UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

PROSPECTUS

Academic Year 2013-2014

POSTAL ADDRESS
Department of
Computer Science
University of Cyprus
P.O. Box 20537
CY-1678 Nicosia
CYPRUS
Tel.: +357-22892700
Fax: +357-22892701
E-mail: cs@ucy.ac.cy
Web: http://www.cs.ucy.ac.cy
Twitter: @csdeptucy

Open Letter from the Chairperson of the Department

Dear Students,

This study guide contains a concise and comprehensive presentation of the Computer Science Department of the University of Cyprus. Furthermore, in this guide you will find information regarding your studies at the Department, including undergraduate and graduate curriculums, course descriptions, rules of attendance, prerequisites for graduation, short biographies of the faculty, etc. Up to date information, announcements, news, etc. are published through the website of the department at http://www.cs.ucy.ac.cy and our Twitter (@csdeptucy) or Facebook accounts; also through the e-mail and the announcement boards near the Secretariat of the Department. For personal advise on your studies, for scientific or professional orientation and on any other matter that may affect your studies, I encourage you to keep regular contact with the academic advisor who is assigned to you during your enrolment to the Department.

The Computer Science discipline, albeit a relatively young scientific field, has been recognized already as a key scientific and technological discipline with impressive development and a strong influence on the development of other sciences and the society at large. The growing significance and the applicability of Computer Science has established the Department of Computer Science as one of the most dynamic and competitive departments in the University of Cyprus. The skills and the high educational level that our students gain during their studies are acknowledged by both the local industries and worldwide, as our graduates achieve commendable performance either as employees in the domestic industry or as postgraduate students at leading Universities in Europe and America. As of the year 2012, the Department of Computer Science admits 90 new undergraduate students and approximately 35-40 graduate students. Currently, 350 undergraduate, 88 postgraduate and 40 PhD students are enrolled in the Department. Our Department is working systematically to attract outstanding students by organizing and supporting annual Computer Science seminars and competitions, such as the Workshop for High Scholl Students and the Logipaignion game-creation contest.

Both the undergraduate and postgraduate curriculums have been designed to meet the current challenges and advancements of Computer Science, and cover theoretical foundations, technical knowledge, and experimental methodologies of Computer Science with a strong problem-solving focus. The Department offers three directions in the undergraduate curriculum: Computer Science - General Direction, Direction of Computer Systems and Networks, and Direction of Software Engineering. Also, the Department offers a secondary Computer Science curriculum (minor) for the students of the other departments of the University of Cyprus. More details are included in this prospectus.

At the postgraduate level, our department offers a Ph.D. program in Computer Science and five Master's degree programs in: "Computer Science", "Internet Computing", "Intelligent Systems", a Professional Master in "Advanced Information Technologies," and an interdisciplinary, inter-University Master in the "Design and Development of Computer Games and Interactive Technologies", in collaboration with the Cyprus University of Technology.

Both the undergraduate and graduate curriculums comply with the European Credit Transfer and

Accumulation Credits (ECTS).

Our Department is in constant dialogue with industry, relevant government departments and centers of excellence abroad to contribute decisively in promoting Cyprus as a center of services, expertise and innovation in Computer, Information and Communication technologies. Also, the Department is working vigorously for the international promotion of our students' skills through attendance and awards in international competitions of programming competitions, innovation, entrepreneurship, etc.

In these efforts the strong presence of the Department in research is vital. In the twenty one years since its foundation, the Department has developed highly important activities and contributes to the promotion of science and knowledge worldwide. Particularly important is the participation of the Department in research programs of the European Union (EU), as well as domestic programs of the Research Promotion Foundation (RPF) of Cyprus. Specifically, since 2005, members of the Faculty of the Department have attracted more than 123 research projects by the EU and the RPF with a total budget exceeding 18,000,000 euros. Most of these funds has been used to subsidize hundreds of young scientists - researchers, doctoral students, postdoctoral associates and graduate students from Cyprus and abroad, to develop modern research and teaching infrastructure and the development of innovative software systems, hardware and applications. Research results are presented by faculty members and researchers of the Department each year in top journals and conferences of Computer Science and are exploited by the international scientific community and industry. Finally, a particularly valuable contribution of the Department is the support of governmental actions for the uptake of Information and Communication Technologies by the public sector, the promotion of the Internet and the Web in Cypriot businesses and society, and the technological modernization of Secondary Education and Health Services.

During your studies, I urge you to actively engage in the Departmental teaching and research activities to get the most knowledge and experience the Department can offer to you. Also, I invite you to take advantage of courses in other disciplines offered by the various Departments of the University, in order to enhance your knowledge and gain a more comprehensive and universal education. This feature is also available through the European Student Exchange Programme of ERASMUS; students of our University can complete part of their studies in universities of other European Union countries and vice versa.

Finally, on behalf of all the members of the Department, I once again welcome you and wish everyone a fruitful and productive year!

With best wishes,

Marios D. Dikaiakos, Professor

Chairman of the Department of Computer Science

TABLE OF CONTENTS

STAFF

Academic Faculty

Marios D. Dikaiakos *(Chairperson)*
Yannis Dimopoulos *(Vice-Chairperson)*
Chris Christodoulou
Yiorgos Chrysanthou
Paraskevas Evripidou
Chryssis Georgiou
Antonis Kakas
Georgia Kapitsaki
Elpida Keravnou-Papailiou
Marios Mavronicolas
George Pallis
Constantinos Pattichis
George Angelos Papadopoulos
Anna Philippou
Andreas Pitsillides
George Samaras
Yanos Sazeides
Christos N. Schizas
Pedro Trancoso
Vasos Vassiliou
Demetris Zeinalipour

Adjunct Faculty

Panos K. Chrysathis
(University of Pittsburgh, USA)
Petros Ioannou
(University of Southern California, USA)

Visiting Faculty

Georgios Chatzimilioudis

Special Teaching Staff

Pavlos Antoniou
Piros Bratsakas
Giorgos Hadjipollas
Yiannakis Mylonas
Christoforos Panagiotou
Petros Panayi

Secretariat

Dora Georgiou
Savvoula Efstathiou
Maria Kittira
Melina Menelaou-Chrysostomou

Administrative Support to Research Programmes

Zacharias Aristidou
Helen Kakos

Information Systems Technical Support

Andreas Kasenides
Andreas Kekkou
Andry Michaelidou
Savvas Nikiforou
Maria Tsiolakki
Andreas Filippou
Loizos Papadopoulos

Academic Faculty Information

Academic Faculty

Chris Christodoulou
Associate Professor
Office: FST 01 B113
Phone: +357-22892752
Email: cchrist@cs.ucy.ac.cy

Yiorgos Chrysanthou
Associate Professor
Office: FST 01 013
Phone: +357-22892719
Email: yiorgos@cs.ucy.ac.cy

Marios D. Dikaiakos
Professor
Office: FST 01 012
Phone: +357 22892720
Email: mdd@cs.ucy.ac.cy

Yannis Dimopoulos
Associate Professor
Office: FST 01 014
Phone: +357-22892718
Email: yannis@cs.ucy.ac.cy

Paraskevas Evripidou
Professor
Office: FST 01 115
Phone: +357-22892696
Email: skevos@cs.ucy.ac.cy

Chryssis Georgiou
Associate Professor
Office: FST 01 010
Phone: +357-22892745
Email: chryssis@cs.ucy.ac.cy

Georgia Kapitsaki
Lecturer
Office: FST 01 119
Phone: +357-22892692
Email: gkapi@cs.ucy.ac.cy

Antonis Kakas
Professor
Office: FST 01 110
Phone: +357-22892706
Email: antonis@cs.ucy.ac.cy

Elpida Keravnou-Papailiou
Professor
Office: FST 01 117
Phone: +357-22892694
Email: elpida@cs.ucy.ac.cy

Marios Mavronicolas
Professor
Office: FST 01 106
Phone: +357-22892702
Email: mavronic@cs.ucy.ac.cy

George Pallis
Lecturer
Office: FST 01 B119
Phone: +357-22892743
Email: gpallis@cs.ucy.ac.cy

George Angelos Papadopoulos
Professor
Office: FST 01 118
Phone: +357-22892693
Email: george@cs.ucy.ac.cy

Constantinos Pattichis
Professor
Office: FST 01 114
Phone: +357-22892697
Email: pattichi@cs.ucy.ac.cy

Anna Philippou
Associate Professor
Office: FST 01 105
Phone: +357-22892699
Email: annap@cs.ucy.ac.cy

Andreas Pitsillides
Professor
Office: FST 01 116
Phone: +357-22892695
Email: cspitsil@cs.ucy.ac.cy

George Samaras
Professor
Office: FST 01 113
Phone: +357-22892698
Email: cssamara@cs.ucy.ac.cy

Yanos Sazeides
Associate Professor
Office: FST 01 109
Phone: +357-22892704
Email: yanos@cs.ucy.ac.cy

Christos N. Schizas
Professor
Office: FST 01 111
Phone: +357-22892705
Email: schizas@ucy.ac.cy

Pedro Trancoso
Associate Professor
Office FST 01 107
Phone: +357-22892703
Email: pedro@cs.ucy.ac.cy

Vasos Vassiliou
Assistant Professor
Office: FST 01 B114
Phone: +357-22892750
Email: vasosv@cs.ucy.ac.cy

Demetris Zeinalipour
Assistant Professor
Office: FST 01 B106
Phone: +357-22892755
Email: dzeina@cs.ucy.ac.cy

Visiting Faculty

Georgios Chatzimilioudis
Office: FST 01 B118
Phone: +357-22892746
Email: gchatzim@cs.ucy.ac.cy

Special Teaching Staff

Pavlos Antoniou
Office: FST 02 B130
Phone: +357-22893927
Email: paul.antoniou@cs.ucy.ac.cy

Pyrros Bratskas
Office: FST 02 059
Phone: +357-22893930
Email: bratskas@cs.ucy.ac.cy

Giorgos Hadjipollas
Office: FST 02 B205
Phone: +357-22892717
Email: hpollas@cs.ucy.ac.cy

Yiannakis Mylonas
Office: FST 02 B205
Phone: +357-22893931
Email:mylonasy@cs.ucy.ac.cy

Petros Panagi
Office: FST 02 059
Phone: +357-22893926
Email: petrosp@cs.ucy.ac.cy

Christophoros Panagiotou
Office: FST 02 B130
Phone: +357-22893928
Email: cs95gp1@cs.ucy.ac.cy

Information Systems Technical Support

Andreas Filippou
Assistant IT Officer
Office: FST 01 001
Phone: +357-22892711
Email: andreasf@cs.ucy.ac.cy

Andreas Kasenides
Senior IT Officer
Office: FST 01 018
Phone: +357-22892714
Email: kasenid@cs.ucy.ac.cy

Andreas Kekkou
IT Officer
Office: FST 01 001
Phone: +357-22892728
Email: kekkou.a@cs.ucy.ac.cy

Andry Michaelidou
IT Officer
Office: FST 01 B104
Phone: +357-22892734
Email: andrim@cs.ucy.ac.cy

Savvas Nikiforou
IT Officer
Office: FST 01 006
Phone: +357-22892730
Email: savvasn@cs.ucy.ac.cy

Loizos Papadopoulos
Assistant IT Officer
Office: FST 01 001
Phone: +357-22892711
Email: louispap@cs.ucy.ac.cy

Maria Tsiolakki
Assistant IT Officer
Office: FST 01 001
Phone: +357-22892727
Email: tmaria@cs.ucy.ac.cy

Administrative Support to Research Programmes

Zacharias Aristidou
University Officer
Office: FST 01 017
Phone: +357-22892670
Email: zacharia@cs.ucy.ac.cy

Helen Kakos
University Assistant Officer
Office: FST 01 017
Phone: +357-22892713
Email: kakos@cs.ucy.ac.cy

Secretariat

Melina Menelaou Chrisostomou
Secretary
Office: FST 01 019
Phone: +357-22892721
Email: melina@cs.ucy.ac.cy

Savvoula Efstathiou
Secretary
Office: FST 01 009
Phone: +357-22892669
Email: savvoula@cs.ucy.ac.cy

Dora Georgiou
Secretary
Office: FST 01 019
Phone: +357-22892722
Email: addora@cs.ucy.ac.cy

Maria Kittira
Secretary
Office: FST 01 007
Phone: +357-22892700
Email: manak@cs.ucy.ac.cy

ACADEMIC CALENDAR **2013-2014**

	Fall Semester 2013-2014	**Spring Semester 2013-2014**
Registration and orientation of new entering students	26/08/2013 – 30/08/2013	13/01/2014 – 17/01/2014
Classes begin	02/09/2013	20/01/2014
Deadline for course selection	13/09/2013	31/02/2014
Deadline for dropping a course	20/09/2013	07/02/2014
Deadline for withdrawal	18/10/2013	07/03/2014
End of classes	29/11/2013	02/05/2014
Easter Holidays		14/04/2014 – 27/04/2014
Reading period	02/12/2013-06/12/2013	05/05/2014-09/05/2014
Examination Period	07/12/2013-22/12/2013	10/05/2014-25/05/2014
Christmas Holidays	23/12/2013–12/01/2014	
Public Holidays	1st October 28th October 6th Jaunuary	3rd March 25th March 1st April 20th April (Easter) 1st May

Introduction

The Computer Science field at its early days of birth had a specific scope, mainly concerned with the automation of mathematical calculations; quickly though, it developed into an exciting amalgam of science, technology, and theory. Today, Computer Science addresses a wide range of topics such as extending the range of problems which can be effectively solved using computers; creation, maintenance, and optimization of software and hardware for high performance computer systems; the way in which humans reason, discuss, and plans their activities; modeling the operation of the human brain; and the role of language and logic in problem-solving. Consequently, Computer Science has direct links with all of the Natural Sciences, and with many other branches of human knowledge such as Philosophy, Psychology, Cognitive Sciences, Linguistics, and Management. This multidisciplinary nature of our relatively young field inherits unlimited opportunities for discovering new applications and exploring new research directions.

Nowadays, Computer Science influences essentially every field of human endeavor, with important applications in industry, commerce, economics, education, and medicine. The fast technological growth and the wide spread of computers in our lives has significantly improved the quality of our lives by offering a huge variety of new services such as performing tasks which are too dangerous or too complicated for humans and contributing to the dissemination of knowledge and expertise via knowledge-based systems. Computer networks and distributed processing techniques are widely used and have become essential in today’s working environment. The initial misapprehension that people would be replaced by computers has now been dispelled by the realization that computers merely amplify people’s natural abilities and provide assistance in performing their work more effectively.

The ECTS System

As of the academic year 2005-2006, the University of Cyprus has adopted the European Community Course Credit Transfer and Accumulation System, or ECTS for short.

What is ECTS?

ECTS, or *European Credit Transfer and Accumulation Credits*, was developed by the Commission of the European Communities in order to provide common procedures to guarantee academic recognition of studies uniformly throughout Europe. It provides a way of measuring and comparing learning achievements and transferring them from one European institution to another.

ECTS provides transparency through the following means:

- *ECTS Credits* which are a numerical value allocated to course units to describe the student *workload* required to complete them.
- The *Information Package,* which supplies written information to students and staff on institutions, departments/faculties, the organizations and structure of studies and course units.
- The *Diploma Supplement*, which shows students' learning achievements in a way which is comprehensive, commonly understood and easily transferable from one institution to another.

The ECTS Credits

ECTS credits are allocated units to describe the student *workload* required to complete them. They reflect the *quantity* of work each course requires *in relation* to the total quantity of work required to complete a full year of academic study at the institution that is, lectures, practical work, seminars, private work – in library or at home – and examinations or other assessment activities. ECTS credits express a *relative value*. In ECTS, *60 credits* represent the workload of a year of study and *30 credits* are given for a semester.

Credits are awarded only when the course has been completed and all required examinations have been successfully taken.

The expected time for the completion of the Undergraduate Program of Studies is eight semesters. This time can be extended up to twelve semesters. A student is required to complete 240 ECTS units in order to acquire a Batchelor degree in Computer Science at the Undergraduate Programme of Studies. Furthermore for a Master Degree , successful completion of at least 90 ECTS is required with the excemption of the Professional Master in Advanced Information Technologies that requires successful completion of at least 75 ECTS.

General Information

Class Attendance and Teaching

Teaching is carried out in the internationally established way through lectures, tutorials, laboratories, etc. Students are expected to consistently participate in all activities of classes they are attending (e.g., lectures, tutorials, laboratories, etc.). The Department reserves the right to disallow students who systematically skip the activities of attended classes from taking the corresponding final examinations.

In most classes, homework is assigned regularly to help students master the course material and develop practical skills. Homework is carried out either individually or collaboratively small student groups.

Evaluation is usually based on assigned homework, written and oral examinations, etc. Special efforts are made to use continuous evaluation whenever possible.

Students are informed about the specific procedure of teaching and evaluation for each class from its class Instructor. The relevant information is included in the "General Information" sheet for the class, which is handed out to the students during the first week of classes each semester. The Instructor is available to the students during predetermined weekly office hours.

The general rules of studies of the University, the rules that govern the rights and responsibilities of the students, as well as the procedures that need to be followed, are included in separate leaflets. These are made available to the students by the Administrative Service of Academic Affairs and Student Welfare. Student must know all the rules that concern them.

