211

Course/Module Objectives:
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.

Course/ module components

- **Textbook:**

  D.S. Malik , Thomson, C++ Programming: From Problem Analysis to Program Design, Third Edition, Course Technology, 2007

- **Supporting material(s):** Lectures handouts

Teaching methods:
Duration: 16 weeks, 80 hours in total
Lectures: 32 hours (2 hours per week),
Tutorials: 16 hours (1 per week),
Laboratories: 32 hours, 2 per week