Academic Advisor

A member of the Department's academic staff is assigned to each student as an Academic Advisor. The Academic Advisor provides assistance or advice on academic matters (e.g., difficulties in following a class, appropriate combinations of restricted choice classes, etc.), but also advice on other matters such as personal problems.

The role of the Academic Advisor can not be effective without the cooperation of the student. So, both undergraduate and graduate students, especially the freshmen, are strongly encouraged to have frequently meetings with their academic advisors, in order to achieve better preparation and organization of their program of study as well as to resolve related issues and problems. It should be noted that the students are held responsible for their final decisions regarding their academic choices. However, they should keep informed their academic advisor of their decisions.

Student Representation

Six elected representatives of the students are currently participating in the Departmental Board as members. Furthermore, during each Fall Semester, the Department requests that students elect two representatives per year in order to facilitate intercommunication between the year's students and the Department.

Secretariat

Daily activities can be carried out through the Departmental Secretariat. Students are informed by their Academic Advisors on services or other facilities the Secretariat can provide as well as about the specific timings the Secretariat is available to the Students. In particular, the Secretariat can provide any kind of general information regarding the Department or the University.

Course Schedule

The Class Schedule is not included in this Prospectus, but it is handed out to the students at the beginning of each Semester. Although every possible effort is made to satisfy the scheduling of any reasonable combination of classes, there is, however, a possibility that some students will face difficulties in combining certain of their (restricted or unrestricted) choices due to, e.g., concurrent class meetings. In such cases, students are urged to immediately notify their Academic Advisors.

Library Use

The University's Library is equipped with a large number of books and scientific journals in Computer Science. These include books that will assist the students in mastering the taught course materials, and, in particular, the books recommended or suggested by Instructors. Therefore the students should have their student ID in order to have access to the library facilities. Regulations for using the Library are separately handed out to the students at the beginning of each academic year.

Electronic Notices / Departmental Website

The Department uses electronic mail for the fast dissemination of information among the students and its Faculty/Staff members.

Also, all necessary information (courses, talks, announcements, etc) can be found at the Departmental website (http://www.cs.ucy.ac.cy). There is also a spectial section for students called Student's Corner where students can obtain up-to-date important academic information. Furthermore, the department is using social media (facebook and twiter pages) for the better communication among its members.

Electronic Mail

The use of Electronic Mail for communication between academic staff and students, and also between students themselves, is considered a must, and students are strongly encouraged to learn as soon as possible how to effectively use Electronic Mail. However, it is emphasized that the use of Electronic Mail is not a right of the students, but rather a service provided by the Department. In cases where inappropriate use of Electronic Mail is noticed, the Department reserves the right to disallow its use.

Laboratory Equipment

A significant part of our Curricula is of practical content. The Department has five laboratories, fully equipped for teaching and research in Computer Science.

- The *Postgraduate Laboratory (201)* consists of 30 last generation computers with Microsoft Windows 7 Enterprise operating system and special software for the support of specialized courses at the graduate level and, more specifically, in the directions of Internet Technologies and Intelligent Systems.
- The *General Teaching Unix Laboratories (B103)* and (103) consists of 33 workstations of the latest generation equipped with the Linux (CentOS 6.0) operating system and Free NX server, and they provide software for the development of various

applications.

- The *General Undergraduate Laboratory 1 (B121)* consists of 30 workstations of the latest generation, equipped with the operating system Windows 7, and provided with software for the development of various applications.
- The *General Undergraduate Laboratory 2 (B123)* consists of 31 workstations each, of the latest generation, equipped with the operating system Windows 7, and it provides software for the development of various applications.
- The *Digital Systems Design and Microprocessor Laboratory (101)* consists of 27 last generation computers with Microsoft Windows 7 Enterprise operating system. Furthermore, there are digital logic and microprocessor boards, oscilloscopes, digital multimeters, logic probes, signal generators and integrated circuit testers. This equipment is used for practical training in hardware. In addition, this Laboratory includes peripheral devices for training in the design of microprocessor systems, and in the development of prototypes.
- The *Walk-in Laboratory (B101)*, in which both personnel and students may bring their portable computers and connect to the Internet and the departmental computer systems.
- The *Tele-education room (148)* is a lecture room of 20 seats and it can be used for: (i) data distribution to remote users, (ii) live and on demand streaming of lectures and presentations, (iii) lecture capturing on DVDs.

More information on the equipment is provided on the website of the Information Systems Technical Support (http://its.cs.ucy.ac.cy/).

Laboratory Equipment Use Regulations

Students are kindly requested to respect fundamental principles of professional behavior regarding health and safety in common rooms and responsible use of laboratory equipment. In particular, the following are indicative examples of disallowed tasks:

- Access to computer systems using somebody else’s account
- Inappropriate use of Electronic Mail
- Use of computer systems for purposes other than the “normal” (e.g., development of commercial products, disturbance of users, etc.)
- Use of software products other than those provided by the Computer Center, without the consent of the Computer Center or the Department, respectively.
- Attempt to access confidential information
- Copying software products that belong to others, in violation of international laws of liability.

Specific information regarding Regulations and Time Table for using the Computer Laboratories are handed out to the students at the beginning of each academic year.

Awards

For each academic year, the Department grants, on the basis of academic merit, awards to its undergraduate students. These awards have been founded by external sponsors as follows:

[1] *Award of the Pancypriot Cooperative Computer Society (SEM) Ltd* to the graduating student who has attained the highest overall academic performance. (Monetary award of 1200 Euro)

[2] *Award of the Pancypriot Cooperative Computer Society (SEM) Ltd* to the graduating student who has attained the second highest overall academic performance. (Monetary award of 800 Euro)

[3] *Award of Cyprus Telecommunications Authority* to the graduating student who has attained the highest overall academic performance. (Monetary award of 1000 Euro) (not offered in June 2013)

[4] *Award of Union of Scientific Staff of Electricity Authority of Cyprus* to the graduating student who has attained the highest overall academic performance. (Monetary award of 350 Euro)

Also, the Department itself grants the following awards to its undergraduate students:

[1] *Special Award* to the undergraduate student who is completing the Minor Program of Studies in Computer Science and has attained the highest overall academic performance in the courses of the minor program. (Monetary award of 340 Euro)

[2] *Special Award* for the graduating student who has demonstrated exceptional social contribution. (Monetary award of 340 Euro)

[3] *Award* for the graduate student who has demonstrated outsanding and praise-deserving effort in Memory Polyvios Polyviou. (Monetary award of 500 Euro)

[4] *Department Award of excellence* to the graduating students of the Departmer of Computer Science who have achieved an excellent performance . (Over 8.5/10) (Monetary award of 200 Euro/each)

Also, the Department grants, every year, the following awards to its postgraduate students:

[1] *Award of Mariou Tsakalaki*, donated by the family of the deceased Marios Tsakalakis, for the graduating student who has attained the highest overall academic performance. (Monetary award of 500 euro)

[2] *1st Award* for the graduating student who has attained the highest overall academic performance. (Monetary award of 680 euro)

[3] *2nd Award* for the graduating student who has attained the second highest overall academic performance. (Monetary award of 340 euro)

Administrative Duties of Academic Staff

Committees/Groups

[1] **Postgraduate Program Committee:** *P. Trancoso (coordinator), Chr. Christodoulou, G. Samaras*

[2] **Undergraduate Program:** *Ch. Georgiou (coordinator), C. Pattichis, A. Philippou*

[3] **Undergraduate Transfers Committee and Correspondence Students:** *G. Papadopoulos (coordinator), Y. Chrysanthou, G. Samaras*

[4] **Program Erasmus/Socrates:** *Y. Sazeides (coordinator), G. Kapitsaki, P.Trancoso*

[5] **Communication and Promotion:** *G. Kapitsaki (publication coordinator), G. Pallis (website coordinator)*

[6] **Computer Systems Committee:** *M. Dikaiakos (coordinator), P. Evripidou, D. Zeinalipour, V. Vasiliou*

[7] **Logipaignion Contest:** *Y. Chrysanthou, G. Pallis, V. Vasiliou, D. Zeinalipour*

[8] **Committe of International Olympics of Informatics:** *Y. Sazeides (coordinator), D. Zeinalipour, G. Papadopoulos*

Representatives/Coordinators:

[1] **Library:** *A. Philippou*

[2] **Diploma Project:** *A. Philippou*

[3] **Teaching Assistants:** *G. Samaras*

[4] **Departmental Seminars:** *D. Zeinalipour*

Undergraduate Program of Studies

According to the ECTS System, at least 240 ECTS units are required for acquiring a Batchelor's degree in Computer Science.

Objectives and Prospects

The Undergraduate Program of Studies leads to a Batchelor's degree in Computer Science. The Department aspires to prepare graduates able to pursue careers in positions of responsibility in either academia or industry, where they will effectively drive the development and application of new methods and ideas. The Department attaches particular importance to maintaining links and continuous dialogue with local industry, and we expect our graduates to play a significant role in this respect.

Any curriculum, no matter what its subject, must offer the student education in the widest sense of the term, and cultivate the desire for continuous learning, which, in turn, leads to maturity and develops the facilities for independent and critical thinking. This general objective coexists with and enriches the more specific objectives of individual programs.

We hope that our graduates will acquire a deep understanding of Computer Science, both as a science, and in terms of its more general applications and effects on society. The Curriculum covers the essential practical techniques, together with the deeper principles which underpin them. Our graduates will be well qualified to obtain immediate employment in Cyprus or abroad as IT professionals, or as teachers of Computer Science in secondary education; alternatively, they may choose to continue their studies to postgraduate level, and further pursue a career in either academia or research. Whichever career path they choose, their studies will have given them the necessary grounding to keep abreast of the incredibly rapid scientific and technological developments in Computer Science.

The expected time for the completion of the Undergraduate Program of Studies is eight semesters. This time can be extended up to twelve semesters.

Course Areas

The Department's undergraduate course material is conceptually divided into the following areas or components: *Theory, Computer Systems, Problem Solving* and *Applications.*

The *Theory* component is concerned with the foundations of Computer Science: theory and models of computation, and the design and analysis of algorithms. Generally this component aims at cultivating a formal approach to thinking, and organizing and processing information. Logic, and its role as the calculus of Computer Science, is an important topic. Essential concepts of Discrete Mathematics are taught as an integral part of related courses offered by the Department. Students are also required to follow a number of courses offered by the Department of Mathematics and Statistics in order to develop their abilities for abstraction and formal thinking, and to acquire other useful mathematical skills.

The *Computer Systems* component is concerned with hardware and software systems and elaborates on the concepts of parallel and embedded systems. It includes basic principles of computer architecture and organization, operating systems, programming language design and implementation, microprocessor systems, data communications, networks, distributed systems, and parallel and novel architectures.

The *Problem Solving* component aims at developing algorithmic thinking, with emphasis on principles of programming and program design. Through this component, students will acquire competence in a number of programming languages using a variety of programming paradigms (imperative, object-oriented, logic-based). Concurrent, parallel, distributed and heuristic techniques of problem solving are addressed in restricted choices. In this component, students learn how to design, implement and evaluate solutions to significant, albeit relatively small, problems. The wider integration of these techniques in building methodological frameworks to solve real-world problems is studied in courses on systems analysis and design, and software engineering as indicated in the Applications component.

The *Applications* component aims at bringing together the knowledge and skills acquired in the other three components for the development of useful applications to solve "real-world problems". Important technologies such as databases, knowledge bases, graphics and user interface managers are introduced as examples of applications in themselves and as vital tools for the construction of higher level applications. Modern Software Engineering methodologies that address every stage in the planning, design, development, and maintenance of high-quality applications are studied, and subsequently put into practice in the context of a group project undertaken in the framework of the course Software Engineering Professional Practice. Finally, important social and ethical issues concerned with the spread of computers are raised and discussed.

The Undergraduate Program of Studies consists of *Compulsory Courses* covering core material, *Restricted Choices* in Computer Science or related subjects, and *Unrestricted Choices* offered by other Departments. Some courses have other courses as prerequisites. The course dependencies between the compulsory courses are depicted in Table 1. Almost all courses offered during the first six semesters are compulsory, whereas the last two semesters comprise mostly Restricted Choices, in several topics in Computer Science.

The Computer Science Department offers to students three directions for their studies:

- Computer Science: General Direction,
- Direction of Computer Systems and Networks and
- Direction of Software Engineering

The direction of **Computer Science: General Direction** aims to offer to its graduates a broad background in the essential practical techniques together with their underlying principles. This will enable them to develop technical expertise, professional skills, and critical thinking.

The graduates of this direction may find employment in the local or European market as Information Technology scientists in a variety of professions (e.g. database management, network management, software development) or deal with education. Also, they can continue their studies by pursuing a Master's degree or by conducting research in a wide range of modern sectors such as the *Intelligent Information Systems, Distributed Systems, Graphics, Machine Learning and Data Mining*.

The direction of **Computer Systems and Networks** aims to present the basic principles and recent developments in the organization and programming of computer systems and communication networks. Through both course work and laboratories, students are given the opportunity to cover topics related to this specialization, with emphasis on parallel and embedded systems, high-performance networks and ubiquitous communication and networking.

Areas of Computer Systems and Networks are traditionally amongst the most attractive subjects of study due to the need for substantial knowledge of the organization and operation of a system or network for its efficient usage. Nowadays, this need has become even more urgent with the growing diffusion of computer systems and networks in every aspect of everyday life in the modern information society and communication. Specialization on systems and networks provide graduates a wide range of choices for academic, research and professional careers.

The direction of **Software Engineering** aims at preparing students for becoming Software Engineers. In the software industry familiarity with software development processes and experience using one or more programming languages are essential competences. Students will have the opportunity to gain knowledge on the tools of the software engineering industry, work in groups and implement software products. Specialization on Software Engineering will give students the necessary background for employment in the software industry from a professional or research perspective.

In the framework of the direction the principles and advancements in the field of software engineering will be presented, ranging from the concept to the provision of the software system. The aim is to learn to move between theory and practice and be able to acquire special domain knowledge for the purpose of supporting software development in specific domains of application.

Almost all courses offered by the Department in the first four semesters are compulsory and common to all students. Students are asked to choose a direction at the beginning of the fourth semester of their studies. In the fifth and sixth semesters, students are required to take the compulsory courses of their chosen direction, as well as elective courses from other departments. The last two semesters include restricted elective courses (three courses from the chosen direction and any two courses in Computer Science 300 or 400 level) within the direction and elective courses; also, each student must undertake an individual Diploma Project, under the supervision of a member of the Department's academic staff. The topic of the Undergraduate Thesis Project must relate to the student's chosen direction of study. Some courses are a prerequisite for other courses; these dependencies are shown in Table 1.

Curriculum

Each course code has the form XYZ, where X represents the type of the course and Y the area which it belongs to. Courses offered for students of other Departments have, by default, type 0. Compulsory courses are of 1st, 2nd or 3rd type, restricted elective courses are of 4th type while the Diploma Project is of 4th type. Course areas have code 1 (Theory), 2 (Computer Systems), 3 (Troubleshooting) and 4 (Applications). General content courses have area code 0. The direction of Software Engineering has code 6 and the direction of Computer Systems and Networks has code 7. The General Direction has not gotten any code, since the courses are offered from other areas.

Below we cite the indicative programs of the two directions that our Department offers this academic year 2013/20134.

Indicative Program
Computer Science: General Direction

Semester	Curriculum	ECTS
First Semester	CS111 Discrete Structures in Computer Science and Computation	7,5
	CS131 Programming Principles I	7,5
	MAS012 Calculus for Computer Scientists I	5
	LAN100 General Advanced English	5
	PBA101 Principles of Management	5
Second Semester	CS121 Digital Systems	7,5
	CS132 Programming Principles II	7,5
	MAS013 Calculus for Computer Scientists II	5
	LAN111 English for Computer Science	5
	Elective Course from Departments of Physics, Chemistry, Biological Sciences	5
Third Semester	CS202 Explorations into Computer Science	3
	CS221 Computer Organization and Assembly Programming	7,5
	CS231 Data Structures and Algorithms	7,5
	MAS016 Linear Algebra I for Computer Science	6
	MAS055 Introduction to Probability and Statistics	7
Fourth Semester	CS211 Theory of Computation and Complexity	7,5
	CS222 Operating Systems	7,5
	CS233 Object Oriented Programming	7,5
	CS241 Systems Analysis & Design	7,5
Fifth Semester	CS324 Communications and Networks	7,5
	CS342 Database Systems	7,5
	CS361 Software Engineering	7,5
	Elective Course	5
Sixth Semester	CS323 Theory and Practise of Compilers	7,5
	CS336 Algorithms and Complexity	7,5
	CS341 Artificial Intelligence	7,5
	Elective Course	5
	Elective Course	5
Seventh Semester	CS400 Diploma Project I	7,5
	Restricted Elective Course within the Direction	7,5
	Restricted Elective Course within the Direction	7,5
	Restricted eEective Course - 300 or 400 type in Computer Science	7,5
Eighth Semester	CS401 Diploma Project II	10
	Restricted Elective Course within the Direction	7,5
	Restricted Elective Course - 300 or 400 type in Computer Science	7,5
	Elective Course	5

Elective Courses from Physics, Chemistry, and Biological science departments

BIO102	Integrative Biology of Organisms
PHY131	General Physics I: Mechanics, Waves and Thermodynamics
PHY132	General Physics II: Electricity, Electromagnetic and Optics
CHE121	Intoduction to Chemistry

Restricted Elective Courses for the Computer Science: General Direction

CS362	Software Engineering II
CS363	Professional Practice in Software Engineering
CS371	Systems Programming
CS372	Parallel Processing
CS375	Advanced Networks
CS412	Logic in Computer Science
CS413	Computational Geometry
CS425	Internet Technologies
CS426	Computer Graphics
CS431	Synthesis of Parallel Algorithms
CS432	Distributed Algorithms
CS433	Constraint Satisfaction and Programming
CS434	Logic Programming and Artificial Intelligence
CS442	Computational Learning Systems
CS444	Computational Intelligent Systems
CS445	Digital Image Processing
CS446	Advanced Topics in Databases
CS447	Computer Vision
CS463	Software Reuse

Indicative Program
Computer Systems and Networks Curriculum

Semester	Curriculum	ECTS
First Semester	CS111 Discrete Structures in Computer Science and Computation	7,5
	CS131 Programming Principles I	7,5
	MAS012 Calculus for Computer Scientists I	5
	LAN100 General Advanced English	5
	PBA101 Principles of Management	5
Second Semester	CS121 Digital Systems	7,5
	CS132 Programming Principles II	7,5
	MAS013 Calculus for Computer Scientists II	5
	LAN111 English for Computer Science	5
	Elective Course from Departments of Physics, Chemistry, Biological Sciences	5
Third Semester	CS202 Explorations into Computer Science	3
	CS221 Computer Organization and Assembly Programming	7,5
	CS231 Data Structures and Algorithms	7,5
	MAS016 Linear Algebra I for Computer Science	6
	MAS055 Introduction to Probability and Statistics	7
Fourth Semester	CS211 Theory of Computation and Complexity	7,5
	CS222 Operating Systems	7,5
	CS233 Object Oriented Programming	7,5
	CS241 Systems Analysis & Design	7,5
Fifth Semester	CS324 Communications and Networks	7,5
	CS342 Database Systems	7,5
	CS370 Computer Architecture	7,5
	Elective Course	5
Sixth Semester	CS371 Systems Programming	7,5
	CS372 Parallel Processing	7,5
	CS375 Advanced Networks	7,5
	Elective Course	5
	Elective Course	5
Seventh Semester	CS400 Diploma Project I	7,5
	Restricted Elective Course within the Direction	7,5
	Restricted Elective Course within the Direction	7,5
	Restricted Elective Course - 300 or 400 type in Computer Science	7,5
Eighth Semester	CS401 Diploma Project II	10
	Restricted Elective Course within the Direction	7,5
	Restricted Elective Course - 300 or 400 type in Computer Science	7,5
	Elective Course	5

Elective Courses from
Physics, Chemistry, and Biological science departments

BIO102	Integrative Biology of Organisms
PHY131	General Physics I: Mechanics, Waves and Thermodynamics
PHY132	General Physics II: Electricity, Electromagnetic and Optics
CHE121	Intoduction to Chemistry

Restricted Elective Courses:
Computer and Networks Curriculum

CS323	Theory and Practise of Compilers
CS361	Software Engineering I
CS424	Digital Signal Processing
CS425	Internet Technologies
CS431	Synthesis of Parallel Algorithms
CS432	Distributed Algorithms
CS445	Digital Image Processing
CS451	Data Mining in the WWW
CS470	Design of Embedded Systems
CS475	Network and Information Security
CS476	Wireless Networks
ECE406	VLSI Design (6 ECTS)*
ECE408	Digital Design with FPGA (6 ECTS)*
ECE427	Embedded and Real-Time Systems (6 ECTS)*
ECE453	Wireless Telecommunications Networks (6 ECTS)*

* Students may select a course (6 ECTS) offered by the Department of Electrical and Computer Engineering as one of the required restricted elective courses of the direction. In order to cover the remaining 1.5 ECTS, students will have to take an independent study course.

Indicative Program
Software Engineering Curriculum

Semester	Curriculum	ECTS
First Semester	CS111 Discrete Structures in Computer Science and Computation	7,5
	CS131 Programming Principles I	7,5
	MAS012 Calculus for Computer Scientists I	5
	LAN100 General Advanced English	5
	PBA101 Principles of Management	5
Second Semester	CS121 Digital Systems	7,5
	CS132 Programming Principles II	7,5
	MAS013 Calculus for Computer Scientists II	5
	LAN111 English for Computer Science	5
	Elective Course from Departments of Physics, Chemistry, Biological Sciences	5
Third Semester	CS202 Explorations into Computer Science	3
	CS221 Computer Organization and Assembly Programming	7,5
	CS231 Data Structures and Algorithms	7,5
	MAS016 Linear Algebra I for Computer Science	6
	MAS055 Introduction to Probability and Statistics	7
Fourth Semester	CS211 Theory of Computation and Complexity	7,5
	CS222 Operating Systems	7,5
	CS233 Object Oriented Programming	7,5
	CS241 Systems Analysis & Design	7,5
Fifth Semester	CS324 Communications and Networks	7,5
	CS342 Database Systems	7,5
	CS361 Software Engineering I	7,5
	Elective Course	5
Sixth Semester	CS362 Software Engineering II	7,5
	CS363 Professional Practice in Software Engineering	7,5
	CS371 Systems Programming	7,5
	Elective Course	5
	Elective Course	5
Seventh Semester	CS400 Diploma Project I	7,5
	Restricted Elective Course within the Direction (group A)	7,5
	Restricted Elective Course within the Direction (group A or group B)	7,5
	Restricted Elective Course - 300 or 400 type Computer Science	7,5
Eighth Semester	CS401 Diploma Project II	10
	Restricted Clective Course within the Direction (group A or group B)	7,5
	Restricted Elective Course - 300 or 400 type Computer Science	7,5
	Elective Course	5

Elective Courses from
Physics, Chemistry, and Biological science departments

BIO102	Integrative Biology of Organisms
PHY131	General Physics I: Mechanics, Waves and Thermodynamics
PHY132	General Physics II: Electricity, Electromagnetic and Optics
CHE121	Intoduction to Chemistry

Elective Courses in Software Engineering (Group A)

CS461	Software Validation, Verification and Quality
CS462	Software Analysis, Modelling and Design
CS463	Software Reuse

Elective Courses in Software Engineering (Group B)

CS323	Theory and Practice of Compilers
CS336	Algorithms and Complexity
CS372	Parallel Processing
CS425	Internet Technologies
CS431	Synthesis of Parallel Algorithms
CS432	Distributed Algorithms
CS435	Human Computer Interaction
CS446	Advanced Topics in Databases
CS451	Data Mining in the World Wide Web

Regarding the distribution of elective courses and restricted elective courses in the 3rd and 4th year of their studies in both semesters, the student has the possibility to choose the following alternative program of study, is he/she wishes to (regardless of the selected direction):

5th Semester 3 Compulsory Courses
1 Restricted Elective Course
6th Semester: 3 Compulsory Courses
1 Restricted Elective Course
7th Semester: Diploma Project I
1 Restricted Elective Course
3 Elective Courses
8th Semester: Diploma Project II
2 Restricted Elective Courses
1 Elective Course

The aforementioned restricted elective courses refer either to “restricted elective course within the direction” of to “restricted elective course - 300 or 400 type Computer Science course” as indicated in the indicative program of each direction provided that the total number of required courses is maintained.

Concerning the courses of the 4th year of their studies students have also the possibility to choose a 2nd “restricted elective course - 300 or 400 type Computer Science course” in the 7th semester chossing the required “restricted elective course within the direction” in the 8th semester.

Restricted Elective Courses

In consultation with her/his academic advisor, each student selects Restricted Choices according to her/his interests and professional goals. A student may choose to expand her/his knowledge in one specific area, or may select a combination of courses in more than one area. Thus the Restricted Choices may be selected so that they satisfy, to a certain significant degree, the student’s goals and aptitudes.

It is noted that the student should choose **three** restricted courses from her/his direction (from the list found in this guide) and *any* **two** courses in Computer Science direction of 3 or 4 level. (These two courses are: Core courses or other courses from the restricted list or restricted courses of the student’s direction.) Graduate courses offered by the Department of Computer Science may also be selected by the student as Restricted Choices for her/his direction after appropriate approval from the Committee of the Undergraduate Program.

lective Courses

Each student, in consultation with her/his Academic Advisor, selects elective courses. The selection is made according to the interests and the goals of the student. The Unrestricted Choices must be courses offered by other Departments. According to the Rules for Undergraduate Studies of the University, the elective courses must include courses *from at least three* different Schools of the University.

Foreign Language Courses

Each student must successfully attend two courses in a foreign language. The Department has allocated 10 ECTS credits for these courses and identifies English as a foreign language.

Diploma Project

During the last two semesters of her/his studies, each student undertakes an individual Diploma Project in accordance with regulations approved by the Departmental Board (Meeting of 19/7/95). These regulations have been revised by the Departmental Board (Meeting of 06/12/10) and can be found in Appendix A. The Diploma Projects are collected within the open access Digital Library (http://godigital.cs.ucy.ac.cy).

Minor Program in Computer Science

The Minor Program in Computer Science is open to all students of the University outside the Department of Computer Science. It consists of eight courses with a total workload of at least 60 credits. Given that schooling starts in Spring, the program can be completed in four consecutive semesters for monitoring two courses per semester. The organization of courses is as follows:

1st Semester (considered as a Fall Semester)
CS131 Programming Principles
CS121 Digital Systems

2nd Semester
CS132 Programming Principles II
CS221: Computer Organization and Assembly

3rd Semester
CS231 Data Structures and Algorithms
CS241 Systems Analysis & Design

4th Semester
Two courses from core courses or from the restrictive elective courses of the Computer Science curriculum.

The number of admitted students in the Minor Program in Computer Science is 10. Necessary criterion for admission is that the weighted average for all previous semesters must be at least 6.5.

Table 1: Dependencies among courses

Code	Course	Prerequisites / Conditions
CS111	Discrete Structures in Computer Science and Computation	
CS121	Digital Systems	
CS131	Programming Principles I	
CS132	Programming Principles II	CS131- Programming Principles I
CS202	Explorations into Computer Science	
CS211	Theory of Computation and Complexity	CS111- Discrete Structures in Computer Science and Computation MAS012- Calculus for Computer Scientists I
CS221	Computer Organization and Assembly Language Programming	CS121- Digital Systems CS131- Programming Principles I
CS222	Operating Systems	CS221- Computer Organization and Assembly Language Programming
CS231	Data Structures and Algorithms	CS111- Discrete Structures in Computer Science and Computation CS132- Programming Principles II
CS233	Object-Oriented Programming	CS231- Data Structures and Algorithms
CS241	Systems Analysis and Design	
CS323	Theory and Practice of Compilers	CS211- Theory of Computation and Complexity CS231- Data Structures and Algorithms
CS324	Communications and Networks	CS131- Programming Principles I
CS336	Algorithms and Complexity	CS231- Data Structures and Algorithms
CS341	Artificial Intelligence	CS231- Data Structures and Algorithms
CS342	Databases	CS231- Data Structures and Algorithms
CS361	Software Engineering I	CS132 - Programming Principles II CS241- Systems Analysis and Design
CS362	Software Engineering II	CS361 – Software Engineering I
CS363	Software Engineering Professional Practice	CS361- Software Engineering I
CS370	Computer Architecture	CS222- Operating Systems
CS371	System Programming	CS222-Operating Systems CS231- Data Structures and Algorithms
CS372	Parallel Processing	CS222- Operating Systems
CS375	Advanced Networks	CS324- Communications and Networks
CS400 – CS401	Diploma Project	Approval by Academic Advisor
CS412	Logic in Computer Science	CS111- Discrete Structures in Computer Science and Computation
CS413	Computational Geometry	CS231- Data Structures and Algorithms
CS424	Digital Signal Processing	CS111- Discrete Structures in Computer Science and Computation MAS016- Linear Algebra for Computer Science MAS012- Calculus for Computer Scientists
CS425	Internet Technologies	CS233- Object-Oriented Programming CS324- Communications and Networks
CS426	Computer Graphics	CS132- Programming Principles II
CS431	Synthesis of Parallel Algorithms	CS231- Data Structures and Algorithms
CS432	Distributed Algorithms	CS211- Theory of Computation and Complexity CS231- Data Structures and Algorithms
CS433	Constraint Programming and Satisfaction	CS111- Discrete Structures in Computer

		Science and Computation CS231- Data Structures and Algorithms
CS434	Logic Programming and Artificial Intelligence	CS111- Discrete Structures in Computer Science and Computation
CS435	Human - Computer Interaction	
CS442	Computational Learning Systems	CS231- Data Structures and Algorithms
CS444	Computational Intelligence System	CS442- Computational Learning Systems
CS445	Digital Image Processing	CS231- Data Structures and Algorithms MAS016- Linear Algebra for Computer Science
CS446	Advanced Topics in Databases	CS342- Databases
CS447	Computer Vision	CS231- Data Structures and Algorithms MAS016 Linear Algebra for Computer Science
CS451	Data Mining on the Web	CS231- Data Structures and Algorithms CS342- Databases
CS461	Software Validation, Verification and Quality	CS361- Software Engineering I
CS462	Software Analysis, Modelling and Design	CS233- Object-Oriented Programming CS361- Software Engineering I
CS463	Software Reuse	CS233- Object-Oriented Programming CS361- Software Engineering I
CS470	Design of Embedded Systems	CS221- Computer Organization and Assembly Language Programming
CS475	Network and Information Security	CS324- Communications and Networks
CS476	Wireless Networks	CS324- Communications and Networks
MAS013	Calculus for Computer Science II	MAS012- Calculus for Computer Science I

Short Description of Courses

Each course description includes the name of the instructor that offers the course in the current academic year 2013-2014. For the restricted courses that are not offered in this academic year, the name of the instructor who suggested the course is included instead. The language of instruction for all courses is **Greek**.

CS111 Discrete Structures in Computer Science and Computation
Type: Compulsory for all direcitons
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: A. Kakas
Objectives: Introduction to basic mathematical concepts that are applicable to Computer Science. Development of mathematical way of thinking about problems. Provision of some necessary mathematical tools.
Content: Foundations: sets and functions. Logic: Propositional Logic: basics of Predicate Logic. Mathematical Reasoning: methods of proof, induction. Counting: basics of counting, pigeonhole principle, permutations and combinations. Relations: properties and applications, equivalence relations, partial orders. Graphs: basic concepts.
Prerequisites: ----------
Bibliography:

1. K. Rosen, *Discrete Mathematics and its Applications*, 5th Edition, McGraw-Hill, 2003.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam and homework.

CS121 Digital Systems
Type: Compulsory for all direcitons
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: C. Pattichis
Objectives: Introduce various representations of information in a digital computing system. Presentation of the building blocks of digital computing systems. Teach the basic methods for the design and analysis of digital systems. Practical implementation of a simple digital computing system.
Content: Principles of design and construction of digital electronic systems and computers. Representation of data with binary sequences. Data storage and processing by electronic digital circuits. Consolidation of theoretical knowledge through practical exercises in the design and construction of digital circuits in the laboratory for Digital Systems Design and Microprocessors.
Prerequisites: -------------
Bibliography:

1. M. Mano and D.Kime, *Logic and Computer Design Fundamentals and Xilinx 4.2 Package*, 3rd Edition, Prentice Hall, 2003.

Teaching Methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, quizzes and homework (final project and exercises).

CS131 Programming Principles I
Type: Compulsory for all direcitons
Level: Undergraduate
Semester: Fall and Spring
Credit: 7,5 ECTS units
Instructor: Y. Sazeides (fall) / …………….. (spring)

Objectives: Introduction of methods for problem-solving through programming. Development of procedural and object-oriented problem solving skills and algorithmic thinking. Provision of deep understanding of basic programming principles and algorithmic techniques, design, implementation, testing and debugging of modular programs. Understanding the important concepts of program abstraction and data abstraction. Mastering of a high-level programming language (Java).
Content: Presentation of the software development process and introduction to the basic principles of programming and program design using the Java language. Global overview of the Java language with emphasis on built-in and abstract data types, control structures, functions, modular programming and recursion.
Prerequisites: -------------
Bibliography:

1. R. Sedgewick and K. Wayne, *Introduction to Programming in Java: An Interdisciplinary Approach*, Addison Wesley, 2008.

Teaching methods: Lectures (3 hours weekly), Laboratory sessions (4 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam, homework (programming assignments) and quizzes.

CS132 Programming Principles II

Type: Compulsory for all direcitons
Level: Undergraduate
Semester: Fall and Spring
Credit: 7,5 ECTS units
Instructor: P. Trancoso (Fall) / D. Zeinalipour (Spring)
Objectives: Provide an in-depth understanding of programming principles underlying modern application and systems software. The course familiarizes the students with advanced programming constructs utilized for handling memory and files, basic software structures and their associated algorithms, low-level programming, building, debugging, documenting and optimizing large-scale software systems individually and in groups through integrated software environments. The course is taught in the C programming language.
Content: i) Introduction to C for Programmers: types x86/x64, loops, selections, expressions, arrays, functions, IO, basic program organization, ii) Advanced C programming constructs: program anatomy and processes, memory and addresses (pointers, pointers and arrays, strings and examples), structures, unions and enumerations. Linear and non-linear programming data structures (dynamic memory allocation, lists, queues, doubly-linked lists, trees, applications and examples). iii) Advanced Compilation Topics and Tools: preprocessor directives, compiling multiple files with makefiles, static (.a) and dynamic (.so) linking of object files (.o), error handling (assert.h), static and dynamic code analysis (valgrind and gprof), iv) Methodology for Large-scale Program Development: the eclipse IDE, modules, information hiding, design issues, version management with SVN, introduction to software engineering (unit testing, programming in groups, open source development), v) Selected Topic: low-level programming (binary operators and examples, binary files and hexdump).
Prerequisites: CS131
Bibliography:

1. K. N. King, *C Programming: A Modern Approach*, 2nd Edition, W. W. Norton & Company, 2008.
2. (in Greek) Νίκος Χατζηγιαννάκης, *Η Γλώσσα C σε Βάθος*, 3rd Edition, Κλειδάριθμος, 2009.

Teaching methods: Lectures (3 hours weekly), Recitation (1.5 hours weekly) and Laboratory sessions (4 hours weekly).
Assessment: Final exam, midterm exam, programming assignments and final group project.

CS202 Explorations into Computer Science

Type: Compulsory for all direcitons
Level: Undergraduate

Semester: Fall
Credit: 3 ECTS units
Instructor: M. Dikaiakos
Objectives: Introduction to topics that compose a global picture of Computer Science. Creation of enthusiasm and interest in Computer Science. Update about current developments in Computer Science. Familiarization with practical applications of Computer Science.
Content: Weekly lecturers/seminars that cover a broad spectrum of Computer Science and its basic areas, starting from its birth and reaching its modern evolutions. Revolutionary ideas for the foundation and development of Computer Science.
Prerequisites: ----------
Bibliography: ----------
Teaching methods: Lectures / Seminars (1.5 hours weekly).
Assessment: Group project and presentation, class participation and attendance.

CS211 Theory of Computation and Complexity
Type: Compulsory for all direcitons
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: A. Philippou
Objectives: Introduction to foundational concepts of the Theory of Computation. Development and cultivation of formal and syllogistic reasoning. Familiarization with fundamental techniques for proofs and mathematical reasoning. Realization of the limitations on the capabilities of computers.
Content: Formal methods of computation based on machines, grammars and languages: finite automata vs. regular languages; pushdown automata vs. context-free grammars; Turing machines vs. unrestricted grammars. Models of computation equivalent to Turing machines and Church’s Thesis. Computability and Uncomputability. Introduction to Theory of Computational Complexity with emphasis on the Theory of NP-completeness.
Prerequisites: CS111, MAS012
Bibliography:

1. M. Sipser, *Εισαγωγή στη Θεωρία Υπολογισμού*, Πανεπιστημιακές Εκδόσεις Κρήτης, 2007
2. M. Mavronicolas, *Θεωρία Υπολογισμού*, προσχέδιο βιβλίου, Αύγουστος 2005.
3. H. R. Lewis and C. H. Papadimitriou, *Elements of the Theory of Computation*, 2nd Edition, Prentice Hall, 1998.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam and homework.

CS221 Computer Organization and Assembly Language Programming
Type: Compulsory for all direcitons
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: P. Trancoso
Objectives: Introduction to the basic concepts and methods of computer organization. Present trends of various computer technology parameters. Introduction of the concept of instruction set architecture. Teach different instruction sets and assembly programming. Design of a pipelined datapath. Learn basic principles of memory hierarchy. Implement a simple computing system. Practical experience in the Digital Logic and Microprocessors Laboratory.
Content: Introduction to computer organization and architecture. Types of instructions, coding of instructions, Arithmetic and Logic Unit. Basic principles of the organization of the main functional units of a computer system at machine level: Central Processing Unit (CPU), memory, and Input/Output. Interfacing CPU and peripheral units. Programming in assembly language for MIPS R2000/R3000 and Intel Pentium.
Prerequisites: CS121, CS131

Bibliography:

1. D. A. Patterson, J. L. Hennessy, *Computer Organization and Design - The Hardware/Software Interface*, Third Edition, Morgan Kaufmann, 2005.

Teaching Methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).

Assessment: Final and midterm exams, quizzes and homework (final project and exercises).

CS222 Operating Systems

Type: Compulsory for all direcitons

Level: Undergraduate

Semester: Spring

Credit: 7,5 ECTS units

Instructor: G. Papadopoulos

Objectives: Introduction to the basic principles of design and operation of modern operating systems. Familiarization with the various operation levels and mechanisms, case studies involving typical operating systems like UNIX and Windows as well as the dual role of the operating system, as manager of the various parts of the computer hardware and supplier of offered services to the user.

Content: Introduction, history and evolution of operating systems. General structure, operations and characteristics of an operating system. Concurrency. Process management. Scheduling and dispatch. Real and virtual memory management. I/O management and disk scheduling. File management. Protection, security and reliability.

Prerequisites: CS221

Bibliography:

1. W. Stallings, *Operating Systems: Internals and Design Principles*, 6th Edition, Prentice Hall, 2009.

Teaching methods: Lectures (3 hours weekly) and tutorial sessions (1.5 hours weekly).

Assessment: Final exam, midterm exam and homework (theoretical and programming assignments).

CS231 Data Structures and Algorithms

Type: Compulsory for all direcitons

Level: Undergraduate

Semester: Fall and Spring

Credit: 7,5 ECTS units

Instructor: G. Pallis (Fall) / ……………. (Spring)

Objectives: Familiarization with data structures and the algorithms manipulating them. Appreciation of the importance of careful organization of information for efficient searching and modification. Acquaintance with techniques for the analysis of algorithm efficiency. Developments of capabilities for designing algorithms to minimize their execution time and space requirements.

Content: Study of data structures for the organization and efficient processing of data. Linear and non-linear data structures. Hashing techniques. Issues of memory management. Sorting Algorithms. Graph Algorithms. Introduction to algorithm design techniques. Analysis of the average and worst-case complexity of algorithms.

Prerequisites: CS111, CS132

Bibliography:

1. M. A. Weiss, *Data Structures and Algorithms in Java*, Pearson, 2012.
2. M. Goodrich, R. Tamassia. *Data Structures and Algorithms in Java*, Wiley, 2011.

Teaching methods: Lectures (3 hours weekly), Laboratory sessions (2 hours weekly) and Recitation (1 hour weekly).

Assessment: Final exam, midterm exam and homework (theoretical and programming assignments).

CS233 Object-Oriented Programming

Type: Compulsory for all direcitons

Level: Undergraduate

Semester: Spring

Credit: 7,5 ECTS units
Instructor: G. Kapitsaki
Objectives: Familiarization with concepts of Object-Oriented Programming and the use of Object-Oriented Methodology for the solution of computational problems. Familiarization with advanced programming techniques and application of this with the use of the Java programming language.
Content: Development of object-oriented way of thinking and capabilities to apply it to solving complex problems. Problem-solving and programming using object-oriented methodologies. Abstraction and Information Hiding. Libraries and reuse. Object-oriented design. Inheritance. Polymorphism. Interfaces. Inner classes in Java. Exceptions. Input/Output. Threads and Concurrency in Java. Collections. Advanced topics: Annotaions, Networking, Serialization.
Prerequisites: CS132, CS231
Bibliography:

1. Brunel Eckel, *Thinking in Java*, 3rd Edition, Prentice Hall, 2000.
2. B. Kernighan and R. Pike, *The Practice of Programming*, Addison Wesley, 1999.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, homework (programming assignments) and quizzes.

CS241 Systems Analysis and Design
Type: Compulsory for all direcitons
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: Ch. Schizas
Objectives: Introduction to Systems Theory and its use for the analysis, design and development of an Information Technology (IT) system. Familiarization with the procedure followed for the analysis and the role of each stakeholder related to the IT system in being developed. Acquiring relevant practical experience.
Content: Study of the theory and the methodologies which have been developed over the years in the area of systems, with the objective to introduce techniques and methodologies for systems analysis and design of Information Systems. Special attention to the study of "Information Society" and its effect in system development and maintenance.
Prerequisites: -----------
Bibliography:

1. J. Whitten, L. Bentley and K. Dittman, *Systems Analysis and Design Methods*, 7th Edition, McGraw-Hill, 2007.
2. K. Kendall, J. Kendall, *Systems Analysis and Design*, Pearson Education, 2005.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory exercises and group project).

CS323 Theory and Practice of Compilers
Type: Compulsory for GD
Restricted Choice for CSN and SE
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: P. Trancoso
Objectives: Introduction to advanced techniques of compiler design and implementation.
Content: Fundamental principles of compiler design. Relation of translators to formal languages and automata theory. Lexical, syntactic and semantic analysis, code generation and optimization, etc. Practical exercises using lex and yacc.
Prerequisites: CS211, CS231
Bibliography:

1. V. Aho, R. Sethi and J. D. Ullman, *Compilers – Principles, Techniques, and Tools*, Addison-Wesley, 1986.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (theoretical and diagnostic assignments and semester project).

CS324 Communications and Networks

Type: Compulsory for all directions
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: V. Vasiliou
Objectives: Familiarization with fundamental topics in communication networks, with a focus on the Internet.
Content: Fundamental issues in Communication Networks, using the Internet as an example. Networking layers, such as the application, transport, network, link and physical layers. Open systems and internetworking. Networking technologies including wired and wireless Local Area Networks (LANs, WLANs) and network topologies. Algorithms, including routing and congestion control, protocols and standards, quality of service (QoS) and applications. Introduction to more advanced topics, such as real time services and network security.
Prerequisites: CS131
Bibliography:

1. J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, Sixth Edition, Addison-Wesley, 2012.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (2 hours weekly).
Assessment: Final exam, midterm exam and homework (including laboratory exercises).

CS336 Algorithms and Complexity

Type: Compulsory for GD
Restricted Choice for SE
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: Ch. Georgiou
Objectives: Familiarization with fundamental techniques of designing and analyzing algorithms. Familiarization with significant algorithms in various fields that have been suggested in the literature. Familiarization with techniques for implementing and empirically evaluating algorithms.
Content: Topics in the design and analysis of efficient algorithms and their complexity. Significant algorithms in Graph Theory, Algebra, Geometry, Number Theory and Combinatorics. General algorithmic techniques (e.g., divide-and-conquer, backtracking, dynamic programming). Randomized, Parameterized and Approximation algorithms. Fast Fourier Transform. Inherent lower bounds on problem complexity.
Prerequisites: CS231
Bibliography:

1. J. Kleinberg and E. Tardos, *Σχεδίαση Αλγορίθμων*, Εκδόσεις Κλειδάριθμος, 2008. (English Version: J. Kleinberg and E. Tardos, *Algorithm Design*, Addison Wesley, 2006).
2. S. Dasgupta, C. Papadimitriou and U. Vazirani, *Αλγόριθμοι*, Εκδόσεις Κλειδάριθμος, 2009.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam and homework (theoretical and programming assignments).

CS341 Artificial Intelligence

Type: Compulsory for GD

Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: ……………….
Objectives: Artificial Intelligence (AI) covers a broad spectrum of areas and this gives it an interdisciplinary character. The expert systems technology constitutes an important component of the applicative part of AI. The principal objectives of the course are (a) to introduce students to central areas of AI, that are directly relevant to the technology of expert systems, namely problem solving through the use of search and heiristics, and knowledge representation, and (b) to introduce students to the expert systems technology, focusing on first and second generation architectures and knowledge engineering methodologies.
Content: Problem solving techniques in Artificial Intelligence. Knowledge Representation Formalisms (logic, associative networks, frames, production rules). Expert Systems Technology. First and Second Generation Architectures for Expert Systems. Knowledge Engineering. Intelligent Agents. Multi-Agent Systems.
Prerequisites: CS231
Bibliography:

1. E. Keravnou, *Τεχνητή Νοημοσύνη και Έμπειρα Συστήματα*, Ελληνικό Ανοικτό Πανεπιστήμιο, 2000.
2. G.F. Luger and W.A. Stubblefield, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, 5th Edition, Addison-Wesley, 2005.
3. G. Weiss (ed), *Multiagent Systems: a modern approach to distributed AI*, The MIT Press, 1999.
4. P. Jackson, *Introduction to Expert Systems*, 3rd Edition, Addison-Wesley, 1999.
5. E. Rich and K. Knight, *Artificial Intelligence*, 2nd Edition, McGraw-Hill, 1991.
6. S. Russel and P. Norviq, *Artifitial Intelligence: A Modern Approach*, 2nd Edition, Prentice Hall, 2002.

Teaching methods: Lectures (3 hours weekly) and Tutorials (1 hour weekly).
Assessment: Final exam, midterm exam and homework (theoretical and programming assignments).

CS342 Databases

Type: Compulsory for all directions
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: P. Andreou
Objectives: Introduction to the basic principles needed for the design and the use of ta database. Provision of practical exercises in the application of these concepts with the use of an industrial system for database.
Content: Introduction to Databases. Organization and proper management of large quantities of data for use in applications. Database models such as the entity-relation model, the relational model, the network model and the hierarchical model.
Prerequisites: CS231
Bibliography:

1. R. Elmasri and S. Navathe, *Fundamentals of Database Systems*, Fifth Edition, Addison-Wesley, 2007.
2. R. Elmasri and S. Navathe, *Θεμελιώδεις Αρχές Συστημάτων Βάσεων Δεδομένων*, 5th Edition, 1ος τόμος, Εκδόσεις Δίαυλος, 2007.
3. R. Ramakrishnan and J. Gehrke, *Database Management Systems*, 3rd Edition, McGraw-Hill, 2003.

Teaching methods: Lectures (3 hours weekly), Recitation (1.5 hour weekly) and Laboratory sessions (1 hour weekly).
Assessment: Final exam, midterm exam and homework (theoretical assignments, final project).

CS361 Software Engineering I

Type: Compulsory for GD and SE
Restricted Choice for CSN

Level: Undergraduate

Semester: Winter

Credit: 7,5 ECTS units

Instructor: G. Kapitsaki

Objectives: Familiarization with and assimilation of the approaches, methodologies, models and tools used to develop quality software systems. Understanding of software architectures, software modeling and testing. Understanding of software testing process and of software architecture design patterns. Applications of software creation methodologies on the construction of a real software system. Understanding of software testing procedures. Use and application of architecture design patterns.

Content: Methods, tools, and procedures for the development and maintenance of large-scale software systems. Existing life-cycle models (e.g. waterfall model). Introduction to Agile development. Requirements analysis and specification techniques. Software development methodologies. Unified Modelling Language (UML) and supported static and dynamic diagrams. Code transformation. Practical experience with CASE tools for modeling data and procedures (ArgoUML, StarUML). Prototyping for Web applications (HTML, CSS). Archıtectural Design patterns (Model View Controller etc.). Software verification and validation. Unit testing and frameworks (JUnit etc.). CASE tools. Project planning and management.

Prerequisites: CS132, CS241

Bibliography:

1. H. Van Vliet, *Software Engineering: Principles and Practice*, 3rd Edition, John Wiley & Sons, 2008.
2. R. Pressman, *Software Engineering: A Practitioner's Approach*, 7th Edition, Mc Graw Hill, 2010.
3. I. Sommerville, *Software Engineering*, 9th Edition, Addison-Wesley, 2011.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1,5 hour weekly).

Assessment: Final exam, midterm exam, homework (project) and lab exercises.

CS362 Software Engineering II

Type: Compulsory for SE
Restricted Choice for GD

Level: Undergraduate

Semester: Spring

Credit: 7,5 ECTS units

Instructor: G. Papadopoulos

Objectives: Familiarization and understanding of advanced principles, concepts and practices of software engineering. A number of contemporary areas of software engineering will be covered. This course further serves as a “roadmap” for advanced electives and graduate courses in software engineering.

Content: Agile software development. Model driven engineering. Legacy systems. Sociotechnical systems. Software reuse. Component Based systems. Distributed software engineering. Service oriented architectures. Embedded software. Aspect oriented software engineering.

Prerequisites: CS241, CS361

Bibliography:

1. I. Sommerville, *Software Engineering*, 9th Edition, Addison-Wesley, 2011.
2. H. van Vliet, *Software Engineering: Principles and Practice*, 3rd Edition, John Wiley & Sons, 2008.
3. E. J. Baude, M. E. Bernstein, *Software Engineering – Modern approaches*, John Wiley & Sons, 2nd edition, 2011.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).

Assessment: Final exam, midterm exam, homework (theoretical problems – study / analysis of a software system) and lab exercise.

CS363 Professional Practice in Software Engineering

Type: Compulsory for SE
Restricted Choice for GD

Level: Undergraduate

Semester: Spring

Credit: 7,5 ECTS units

Instructor: G. Kapitsaki

Objectives: Embedding and practical application of the theoretical approaches and methodologies of Software Engineering for the development of a product-software system that serves the needs of an organization belonging to the local market. Practical use of processes and tools, such as central and distributed version control systems (SVN and GitHub), testing at different levels, group communication, professionalism, and ethical conduct.

Content: Undertake and carrying out to completion a significant software project by small student groups (of about 2-6 students each). All phases in the development of software. Some of the specific projects come from the industrial sector. Version control systems (SVN and GitHub). Testing. Software system analysis through software metrics. Speciaized issues depending on the project nature (e.g., deployment on web servers, GUI tools and frameworks etc.).

Prerequisites: CS361

Bibliography:

1. Selected articles

Teaching methods: Students are grouped in teams of 2-6 persons. Meetings/ discussions are held regularly (weekly per team). Laboratory sessions (1.5 hours weekly).

Assessment: Assessment of the product-software system, assessment of the corresponding documentation, oral presentation and exam.

CS370 Computer Architecture

Type: Compulsory for CSN

Level: Undergraduate

Semester: Fall

Credit: 7,5 ECTS units

Instructor: P. Evripidou

Objectives: Introduction to current methodology for performance evaluation and comparison of computer systems. Presentation of basic and advanced concepts in the organization of modern microprocessors. Examination of current trends in the computer architecture area.

Content: Introduction to the state-of-the-art uniprocessor, high performance computer architecture. Emphasis on quantitative analysis and cost/performance trade-offs in the design of the basic units of a processor: instruction set, pipelining, memory system and input/output systems. Qualitative analysis of real machines and their performance data.

Prerequisites: CS222

Bibliography:

2. J. Henessy and D. Patterson, *Computer Architecture: A Quantitative Approach.* 3rd Edition. Morgan Kaufmann, 2002.
3. Selected research articles

Teaching Methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).

Assessment: Final exam, midterm exam and homework (project and exercises).

CS371 System Programming

Type: Compulsory for CSN and SE
Restricted choice for GD

Level: Undergraduate

Semester: Spring

Credit: 7,5 ECTS units

Instructor: D. Zeinalipour

Objectives: The main objective of this undergraduate course is to allow students develop

complex system-level software in the C programming language while gaining an intimate understanding of the UNIX operating system (and all OS that belong to this family, such as Linux, the BSDs, and even Mac OS X). Topics covered will include the user/kernel interface, fundamental concepts of UNIX, user authentication, basic and advanced I/O, fileystems, signals, process relationships, and interprocess communication. Fundamental concepts of software development and maintenance on UNIX systems will also be covered. The students are expected to have a good working knowledge of the C programming language and a good working knowledge of fundamental Operating System Concepts.
Contents: Main concepts of System Programming, Introductory and Advanced UNIX commands, System utilities and stream editors (awk,sed), Advanced Shell programming with an emphasis on Bash, Low-Level I/O in C, Files and Filesystem, Processes: Environment, Control and Signals, Interprocess Communication (IPC) with an emphasis on Pipes and Named Pipes (FIFO) in C, XSI IPC (Semaphores, Shared Memory and Message Queues) in C, Network IPC (TCP Sockets) and the client/server model in C, Multithreading in C, Performance evaluation (profiling). Issues in system security and system engineering, Systems Programming in Windows (threads, processes, IPC, sockets and Powershell programming), Scripting Languages: Perl, PHP, Python, TCL/TK.
Prerequisites: CS222, CS231
Bibliography:

1. R. Stevens, S. A. Rago, *Advanced Programming in the UNIX Environment*, 3rd Edition, Addison-Wesley, 2013.
2. S. Das, *Your UNIX/Linux: The Ultimate Guide*, 3rd Edition, McGraw Hill, 2013.

Teaching Methods: Lectures (3 hours weekly), and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, programming assignments and presentation.

CS372 Parallel Processing

Type: Compulsory for CSN
Restricted choice for GD and SE
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: ………………
Objectives: Introduction to the fundamental concepts of Parallel Processing with emphasis on parallel machines and parallel programming.
Content: The entire spectrum of parallel machines as appearing in Flynn's classification: SISD, SIMD, MISD, MIMD. The main approaches for design and operation of multiprocessor systems. Conventional and non-conventional machines (Data-flow and reduction). Parallel programming approaches: (1) Automatic-parallelizing compilers, (2) Extending serial languages with parallelizing constructs, (3) parallel languages for Functional Programming. Special emphasis on parallel architectures and parallel programming.
Prerequisites: CS222
Bibliography:

1. D. E. Culler and J. P. Singh, *Parallel Computer Architecture – A Hardware/ Software Approach*, Morgan Keufmann, 1999.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam, homework and final project.

CS375 Advanced Networks

Type: Compulsory for CSN
Restricted choice for GD
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: V. Vassiliou
Objectives: Extension of the basic knowledge about Computer Networks regarding

architectures, techniques and protocols Multimedia and Multiservice Networks. Familiarization with the state-of-the-art in Computer Networking, and the main research activities in the area of High-Speed Multimedia and Multiservice Networks. Content: Introduction to computer networks and the Internet. The IPv6 Internet protocol. Multicast routing. The TCP protocol. Congestion control. Performance analysis. Multimedia networking applications. Real-time services and protocols. Quality of Service (QoS). QoS routing. MPLS Traffic engineering. Mobile and wireless networks. Issues in security for computer networks.
Content: Introduction to Computer Networks and the Internet, Network Layer - IPv6, Routing - Multicast Routing, TCP, Congestion Control, Performance Analysis, Multimedia Networking Applications. Realtime services and protocols, Quality of Service, MPLS, Traffic Engineering, QoS Routing, Mobile and Wireless Networks, Issues in Security for Computer Networks.
Prerequisites: CS324
Bibliography:

1. J. F. Kurose and K. W. Ross, *Computer Networking – A Top Down Approach Featuring the Internet*, 6th Edition, Addison-Wesley, 2012.
2. Y.-D. Lin, T.-H. Hwang, and F. Baker , *Computer Networks, An Open Source Approach* , McGraw Hill, 2011

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (2 hours weekly).
Assessment: Final exam, midterm exam and homework (including laboratory exercises).

CS412 Logic in Computer Sciences
Type: Restricted choice for GD
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: A. Philippou
Objectives: The main objective of the course is to prepare students for using logic as a formal tool in Computer Science. Furthermore, it aims to develop and cultivate formal and syllogistic reasoning and provide a thorough introduction to computational logic and its applications in Computer Science. Content: Propositional Logic Syntax, Semantics, Normal Forms, Decision Procedures, Proof Theory, Compactness and Resolution. Predicate Logic: Syntax and Semantics, Proof Theory Soundness and Completeness and Resolution. Logic Programming. Programming Language Semantics and Verification. Linear and Branching Temporal Logics: Syntax, Semantics, and model-checking algorithms.
Prerequisites: CS111
Bibliography:

1. M. Huth and A. Ryan, *Logic in Computer Science: Modeling and Reasoning about Concurrent Systems*, Cambridge University Press, 2000.
2. M. Ben-Ari, *Mathematical Logic for Computer Science*, Springer-Verlag, 2nd Edition, 2003.
3. U. Schoning, *Logic for Computer Scientists*, Springer-Verlag, 2nd Printing, 2008.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam and homework.

CS413 Computational Geometry
Type: Restricted choice for GD
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: A. Philippou
Objectives: -----------
Content: Convex hull in the plane: basic properties and algorithms from computing it. Arrangements of line segments: computing intersections, representing subdivisions,

computing an overlay. Arrangements of lines: incremental construction algorithm. The Art Gallery Problem: partition into monotone polygons, monotone polygon triangulation, simple polygon triangulation. Geometric searching: kd-trees, range trees, fractional cascading. Point location: basic properties of the trapezoid map, algorithm for constructing map and search structure, point location algorithm. Voronoi diagram: basic properties and algorithms for computing it. Delaunay triangulation: basic properties, incremental algorithm.
Prerequisites: CS231
Bibliography:

1. M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, 2nd Edition, Springer, 2000.
2. Γ. Ζ. Εμίρης, *Υπολογιστική Γεωμετρία: Μία σύγχρονη αλγοριθμική προσέγγιση*, Κλειδάριθμος, 2009.

Teaching methods: Lectures (3 hours weekly) and Laboratory (1 hour weekly).
Assessment: Final exam, midterm exam and homework (theoretical and laboratory exercises).

CS424 Digital Signal Processing

Type: Restricted choice for CSN
Level: Undergraduate
Semester: …………
Credit: 7,5 ECTS units
Instructor: A. Pitsillides
Objectives: Introduction to Digital Signal Processing (DSP) methods and applications. Content: Discrete signals and systems, sampling of signals, frequency analysis of discrete systems and signals, z-transform, Fourier-Transform, Discrete Fourier Transform, and Fast Fourier Transform, digital filters, application examples.
Prerequisites: CS111, MAS016, MAS017
Bibliography:

1. S. Theodorides, *Digital Signal Processing*, Univeristy of Patras Publishing, 1992.
2. J. H. McClellan, R. W. Schafer and M. A. Yoder, *DSP First*, Prentice Hall, 1998.
3. The Student Edition of Matlab: *User's, Guide*, Prentice Hall, 2005.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory exercises, additional exercises, final project).

CS425 Internet Technologies

Type: Restricted choice for all directions
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: M. Dikaiakos
Objectives: Introduction to Internet and the World-Wide Web Technologies. Emphasis in given on programming of Internet Systems and Services.
Content: Topics of Internet and World-Wide Web technologies, with an emphasis on WWW applications and Internet programming. The foundations of WWW applications including hypertext, navigation in hyperspace, hypertext usability, information overload, markup languages and methodologies of WWW application design. System issues related to Internet programming and performance: protocols, servers, WWW interactivity, Internet-based distributed systems.
Prerequisites: CS233, CS324
Bibliography:

1. A. Moller and M. Schwartzbach, *An Introduction to XML and Web Technologies*, Addison Wesley, 2006.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, homework (weekly assignments) and class participation.

CS426 Computer Graphics

Type: Restricted choice for GD
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: Y. Chrysanthou / E. Stavrakis
Objectives: Introduction to the basic principles of digital image synthesis. Explain how a 3-dimentional virtual world is defined starting from the geometry, the materials, the lights and cameras and how the 2-dimentional resulting image is produced by going through the graphics pipeline. Provision of both the theoretical foundations as well as practical skills through the use of industry standards, such as OpenGL or DirectX.
Content: Scene construction, scene hierarchies, camera specification, projections of primitives, clipping, visible surface determination, polygon rasterisation (z-buffer), texture mapping, local and global illumination, shadows, ray tracing, radiosity, real-time acceleration techniques.
Prerequisites: CS132
Bibliography:

1. M. Slater, A. Steed and Y. Chrysanthou, *Computer Graphics and Virtual Environments: From Realism to Real-Time*, Addison-Wesley, 2001.
2. P. Shirley, M. Ashikhmin and S. Marschner, *Fundamentals of Computer Graphics*, 3rd Revised Edition, 2009..

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS431 Synthesis of Parallel Algorithms

Type: Restricted choice for all direcitons
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: Ch. Georgiou
Objectives: Introduction to the fundamental techniques of parallel algorithm design and the use of these techniques in designing and analyzing parallel algorithms for basic problems. Introduction to the basic failure types appearing in parallel computing and the appropriate fault-tolerance techniques.
Content: Introduction to parallel computing. Complexity and efficiency measurements of parallel algorithms. Parallel computing models. Basic techniques for the design of parallel algorithms. Efficient parallel algorithms in Combinatorics, Graph Theory, and Matrix Theory. Complexity analysis of algorithms on the Parallel Random Access Machine (PRAM). Comparison between various models of computation. Advanced topics (fault-tolerance, atomicity, synchronization, computational limitations of PRAM).
Prerequisites: CS231
Bibliography:

1. J. Jaja, *An Introduction to Parallel Algorithms*, Addison-Wesley, 1992.
2. R. Miller and L. Boxer, Algorithms *Sequential & Parallel: A Unified Approach*, Prentice Hall, 2000.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam and homework (theoretical assignments).

CS432 Distributed Algorithms

Type: Restricted choice for all directions
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: N. Nikolaou
Objectives: Familiarization with the fundamental concepts of the Theory of Distributed Computing. Development of capabilities of designing, proving correct and analyzing distributed algorithms. Cultivation of syllogistic and mathematical approach to the field of

distributed algorithms.
Content: Formal models of distributed computing: shared memory versus message passing, determinism versus randomization, concepts of synchronism, asynchrony and real-time. Design and analysis of distributed algorithms and impossibility/improbability results for fundamental problems such as mutual exclusion, consensus, synchronization, leader election, construction of minimum spanning trees. Fault tolerance: Byzantine generals, wait-free algorithms, fault degrees. Formal methods for proving correctness of distributed algorithms. Advanced topics. Special emphasis throughout the course on lower and upper bounds on time and memory.
Prerequisites: CS211, CS231
Bibliography:

1. H. Attiya and J. L. Welch, *Distributed Computing: Fundamentals, Simulations and Advanced Topics*, Second Edition, John Wiley and Sons, 2003.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam, homework, participation in class and attendance.

CS433 Constraint Programming and Satisfaction

Type: Restricted choice for GD
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: Y. Dimopoulos
Objectives: A significant number of problems in Computer Science over a wide spectrum ranging from Computer Vision and Artificial Intelligence to the management of Computer Networks and Scheduling are special cases of Constraint Satisfaction problem. This course introduces the way to approach such problems and corresponding software. Students will be able to understand the structure and the behavior of Constraint Satisfaction problems and will get exposure to basic algorithms solving them. They will get experience over tools for Constraint Programming, the range of solvable problems and their applications to problem solving.
Content: Definition of constraint satisfaction problems. Constraint representation and complexity. Various forms of consistency. Backtracking and look-ahead techniques. Intelligent backtracking and condition for solution finding without backtracking. Heuristic and local methods for solution searching. Available commercial products. Study of problems from different application domains, their modeling and the complexity of various algorithms solving them.
Prerequisites: CS111, CS231
Bibliography:

1. R. Dechter, *Constraint Processing*, Morgan Keufmann, 2003.
2. E. Tsang, *Foundations of Constraint Satisfaction*, Academic Press, 1993.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam and homework (programming assignments).

CS434 Logic Programming and Artificial Intelligence

Type: Restricted choice for GD
Level: Undergraduate
Semester: ……………….
Credit: 7,5 ECTS units
Instructor: A. Kakas
Objectives: Familiarization with the basic concepts of Logic Programming and practical exercises in implementing them with the PROLOG language. Development of capabilities of applying Logic Programming to problems of Artificial Intelligence.
Content: Basic principles of Logic Programming and implementation using the language Prolog. Relation of Logic Programming to modern considerations regarding Artificial Intelligence. Solving application problems drawn from the fields of Artificial Intelligence and the Semantic Web, making use of Logic Programming and Constraint Logic Programming.
Prerequisites: CS111

Bibliography:

1. L. Sterling and E. Shapino, *The Art of Prolog*, 2nd Edition, The MIT Press, 1994.
2. M. Bramer, *Logic Programming with Prolog*, Springer, 2009.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1 hour weekly).
Assessment: Final exam, midterm exam and homework.

CS435 Human Computer Interaction

Type: Restricted choice
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: S. Loizou Kleanthous
Objectives: Appreciation of the importance of designing good user interfaces and the relation of user interface design and the way users interact with computers. Experience with applying a well-known methodology for designing interactive systems, starting from identifying user's needs, to applying usability evaluation methods.
Content: Analysis of the human as a computer system user (knowledge models, graphical animation, cognitive models). Interactive technologies (input-output devices, window environments, systems for collaborative support, virtual reality). Methodologies for the design of interactive systems.
Prerequisites: ----------
Bibliography:

1. N. Αβούρης, *Εισαγωγή στην επικοινωνία ανθρώπου-υπολογιστή*, Εκδόσεις ΔΙΑΥΛΟΣ, Αθήνα, 2000.
2. J. Preece, Y. Rogers and H. Sharp, I*nteraction Design: Beyond Human-Computer Interacrtion*, John Wiley and Sons, 2002.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS442 Computational Learning Systems

Type: Restricted Choice for GD
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: Ch. Christodoulou
Objectives: Familiarization with theoretical and practical issues involved in Computational Learning Systems. Study of machine learning methods as they have developed in recent years. Implementation and assessment of Computational Learning Systems.
Content: Introduction to Pattern Recognition, Multilayered Neural Networks and backpropagation learning algorithim, Recurrent Neural Networks, Reinforcement Learning, Hopfield Networks & Boltzmann Machines, Radial Basis Functions, Self-Organising Maps. Survey of the developments in artificial intelligence, machine learning, expert systems, cognitive science, robotics and artificial neural networks, which contributed to the development of the theory of learning systems.
Prerequisites: CS231
Bibliography:

1. C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
2. S. Haykin, *Neural Networks and Learning Machines*, 3rd Edition, Prentice Hall, 2009.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (lab exercises).

CS444 Computational Intelligent Systems

Type: Restricted Choice for GD
Level: Undergraduate

Semester: Spring
Credit: 7,5 ECTS units
Instructor: C.N. Schizas, Ch Christodoulou
Objectives: Global overview of Computational Intelligence and its applications in solving "real" problems in various disciplines such as decision making support, classification, prognosis and prediction, system optimization and recreational design. Moreover there will be an introduction to computational neuroscience/neuroinformatics as well as in cognitive science.
Content: Evolutionary Computing. Genetic Algorithms. Artificial Neural Networks. Fuzzy Systems. Artificial Life. Computational Neuroscience/Neuroinformatics; Hodgkin & Huxley and Integrate-and-Fire neuron models; Neural Coding; Hebbian Learning and Synaptic Plasticity; introduction to cognitive science. Development and Implementation of Computational Intelligence Systems.
Prerequisites: CS442
Bibliography:

1. P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley and Sons, 2nd edition 2007.
2. R. C. Eberhart and Y. Shi, *Computational Intelligence: Concepts to Implementations*, Elsevier, 2007.
3. E. R. Kandel, *Αναζητώντας τη Μνήμη*, (Μετάφραση Α. Καραμανίδης), Πανεπιστημιακές Εκδόσεις Κρήτης, 2008.
4. P. Dayan and L. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modelling of Neural Systems*, MIT Press, 2001.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory exercises, additional exercises, final project).

CS445 Digital Image Processing
Type: Restricted Choice for GD and CSN
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: M. Neophitou
Objectives: Introduction to the basic principles of Digital Image Processing: Digital Image and Video. Analysis and implementation of image and video processing and analysis algorithms and their application in industrial and biomedical systems. Content: Binary Image Representation. Image Histogram and Point Operations. Discrete Fourier Transform. Linear Image Filtering. Non Linear Image Filtering Pipeling. Image Compression. Image Analysis I. Image Analysis II. Digital Video Processing.
Prerequisites: CS132, MAS016
Bibliography:

1. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd Edition, Addison-Wesley, 2002.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory exercises, additional exercises, final project).

CS446 Advanced Topics in Databases
Type: Restricted Choice for GD and SE
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: ………………
Objectives: Familiarization with advanced topics in the design and management of Databases (and special kinds of those). Exposure to significant open problems and research directions in the field of Databases.

<u>*Content*</u>: Theoretical approach to logical and physical design of databases. Algorithms for logical and physical design of databases. Primary and secondary indexing techniques. Advanced query processing and query optimization. Query parallelism. Concurrency control and recovery, integrity and security of data. Distributed databases and introductory concepts distributed transaction processing involving multiple and heterogeneous databases. Problems of interfacing a database with software.
<u>*Prerequisites*</u>: CS342
<u>*Bibliography*</u>:

1. R. Ramakrishnan and J. Gehrke, *Database Management Systems*, 3rd Edition, McGraw-Hill, 2003.
2. R. Elmasri and S. Navathe, *Fundamentals of Database Systems*, 5th Edition, Addison-Wesley, 2007.

<u>*Teaching methods*</u>: Lectures (3 hours weekly) and Recitation/Laboratory sessions (1 hour weekly).
<u>*Assessment*</u>: Final exam, midterm exam, programming exercises and presentation.

CS447 Computer Vision

<u>*Type*</u>: Restricted Choice for GD
<u>*Level*</u>: Undergraduate
<u>*Semester*</u>: Fall
<u>*Credit*</u>: 7,5 ECTS units
<u>*Instructor*</u>: P. Kaimakis
<u>*Objectives*</u>: ---------
<u>*Content*</u>: Basic concepts and methodologies relating to the subject of Computer Vision. Image information, image processing, feature extraction. Image segmentation, clustering, multiple-image processing, case studies.
<u>*Prerequisites*</u>: CS231, MAS016
<u>*Bibliography*</u>:

1. D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice-Hall, 2003.
2. R. Hartley and A. Zeisserman, *Multiple View Geometry*, Cambridge University Press, 2003.
3. C. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag, 2007.
4. O. Faugeras and Q.T. Luong, *Geometry of Multiple Images*, MIT Press, 2001.
5. B. Horn, *Robot Vision*, MIT Press, Cambridge, Massachusetts, 1986.

<u>*Teaching methods*</u>: Lectures (3 hours weekly) and Laboratory sessions (1 hour weekly).
<u>*Assessment*</u>: Final exam, midterm exam, programming exercises and presentation.

CS451 Data Mining on the Web

<u>*Type:*</u> Restricted Choice for CSN and SE
<u>*Level:*</u> Undergraduate
<u>*Semester:*</u> Spring
<u>*Credit:*</u> 7,5 ECTS units
<u>*Instructor:*</u> G. Pallis
<u>*Objectives:*</u> Introduction to data mining, Clustering, Classification, Association Rules, Link Analysis, Web communities, Web Personalization.
<u>*Content:*</u> Data mining on the Web refers to the automatic discovery of interesting and useful patterns from the data associated with the usage, content, and the linkage structure of Web resources. It has quickly become one of the most popular areas in computing and information systems because of its direct applications in e-commerce, information retrieval/filtering, Web personalization, and recommender systems. The primary focus of this course is on examining techniques from data mining to extract useful knowledge from Web data. This course will be focused on a detailed overview of the data mining process and techniques, specifically those that are most relevant to Web mining. Several topics will be covered such as Map-Reduce framework, Web data clustering, classification, association rules, recommendation systems, link analysis, social networks and Web advertising.

Prerequisites: CS231, CS342

Bibliography:

1. J. Ham and M. Kamber, *Data Mining: Concepts and Techniques*, Second Edition, Morgan Kaufmann, 2006.
2. M. Χαλκίδη και M. Βαζιργιάννης, *Εξόρυξη Γνώσης από Βάσεις Δεδομένων και Παγκόσμιο Ιστό*, Εκδόσεις Gutenberg, 2005.
3. A. Rajaraman and J.D. Ullman, *Mining of Massive Data Sets*, Stanford University Press, 2010.
4. B. Liu, *Web Data Mining: Exploring Hyperlinks*, Contents and Usage Data, Springer, 2007.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1 hour weekly).

Assessment: Final exam, midterm exam, homework (programming and theoretical assignments) and semester project.

CS461 Software Validation, Verification and Quality

Type: Restricted Choice for SE

Level: Undergraduate

Semester: Spring

Credit: 7,5 ECTS units

Instructor: ……………….

Objectives: Teaching and assimilation of basic and specialized concepts on software validation and verification. Acquisition of knowledge for designing and implementing different types of audits and debugging procedures. Deepening in concepts and practices for ensuring quality in software systems.

Content: Basic concepts and terminology for validation and verification, reviews, inspections, control (parts, system combination, regression, acceptance, coverage criteria), control of specific categories of software systems, analyzing problems and documentation. Basic principles of quality factors and quality characteristics of software, methodologies, tools, quality procedures, quality standards, models and quality metrics, quality software production process, quality plan, organize quality assurance process, quality assessment process.

Prerequisites: CS361

Bibliography:

1. P. Jorgensen, *Software Testing, A Craftsman's Approach*, 3rd edition, Auerbach Publications, 2008.
2. P. Ammann and J. Offutt, *Introduction to Software Testing*, Cambridge University Press, 2008.
3. J. Tian, *Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvemen*t, Wiley, 2005.
4. William E. Lewis, Gunasekaran Veerapillai, *Software Testing and Continuous Quality Improvement*, 3rd Edition, 2008.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).

Assessment: Final exam, midterm exam, homework and laboratory assignments.

CS 462 Software Analysis, Modelling and Design

Type: Restricted Choice for SE

Level: Undergraduate

Semester: Spring

Credit: 7,5 ECTS units

Instructor: ……………….

Objectives: Deepening in fundamental concepts of software analysis, modelling and design. Design of effective, efficient and competitive software products. Training in software architectures, notations, standards and methods.

Content: Fundamental concepts of software modeling (principles, conditions, properties), modeling languages, virtual models, model types. Early analysis (consistency, sufficiency, accuracy, quality). The role of models in the development and the connection with software development methodologies. Model-driven engineering. The role of software architecture in the software development lifecycle (user needs, design, implementation). The

architectural design of a software system. Architectural representation.
Prerequisites: CS361
Bibliography:

1. K. Qian, X. Fu, L. Tao, C. Xu, *Software Architecture and Design Illuminated*, Jones and Bartlett Learning, 2010.
2. A. Kleppe, J. Warmer, W. Bast, *MDA Explained. The Model Driven Architecture: Practice and Promise*, Addison-Wesley Professional, 2003.
3. R.N. Taylor, N. Medvidovic and E.M. Dashofy, *Software Architecture*, Wiley, 2010.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, homework and laboratory assignments.

CS 463 Software Reuse

Type: Restricted Choice for GD and SE
Level: Undergraduate
Semester: Winter
Credit: 7,5 ECTS units
Instructor: A. Achilleos
Objectives: Understanding the usefulness of software reuse. Deepening in the different levels of reuse and understanding the differences between them. Use of software components in practice.
Content: Levels of reuse. Component-based development and composition. Best practices for reuse. Evolution of reuse. Software repositories. Search and retrieval. Design patterns. Object-oriented programming standards. Open source software. Open source licensing and legal issues. Organization policies and open-source based development. Outsourcing. Model-Driven Engineering principles. Service-Oriented Computing. Aspect-Oriented Programming.
Prerequisites: CS233, CS361
Bibliography:

1. M. Ezran, M. Morisio, C. Tully, *Practical Software Reuse*, Practicioner Series, 2002.
2. E. Freeman, E. Robson, B. Bates, K. Sierra, *Head First Design Patterns*, O'Reilly Media, 2004.
3. C. Horstmann, *A Practical Guide to Open Source Licensing*, Wiley, 2nd Edition, 2006.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final and midterm exam, homework and semester project.

CS470 Design of Embedded Systems

Type: Restricted Choice for CSN
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: P. Evripidou
Objectives: ------------
Content: A review of embedded system processors. Organization of embedded systems: CPUs, RAM, ROM, buses, peripherals, sensors, actuators, interfacing. Examples of widely used processors buses and peripherals. Interfacing with peripherals: sampling, interrupts, advantages and disadvantages. Process distribution between hardware and software. Tools for the development of embedded systems and real time operating systems. Hands on experience with the development and implementation of embedded systems.
Prerequisites: CS221
Bibliography:

1. W. Woff, *Computers as Components: Principles of Embedded Computing System Design*, Morgan Kaufman.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory (2 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS475 Network and Information Security

Type: Restricted choice for CSN
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: V. Vassiliou

Objectives: Introduction to network and information security principles, understanding of basic areas in Cryptography, Authentication and Confidentiality. Gain of knowledge in methods for the evaluation of Software, Applications and Systems with respect to security. Application of tools for the protection of networks, applications and information.

Content: Introduction to Security Threats and Attacks, Cryptographic Techniques (encryption, cryptanalysis, authentication, confidentiality), identification and authentication (Kerberos, PKI), Internet Application security protocols ((PGP, SSL/TLS), Network security (Firewalls, IDS), Defending against threats on endsystems, Checking of networks and applications for vulnerabilities, Other issues in network and information security (privacy, ethics, legal framework).

Prerequisites: CS324

Bibliography:

1. W. Stallings, *Network Security Essentials*, Third Edition, Pearson-Prentice Hall, 2007.
2. C.P. Pfleeger, S.L. Pfleeger , *Security in Computing*, Fourth Edition, Prentice Hall, 2006.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory (2 hours weekly).

Assessment: Final exam, midterm exam and homework.

CS476 Wireless Networks

Type: Restricted Choice for CSN
Level: Undergraduate
Semester:Fall
Credit: 7,5 ECTS units
Instructor: ……………..

Objectives: The objective of this course is to introduce students into wireless mobile/local/cellular networks with an emphasis on the fundamental concepts and principles of the technology which are important for the design, application, evaluation and development of these systems. The course will also cover new architectures and topologies, existing and proposed standards as well as open research issues.

Content: Wireless environment, Interference and other problems in wireless communications, Architectures and technologies of wireless networks and wireless communication, Wireless Local Area Networks (WLAN), Mobility Management Protocols at the Network layer and at higher layers (transport, application), New network technologies (ad-hoc, sensor, vehicular networks), Open research issues and challenges.

Prerequisites: CS324

Bibliography:

1. J. Schiller, *Mobile Communications*, Second edition, Addison-Wesley, 2003.
2. R. S. Koodli and C. E. Perkins, *Mobile Internetworking with IPv6: Concepts, Principles and Practices*, Wiley-Interscience, 2007.
3. W. Stallings, *Wireless Communications and Networks*, Prentice Hall, 2nd Edition, 2002.
4. H. Karl and A. Willing, *Protocols and Architectures for Wireless Sensor Networks*, Wiley, 2007.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory (2 hours weekly).

Assessment: Final exam, midterm exam and homework.

CS 499 Special Issues in Computer Science: Mobile Computing Systems

Type: Restricted Choice for CSN

Level: Undergraduate
Semester:Fall
Credit: 7,5 ECTS units
Instructor: N. Stylianides
Objectives: Design and functional principles of modern mobile computing. Provide knowledge in mobile computing application design principles, development procedure, communication schema and usability. Development of skills and attitudes for the analysis, evaluation and development of fundamental mobile computing components.
Content: General review of operating systems. Mobile operating systems. Wireless networks and object oriented programming. Software design patterns. Android OS application development procedure: application development framework, peripheral devices, database communication, user interface development, application publication and installation.
Prerequisites: CS233
Bibliography:

1. R. Meier, *Professional Android 2 Application Development*, Wiley Publishing, Inc. 2010.
2. B. Eckel, *Thinking in Java*, 4th Edition, Prentice Hall, 2006.
3. D. Gallardo, E. Burnette and R. McGovern, Eclipse in Action. A guide for Java developers, Manning, 2003.
4. C. Horstmann , *Object-Oriented Design and Patterns* , Wiley, 2nd Edition, 2006.
5. M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, Addison Wesley, 3nd Edition, 2000.
6. Επιλεγμένα άρθρα από τη διεθνή βιβλιογραφία.

Teaching methods: Lectures (3 hours weekly) and Recitation (1,5 hour weekly).
Assessment: Homework, group project and presentation, final exam, participation during lectures.

Courses for Other Departments

These courses are offered to students of other Departments. The content of such courses is suitably determined so that students in other disciplines may appreciate the significance of Computer Science, its relationship to other disciplines, and the potential benefits it offers. Each of the courses for other Departments carries 5, 6 or 7 ECTS units. The courses may be offered every semester or in parallel classes, depending on the needs and capabilities.

CS001 Introduction to Computer Science
Type: Compulsory for the students of the departments of PSY and SPS
Unrestricted Choice (for students of other departments)
Level: Undergraduate
Semester: Fall and Spring
Credit: 6 ECTS units
Instructor: P. Andreou (Fall) / …………….. (Spring)
Objectives: Introduction to the basic concepts and the wide range of Computer Science. Familiarization and global update of students with the structure and use of computers, computer programs, the Internet and the application of Computer Science to other fields.
Content: Fundamentals of Computer Science, the main historical events which have contributed to its development, and the possibilities if offers. Basic constituent elements of Computer Science and methods for making it valuable to other sciences and applications. Practical experience with application packages, basics of programming, and programming in a fourth generation language such as Logo.
Prerequisites: ---------

Bibliography:

1. B. A. Forouzan, *Εισαγωγή στην Επιστήμη των Υπολογιστών*, Εκδόσεις "ΚΛΕΙΔΑΡΙΘΜΟΣ", 2003.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory assignments).

CS002 Introduction to Computer Science

Type: Compulsory for the students of the departments of EDU
Level: Undergraduate
Semester: Fall and Spring
Credit: 5 ECTS units
Instructor: K. Neokleous
Objectives: Introduction to the basic concepts and the wide range of Computer Science. Familiarization and global update of students with the structure and use of computers, computer programs, the Internet and the application of Computer Science to other fields.
Content: Fundamentals of Computer Science, the main historical events which have contributed in its development, and the possibilities it offers. Basic constituent elements of Computer Science and methods for making it valuable to other sciences and applications. Practical experience with application packages, and the UNIX environment. Basic principles of programming in a fourth generation language.
Prerequisites: ----------
Bibliography:

1. B. A. Forouzan, *Εισαγωγή στην Επιστήμη των Υπολογιστών*, Εκδόσεις "ΚΛΕΙΔΑΡΙΘΜΟΣ", 2003.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory assignments).

CS003 Computer Science and Information Systems

Type: Compulsory for the students of the departments of ECO and PBA
Level: Undergraduate
Semester: Fall and Spring
Credit: 6 ECTS units
Instructor: ……………… (Fall) / ……………… (Spring)
Objectives: Familiarization with the most basic concepts in Computer Science, Information Systems and Computer Systems. Touch with the current trends in the practice of Computer Science. Practical experience in the use of various software packages that are useful in the academic and professional worlds.
Content: Fundamentals of Computer Science, the main historical events which have contributed in its development, and the possibilities it offers. Basic constituent elements of Computer Science and methods for making it valuable to other sciences and applications. Practical experience with application packages, basics of programming, and programming in a fourth generation language such as Logo.
Prerequisites: ----------
Bibliography:

1. N. Dale and J. Lewis, Computer Science Illuminated, Jones and Bartlett, Second Edition, 2004.

Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory sessions (1.5 hours weekly).

Assessment: Final exam, midterm exam and homework.

CS011 Introduction to Information Society

Type: Unrestricted Choice (for students of other departments)
Level: Undergraduate
Semester: ……………….
Credit: 6 ECTS units
Instructor:……………….

Objectives: Familiarization with the most prominent concepts and applications of Information Society.
Content: Presentation of the formed framework for Information Society (IST). Basic concepts and constituent elements of IST, and the wider context for its application. Issues such as electronic government, telematics, digital business, electronic commerce, telemedicine, etc. Effects of IST on society and economy.
Prerequisites: ---------
Bibliography:

1. B. A. Forouzan, *Εισαγωγή στην Επιστήμη των Υπολογιστών*, Εκδόσεις “ΚΛΕΙΔΑΡΙΘΜΟΣ”, 2003.

Teaching methods: Lectures (3 hours weekly).
Assessment: Final exam and homework (individual midterm project and final group project).

CS012 Web Design Technologies

Type: Unrestricted Choice (for students of other departments)
Level: Undergraduate
Semester: Spring
Credit: 6 ECTS units
Instructor:……………….
Objectives: (a) basic concepts of the Internet and the WWW, (b) well-know technologies for designing and developing websites and (c) the use of specialized web development software.
Content: Introduction to the Internet and the WWW. Web design and development technologies. Web servers and HTTP, HTML, XHTML, CSS, Javascript. User interface design guidelines. Usability evaluation.
Prerequisites: --------
Bibliography:

1. Σ. Ρετάλης και Γ. Τσέλιος, HTML: Μαθήματα από το Απλό στο Σύνθετο, Εκδόσεις Καστανιώτη, 2003.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).

Assessment: Final exam, midterm exam and homework.

CS013 History and Philosophy of Computation

Type: Unrestricted Choice (for students of other departments)
Level: Undergraduate
Semester: Spring
Credit: 6 ECTS units
Instructor: A. Kakas
Objectives: Basic understanding of the foundational motion of computation, Computational machine and intelligence through their historical evolution. Theoretical and practical understanding of the automation of computation. Understanding of the relationship between computing and human intelligence.
Content: Foundational motions of computation, complexity, computational machine, algorithm, programming language, knowledge and intelligence. Basic computational models and their corresponding problems. Historical development of computers and their theoretical basis. The scientific and engineering character of computer science. The relation of computer science to other disciplines such as mathematics, physics, psychology and biology. Computing today over the internet and its future development. Artificial intelligence and machines with intelligent behavior. The human brain and mind as computing machines. Automation of logic and argumentation. Natural language as a computing language.
Prerequisites: --------
Bibliography:

1. L. Floridi, *Guide to the Philosophy of Computing and Information*, Backwell 2004.
2. Selected articles from the Journal of Minds and Machines.

Teaching methods: Lectures (3 hours weekly) and Tutorial sessions (1 hour weekly).

Assessment: Final exam and homework.

CS031 Introduction to Programming

Type: Compulsory for students of MAS Department
Level: Undergraduate
Semester: Spring
Credit: 7 ECTS units
Instructor: ……………….

Objectives: Introduction of the basic principles of programming with emphasis on structured programming, abstraction, and the design, implementation, checking and debugging of modular programs. Application of these principles using the FORTRAN 90/95 programming language.

Content: Computers and binary system. Hardware and software. Program development cycle, algorithms and flow diagrams. Alphabet and syntax of FORTRAN. Operators. Selection structures and loops. Arrays. Functions and subroutines. Recursion. Formatted input-output. Files. Dynamic data.

Prerequisites: ----------

Bibliography:

1. Δ. Ματαράς και Φ. Κουτελιέρης, *Προγραμματισμός Fortran 90/95 για Επιστήμονες και Μηχανικούς*, Εκδόσεις ΤΖΙΟΛΑ, 2003.
2. L. Nyhoff and S. Leestma, *Introduction to Fortran 90*, Prentice Hall, 1999.

Teaching methods: Lectures (3 hours weekly), Laboratory sessions (1 hour weekly) and Recitation (1 hour weekly).

Assessment: Final exam, midterm exam, homework (theoretical and programming assignments) and quizzes.

CS032 Introduction to Computer Science & Information Systems

Type: Compulsory (for the students of the departments of ECO and PBA)
Level: Undergraduate
Semester: Spring
Credit: 6 ECTS units
Instructor: ……………….

Objectives: Programming is examined as a problem-solving method. In particular the course presents the fundamentals of algorithmic thought and the implementation thereof through a programming language. Also, a high level programming language is introduced. Upon completion of the course students are expected to be able to cast problem solutions into an algorithmic form, and will have obtained a basic exposure to a widely used programming language such as C or Python.

Content: Introduction to the principles of programming with emphasis on structured programming, abstraction, and the design, implementation, checking and debugging of modular programs. Mastering the material through laboratory exercises in the C programming language.

Prerequisites: --------

Bibliography:

1. W. F. Punch and R. Enbody, The Practice of Computing Using Python, Second Edition, Addison-Wesley, 2013..

Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory sessions (1.5 hours weekly).

Assessment: Final exam, midterm exam and homework.

CS033 Introduction to Programming for Engineers

Type: Compulsory for students of the Department of CEE
Level: Undergraduate
Semester: Fall
Credit: 5 ECTS units
Instructor: G. Chatzimilioudis, N. Nikolaou

Objectives: Introduction of methods for problem-solving through programming. Provision of deep understanding of basic programming principles and algorithmic techniques, design, implementation, testing and debugging of modular programs. Application of the basic principles using the C programming language.
Content: Basic principles of programming with emphasis on structured programming, abstraction, and the design, implementation, checking and debugging of modular programs. Mastering of the material through laboratory exercises in a traditional programming language such as C.
Prerequisites:---------
Bibliography:

1. J. R. Hanly and E. B. Koffman, *C Program Design for Engineers*, Second Edition, Addison-Welsey, 2001.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).

Assessment: Final exam, midterm exams and homework (including programming assignments).

CS034 Introduction to Programming for Electrical and Computer Engineers

Type: Compulsory for students of the Department of ECE
Level: Undergraduate
Semester: Spring
Credit: 7 ECTS units
Instructor: ………………
Objectives: Introduction of methods for problem-solving through programming. Development of procedural problem solving skills and algorithmic thinking. Provision of deep understanding of basic programming principles and algorithmic techniques, design, implementation, testing and debugging of modular programs. Understanding the important concepts such as program abstraction and data abstraction. Mastering of a high-level programming language (C).
Content: Introduction to computers and programming languages. Problem solving and programming, problem specification, algorithms and programs, modular programming, program and data abstraction. Software development process, top-down design, problem decomposition, reuse, trial and debugging. Variables: names, values, addresses, basic types (numbers, characters, logical values), operators and expressions, constants, library usage. Input/Output operations. Procedures (functions), parameters, calls, value or address referral. Program flow, variables' scope, lifecycle of variables/function calls, program's state. Procedural programming, algorithmic structures (sequence, selection, loop, recursion), memory. Synthesized and enumerated data types, arrays (vectors and multidimensional), structures, pointers (variables of pointer type, address and indirect referral operators, arrays and pointers and functions). Introduction to dynamic memory allocation.
Prerequisites:---------
Bibliography:

1. J. R. Hanly and E. B. Koffman, *Problem Solving and Program Design in C*, Fourth Edition, Addison-Wesley, 2003.
2. B. W. Kernighan and D. M. Ritchie, *Η Γλώσσα Προγραμματισμού C*, 2nd Edition, Εκδόσεις ΚΛΕΙΔΑΡΙΘΜΟΣ, 1990.

Teaching methods: Lectures (3 hours weekly), Laboratory sessions (2 hours weekly) and Recitation (1 hour weekly).

Assessment: Final exam, midterm exams and homework (programming assignments) and quizzes.

CS035 Data Structures and Algorithms for Electrical and Computer Engineers

Type: Compulsory for students of the Department of ECE
Level: Undergraduate
Semester: Fall
Credit: 7 ECTS units

Instructor: G. Chatzimilioudis
Objectives: This course studies methods of efficient organization and manipulation of data, complexity analysis of algorithm design techniques. The main objectives of the course are to familiarize students with data structures and their associated algorithms, techniques for evaluating the complexity of algorithms, and the development of skills for efficient algorithm design and implementation.
Content: Advanced programming techniques based on the programming language C: Recursion, Structures, Pointers, File and Memory management. Data types and abstract data types. Algorithm complexity analysis: worst-case and average-case analysis. Linear data structures: List, Stack and Queue, using static and dynamic memory allocation methods. Applications of linear data structures. Sorting algorithms: SelectionSort, InsertionSort, MergeSort, QuickSort and BucketSort. Tree data structures: Binary Trees, Binary Search Trees, Balanced Trees, B-trees. Priority Queues and Heaps. Graphs: definitions, data structures, topological sorting algorithms, graph traversal algorithms. Hashing techniques, hash functions and collision resolution techniques.
Prerequisites: CS034
Bibliography:

1. R. F. Gilberg and B. A. Fourouzan, *Data Structures: A Pseudocode Approach with C*, 2nd Edition, Thomson Publishing, 2005.
2. K. N. King, *C Programming: A Modern Approach*, 2nd Edition, W. W. Norton & Company, 2008.
3. Νικόλαος Μισυρλής, *Δομές Δεδομένων με C*, 2002.
4. Mark Allen Weiss, *Data Structures and Algorithm Analysis in C*, Addison Wesley, 1996.

Teaching methods: Lectures (3 hours weekly), laboratory sessions (2 hours weekly) and tutorials (1 hour weekly).

Assessment: Final exam, midterm exam and homework (theoretical and programming assignments)

CS041 e-Health and Medical Informatics
Type: Compulsory for students of the Medical School
Level: Undergraduate
Semester: Fall
Credit: 6 ECTS units
Instructor: Ch. Schizas
Objectives: Introducing the doctor of the future to the new world order of electronic health (e-health) and medical informatics at local, European and international level. Consolidating the legislative and social framework of e-health. Explaining and appreciating the patient-centric approach to medical practice as a prerequisite for successful implementation of e-health. Exploitation of the potential offered by information and communication technologies in medicine and clinical practice, particularly through the modeling of medical practice, procedures and knowledge. Appreciating the importance of the available medical data for making e-health a reality.
Content: Introduction to the e-health environment and the appropriate operating framework. Legislative regulatory and social background needed for its materialization. The importance of information technology in extracting useful information from vast medical databases. Applications of computer systems used for the movement of medical knowledge, medical information management, proper use of a citizen's electronic folder for patients and support a medical decision. Reference to the legal framework that regulates the medical practice, in accordance with European and international directives.
Prerequisite: ---------
Bibliography: no book that covers all material expected to be taught in the course exists. Extensive use of the Internet will be recommended, even in the classroom. The following books are recommended.

1. D. Koutsouris, S. Pavlopoulos, A. Prentza, *Introduction to Biomedical Engineering and Systems Analysis*, Publisher: Tziola, 2003.
2. E. H. Shortliffe, J. J. Cimino (Eds.), *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*, Springer Verlag; 3rd edition, 2006.

Teaching methods: Lectures/presentations (3 hours weekly), tutorial (1 hour weekly), discussions/presentations. Structured visits to hospital clinics for data collection and exploitation as to how these can be used by the medical staff.
Assessment: Written exams (midterm and final), essays (studies and/or exercises).

CS042 eHealth Seminars

Type: Compulsory for students of the Medical School
Level: Undergraduate
Semester: Spring
Credit: 2 ECTS units
Instructor: Ch. Schizas
Objectives: Familiarization with the information and communication technologies (ICT) and their practical application in medicine and the medical environment. Acquiring of knowledge through eHealth practices adopted in real life by the distinguished presenters of the seminars. Study through videos practices in real environments such as, the intensive care unit, laparoscopic operations, robotic assisted operations, telediagnosis, teleconsultation; appreciation of the importance of medical databases. Understanding of what is coming up in future medicine, considering that ICT will continue to grow and offer its services to the medical profession.
Content: Selected presentations/demonstrations by at least six medical experts selected from Cyprus or abroad according to their availability. Teleconferencing may be used for real life presentations and communication with the selected presenters.
Prerequisite: CS041
Bibliography: no book that covers all material expected to be taught in the course exists. Relevant bibliography will be provided by the presenters of the seminars and the Instructor of the course.
Teaching methods: Lectures/presentations (1.5 hours biweekly). Structured visits to hospital clinics in coordination with the presenters and the Medical School.
Assessment: Written e-exam, assignments (one individual and one group assignment).

Post Graduate Program of Studies

The postgraduate programme of studies leads to Master and Ph.D. Degrees in Computer Science.

The post graduate studies in the Department are governed by Corresponding Rules of Post Graduate Studies, approved by the Senate (149th meeting 22/5/2002). These rules are included in the Appendix B of this guide. Also in Appendices C and D present the specifications to be met by the final form of the Master and Ph.D. dissertations, respectively.

Masters Programs

The Department of Computer Science, since the academic year 2009-2010, offers the following four Master Programmes:

1. Master in Computer Science (MCS)
2. Master in Internet Computing (MIC)
3. Master in Intelligent Systems (MIntS)
4. Professional Master in Advanced Information Technologies (PM).

Since 2012-2013 the Department offers in collaboration with the Department of Communication and Internet Studiesof the Cyprus University of Technology (CUT) the following Master Program:

5. Master in Computer Games (MCG)

In accordance with the ECTS System, successful completion of at least 90 ECTS is required for obtaining a Master Degree at the postgraduate level with the excemption of the Professional Master in Advanced Information Technologies that requires successful completion of at least 75 ECTS.

To be admitted to a Master programme, a candidate must have a first degree in Computer Science or a related subject from an accredited university with an overall grade of “Very Good”. Any relevant industrial experience may be considered as an additional advantage.

In the Master Programs that include a Master Thesis, the thesis should be submitted upon completion to the digital library of the University of Cyprus, Lekythos (https://lekythos.library.ucy.ac.cy/?locale-attribute=en). A brief description of these four masters:

1. MASTER in COMPUTER SCIENCE

The Master Programme in Computer Science is designed primarily for Computer Science and other science-related graduates who seek to develop research skills and enhance their knowledge in advanced areas of Computer Science. Students who attend this programme may pursue a PhD degree after their graduation.

The completion of the programme requires 90 ECTS, and the duration of studies should be at least three semesters. These 90 credits correspond to eight courses and a Master thesis. More specifically:

- Seven postgraduate courses of 8 ECTS each (any seven from the postgraduate course list with the supervision of the Academic Advisor)
- One postgraduate course of 4 ECTS (CS670 Research Methodologies and Professional Practices in Computer Science)
- Master Thesis worth of 30 ECTS.

The seven graduate courses must be from the graduate course catalogue of the Department. Each student selects these courses with the guidance of the academic advisor.

In case a student successfully complete at least four courses from the specialized Master Direction of the Department then the student may transferred to the corresponding Specialzed Master where will receive his diploma.

2. MASTER in INTERNET COMPUTING

The Master Programme in Internet Computing is designed primarily for Computer Science and other science-related graduates who seek to deepen their knowledge on subjects like Communication Networks, Distributed Systems and the Internet to develop research skills in these subjects and, potentially, pursue a doctoral degree after their graduation.
The completion of the programme requires 90 ECTS and study duration of at least three semesters. These 90 credits correspond to eight courses and a Master thesis. More specifically:

- Seven postgraduate courses of 8 ECTS, out of which 4 should be related to the area of Internet Computing (identified as such in the Table of Specialization Courses and in the course descriptions).
- One postgraduate course of 4 ECTS (CS670 Research Methodologies and Professional Practices in Computer Science)
- Master Thesis worth of 30 ECTS (the topic of the thesis must be related to the scope of the programme: Internet, Communication Networks, Distributed Computing, etc.)

3. MASTER in INTELLIGENT SYSTEMS

The Master in Intelligent Systems is designed for Computer Science and other science-related graduates who seek to deepen their knowledge in areas like Artificial Intelligence and Computational and Data Mining Systems, to develop research skills in these subjects and, potentially, follow doctoral studies after their graduation.

The completion of the program requires 90 ECTS and the duration of studies should be at least threesemesters. These 90 credits correspond to eight courses and a thesis. More specifically:

- Seven postgraduate courses of 8 ECTS, out of which 4 must be relevant to the area of Intelligent Systems (identified as such in in the Table of Specialization Courses and in the course descriptions)
- One postgraduate course of 4 ECTS (CS670 Research Methodologies and Professional Practices in Computer Science)
- Master Thesis worth of 30 ECTS (the topic of the thesis must be related to the scope of the programme: Artificial Intelligence, Computational Knowledge Mining Systems, etc.)

4. MASTER in ADVANCED INFORMATION TECHNOLOGIES (PROFESSIONAL)

The aim of the Professional Master in Advanced Information Technologies is to help Information Technology professionals to extend and update their knowledge in Advanced Computer Technologies and to acquire up-to-date know-how in subjects related to the national Information Technology industry like Software Engineering, the Internet, and Intelligent Systems.

The completion of the course requires 75 ECTS and the duration of studies must be at least four semesters. In particular:

- Seven postgraduate courses of 8 ECTS, out of which 4 should be related to the programme´s scope (identified as such in the Table of Specialization Courses and in the course descriptions)
- One postgraduate course of 4 ECTS (EPL 670 Research Methodologies and Professional Practices in Computer Science)
- Master Thesis worth of 15 ECTS, which can be replaced with two extra postgraduate courses.

The needs of employees and professionals in the Information Technology industry will be considered during the scheduling of courses (afternoon and evening courses and three-hour meetings).

5. MASTER in DESIGN AND DEVELOPMENT OF COMPUTER GAMES AND INTERACTIVE TECHNOLOGIES

The master program in Design and Development of Computer Games and Interactive Technologies.is jointly awarded by the University of Cyprus (Department of Computer Science) and the Cyprus University of Technology (Department of Multimedia & Graphic Arts). Rapid developments in interactive technologies – especially entertainment technologies - as well as the broad use of interactive virtual environments and computer games, for both entertainment and other purposes (e.g. education, information dissemination etc) are indisputable. The proper design and effective development of such applications require in-depth study and sound understanding of the context of respective scientific areas.

Upon graduation, students will have excellent knowledge and skills for/in:

- Analysis of requirements for the design of electronic games and interactive technologies
- Interactive design in computer games and interactive
- Design and development of 3D virtual environments
- Development of computer games and 3D interactive virtual environments
- Design and development of computer games for specific application domains (i.e. learning, education and information).
- Development of 3D interactive simulators (i.e. driving simulators, flight simulators etc.)
- Use a range of software for computer games development and interactive technologies integration (i.e. virtual reality systems, motion tracking equipment).

The following is the structure of the master program.

<u>1º semester</u> (three out of the following five courses, students with a limited background in Computer Science may be asked to attend the course CS131 Programming Principles I)

MGA 511 Human- Computer Interaction (7.5 ECTS)
MGA 520 3D modelling for Interactive Applications (7.5 ECTS) - CS 607 Graphic and Visual Computation (8 ECTS)
CS 131 Introduction to Programming
Elective Course

<u>2º semester</u>

MGA 518 Games Design (7.5 ECTS)
CS 653 Software engineering for Computer Games (8 ECTS)
Elective Course
Elective Course

3º semester

MGA 580/ CS 700 Master Thesis*

* Compulsory after successful completion of the rest courses

Elective courses (Winter semester)

CS 604	Artificial Intelligence	8 ECTS
MGA 699	Special Topics in Design	
CS 699	Special Topics in Computer Science	8 ECTS

And any of other courses offered by other postgraduate programmes of the two departments and are related to the master program.

Elective courses (Spring semester)

MGA 515	Creative Technologies	7.5 ECTS
MGA 519	Virtual reality	7.5 ECTS
MGA 699	Special Topics in Design	
CS 656	Computer Graphics: Modelling and Realism	8 ECTS
CS 658	Digital Video Processing	8 ECTS
CS 668	Computer Vision	
CS 699	Special Topics in Computer Science	
BUS 434	Entrepreneurship	

And any of other courses offered by other postgraduate programmes of the two departments and are related to the master program.

Table 2: Specialization Courses in Graduate Programs

Course Code and Title	Master in Computer Science[1]	Master in Internet Computing	Master in Intelligent Systems	Professional Master
CS601 – Distributed Systems	√	√		
CS602 – Foundations of Web Technologies	√	√		√
CS603 – Advanced Software Engineering	√			√
CS604 – Artificial Intelligence	√		√	
CS605 – Advanced Computer Architecture	√			√
CS606 – Computer Networks and the Internet	√	√		√
CS607 – Visual Computing	√			√
CS646 – Advanced Topics in Databases	√	√		√
CS651 – Mobile Computing and Data Management	√	√		√
CS653 – Computer Networks	√		√	√
CS655 – Advanced Parallel Processing II	√			√
CS656 – Computer Graphics - Modelling and Realism	√			√
CS657 – Wireless Networks	√	√		
CS658 – Digital Video Processing	√		√	
CS659 – Design with Embedded Processors	√			√
CS660 – Information Retrieval and Search Engines	√	√	√	
CS662 – Machine Learning and Data Mining	√		√	√
CS663 – Computational Logic	√		√	
CS664 – Systems Analysis and Verification	√			
CS665 – Constraint Solving Methods	√		√	
CS667 – Neuroinformatics	√		√	
CS668 – Mechanical Vision	√			√
CS673 – Algorithmic Game Theory	√	√		
CS674 – System and Network Security	√	√		√
CS675 – Web Services and Service Oriented Computing	√	√		√
CS679 – Electronic Health	√		√	√
CS699 – Special Topics in Computer Science	√			
Courses from other Departments				
ECE621 – Random Processes		√		
ECE635 – Optimization Theory and Applications		√		
ECE658 – Computer Systems’ Performance Evaluation and Simulation		√		

[1] No specialization courses exist. All courses are eligible.

Short Course Description

For the current academic year 2013-2014, each course description includes the name of the instructor, however, for the restricted courses that are not offered for the current academic year, 2013-2014, the name of the instructor who suggested the course is included. The language of instruction for all courses is Greek.

CS601 Distributed Systems
Type: Specialisation Course for MIC and Restricted Choice for MCS, MIntS, PM
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: Ch. Georgiou
Objectives: Familiarization with fundamental concepts and principles of distributed systems in both breadth and depth. Development of capabilities of designing, analyzing and programming distributed systems.
Content: Basic concepts and principles of distributed systems. Communication, processes and synchronization. Naming. Distributed file systems and distributed operating systems. Security and cryptography in distributed systems. Distributed shared memory and its consistency. Fault-tolerance. Distributed algorithms and distributed programming. Design and development of applications in distributed environments. Case-studies of specific distributed systems (eg. PlanetLab). Practical exposition with programming project or programming exercises.
Prerequisites: ----------
Bibliography:

1. A. S. Tanenbaum and M. van Steen, *Κατανεμημένα Συστήματα: Αρχές και Υποδείγματα*, Εκδόσεις Κλειδάριθμος, 2005.
   (English Version: A. S. Tanenbaum and M. van Steen, *Distributed Systems: Principles and Paradigms*, Pearson Education, Second Edition, 2007).
2. G. Goulouris, J. Dollimore and T. Kindberg, *Distributed Systems - Concepts and Design*, Fourth Edition, Addison-Wesley, 2005.

Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory Lecture (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (programming assignments).

CS602 Foundations of Web Technologies
Type: Specialisation Course for MIC, PM and Restricted Choice for MCS, MIntS
Level: Graduate
Semester: -----------
Credit: 8 ECTS units
Instructor: M. Dikaiakos
Objectives: To provide students with an advance introduction to the foundations and principles of the world wide web. To examine selected topics in web technologies from the recent research literature.
Content: Web Architecture. Web Protocols. Markup Languages. Web Characterization. Search Engines. Service-provision Infrastructures.
Prerequisites: -----------
Bibliography:

1. R. Krishnamurthy and J. Rexford, *Web Protocols and Practice*, Addison-Wesley, 2001.
2. M. Hofmann and L.R. Beaumont, *Content Networking: Architecture, Protocols and Practice*, Morgan Kaufman, 2005.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, homework and class participation.

CS603 Advanced Software Engineering

Type: Specialisation Course for PM and Restricted Choice for MCS, MIC, MIntS
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: G. Papadopoulos
Objectives: Familiarization with advanced topics in Software Engineering with emphasis on two main axes: (1) Principles of designing and implementing component-based systems on the basis of modern techniques of development such as component-based software, software architectures and middleware-platforms. (2) Special topics on the development of software, such as requirements engineering, real-time software systems, and software cost estimation.
Content: Topics in Component-Based Software: Principles of development of component-based systems based on component-based software. Modeling techniques. Software architectures. Coordination programming. Middleware platforms for the development of systems. Software composition. Elements of the distributed programming. Configuration management. Advanced topics in Software Engineering: Requirements Engineering Processes. Real-time Software Design. Design with Reuse. User Interface Design. Software Change.
Prerequisites: Undergraduate course equivalent to the CS361 (Software Engineering)
Bibliography:

1. C. Szyperski, *Component Software: Beyond Object-Oriented Programming*, 2nd Edition, Addison-Wesley, 2002.
2. I. Sommerville, *Software Engineering*, 9th Edition, Addison-Wesley, 2011.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam and homework (group projects).

CS604 Artificial Intelligence

Type: Specialisation Course for MIntS and Restricted Choice for MCS, MIC, PM
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: Y. Dimopoulos and Ch. Christodoulou
Objectives: Introduction to the fundamental principles, algorithms and techniques that support the operation of intelligent information systems. Upon completion of the course, the students should know the state of the art in Computational or Artificial Intelligence, the main problems under investigation in these areas, and the basic methods used for solving them.
Content: Introduction to Artificial Intelligence. Intelligent Agents. Search. Constraint Satisfaction. Knowledge Representation and Logic Planning. Introduction to Machine Learning. Introduction to Artificial Neural Networks. Single layer and Multi layer Perceptrons. Radial-basis Function Networks. Reinforcement Learning. Self-organizing Maps. Hopfield Neural Networks and Boltzmann Machines. Recurrent Neural Networks.
Prerequisites: ----------
Bibliography:

1. S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, Second Edition, Prentice Hall, 2002.
2. S. Haykin, *Neural Networks and Learning Machines*, 3rd Edition, Pearson Education, 2009.

Teaching methods: Lectures (3 hours weekly) and Recitation (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS605 Advanced Computer Architecture

Type: Specialisation Course for PM and Restricted Choice for MCS, MIC, MIntS
Level: Graduate
Semester: Fall
Credit: 8 ECTS units

<u>*Instructor*</u>: Y. Sazeidis
<u>*Objectives*</u>: Students will learn: (a) current methodology for performance evaluation and comparison of computer systems; (b) basic and advanced concepts in the organization of modern microprocessors; and (c) current trends in the computer architecture area. Also, with the use of different tools, presented in the course, students will perform research projects in certain computer architecture topics.
<u>*Content*</u>: Performance evaluation and comparison, as well as benchmarking programs; Basic microarchitecture concepts of modern processors; Pipelining, instruction-level parallelism, prediction, speculation, memory hierarchy, and static/dynamic instruction scheduling; Examples of modern processors; Current research projects in the area of computer architecture.
<u>*Prerequisites*</u>: Undergraduate course equivalent to the CS221 (Computer Organization and Assembly Programming) and undergraduate course equivalent to the CS222 (Operating Systems).
<u>*Bibliography*</u>:

1. J. Henessy and D. Patterson, *Computer Architecture: A Quantitative Approach*, 3rd Edition, Morgan Keufmann, 2002.
2. Selected research articles from the international literature.

<u>*Teaching methods*</u>: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).

<u>*Assessment*</u>: Final exam, midterm exam, homework (final project, intermediate assignments), class participation and brief presentation.

CS606 Computer Networks and the Internet
<u>*Type*</u>: Specialisation Course for MIC, PM and Restricted Choice for MCS, MIntS
<u>*Level*</u>: Graduate
<u>*Semester*</u>: Fall
<u>*Credit*</u>: 8 ECTS units
<u>*Instructor*</u>: V. Vassiliou
<u>*Objectives*</u>: Understanding (at a graduate level) of the basic concepts and matters regarding Computer Networks and the Internet. Familiarization with modern views of Computer Networks and exposure to the related open research problems.
<u>*Content*</u>: Introduction to Internet and Networking Technologies. TCP/IP suite of protocols, Quality of Service (QoS), New Networking Architectures. Protocols and Standards (e.g. DiffServ, IPv6, MPLS). Network Performance Evaluation (e.g. queueing theory, and simulation tools). Traffic Modeling and Traffic Engineering. Congestion Control and Resource Allocation. Network Design and Optimization.
<u>*Prerequisites*</u>: Undergraduate course equivalent to the CS324 (Communications and Networks)
<u>*Bibliography*</u>:

1. L. Peterson and B. Davies, *Computer Networks: A Systems Approach*, 5th Edition, Morgan Keufmenn, 2011.
2. J. F. Kurose and K. W. Ross, *Computers Networking – A Top Down Approach to the Internet*, 6th Edition, Addison-Wesley, 2011.

<u>*Teaching methods*</u>: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (2 hours weekly).

<u>*Assessment*</u>: Final exam and homework (Individual or Group Project and laboratory exercises).

CS607 Visual Computing
<u>*Type*</u>: Specialisation Course for PM and Restricted Choice for MCS, MIC, MIntS
<u>*Level*</u>: Graduate
<u>*Semester*</u>: Fall
<u>*Credit*</u>: 8 ECTS units
<u>*Instructor*</u>: C. Pattichis / Y. Chrysanthou
<u>*Objectives*</u>: -----------
<u>*Content*</u>: Binary image processing, intensity transformations, the discrete Fourier

transform, linear and nonlinear filtering, image compression, image analysis, basic principles of video processing. Basic principles of 3Dgraphics: polygonal representations, transformations, local and world

coordinate system, scene graph, camera and field of view specification, orthographic and perspective projection, clipping in 2D & 3D, polygon rasterization, back face elimination, visible surface determination with the Z-byffer method and Binary Space Partitioning Trees, local illumination - flat, Phong & Gouraud shading, real-time graphics, applications.

Prerequisites: Programming in C, basic linear algebra

Bibliography:

1. A. Watt and F. Policarpo, *The Computer Image*, Addison–Wesley, 1998.
2. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd Edition, Addison–Wesley, 2002.
3. M.Slater, A. Steed and Y. Chrysanthou, *Computer Graphics and Virtual Environments: From Realism to Real-Time*, Addison-Wesley, 2001.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).

Assessment: Final exam 40%, Midterm Exam 20%, Exercises and Project 40%.

CS608 Programming for Games and Interactive Technologies

Type: Specialisation Course for MCG

Level: Graduate

Semester: Fall

Credit: 8 ECTS units

Instructor: ……………….

Objectives: Learning of theory and software development skills for understanding and solving computational problems that are common in the process of creating video games and other modern interactive applications. This is accomplished through the teaching of methods for analyzing problems, construction of suitable algorithms for their solution and their implementation, initially by using visual programming languages and subsequently third generation programming languages.

Content: Teaching material includes (a) introduction to the use of computers and programming languages for creating games and interactive applications, (b) analysis, specification and problem solving of applications in Computer Graphics, (c) program and data abstraction, (d) construction, articulation, documentation and implementation techniques for good programming practices, debugging and source code reuse, (e) hands on use of visual programming and scripting programming languages.

Bibliography:

1. C. Reas and B. Fry, *Getting Started with Processing*, O'Reilly Media, 2010.
2. A. S. Glassner, *Processing for Visual Artists: How to Create Expressive Images and Interactive Art*, A K Peters, 2010.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).

Assessment: final exam, midterm exam and project.

CS646 Advanced Topics in Databases

Type: Specialisation Course for MIC and PM and Restricted Choice for MCS and MIntS

Level: Graduate

Semester: Winter

Credit: 8 ECTS units

Instructor: D. Zeinalipour

Objectives: The main objectives of this graduate-level course are to provide an in-depth understanding of advanced concepts and research directions in the field of databases. The course is organized in three parts: (i) Fundamentals of Database Systems Implementation; (ii) Distributed, Web and Cloud Databases; (iii) Spatio-temporal Data Management, Sensor Data Management, other selected and advanced topics from the recent scientific literature.

Content: (i) Fundamentals of modern Database Management Systems (DBMSs): storage, indexing, query optimization, transaction processing, concurrency and recovery. (ii) Fundamentals of Distributed DBMSs, Web Databases and Cloud Databases (NoSQL /

NewSQL): Semi-structured data management (XML/JSON, XPath and XQuery), Document data-stores (i.e., CouchDB, MongoDB, RavenDB), Key-Value data-stores (e.g., BerkeleyDB, MemCached), Introduction to Cloud Computing (GFS, NFS, Hadoop HDFS, Replication/Consistency Principles), "Big-data" analytics (MapReduce, Apache's Hadoop, PIG), Column-stores (e.g., Google's BigTable, Apache's HBase, Apache's Cassandra), Graph databases (e.g., Twitter's FlockDB) and Overview of NewSQL (Google's Spanner and Google's F1). (iii) Spatio-temporal data management (trajectories, privacy, analytics) and index structures (e.g., R-Trees, Grid Files) as well as other selected and advanced topics, including: Embedded Databases (sqlite), Sensor / Smartphone / Crowd data management, Energy-aware data management, Flash storage, Stream Data Management, etc.

Prerequisites: Undergraduate course equivalent to the CS342 (Database Systems)

Bibliography:

1. S. Abiteboul, I. Manolescu, P. Rigaux, M.-C. Rousset, P. Senellart, *Web Data Management*, Cambridge University Press, 2011.
2. R. Elmasri, S. Navathe, *Fundamentals of Database Systems*, 6th Edition, Addison-Wesley, 2011.
3. T. Özsu, P. Valduriez, *Principles of Distributed Database Systems*, 3rd Edition, Springer Press, 2011.
4. R. Ramakrishnan, J. Gehrke, *Database Management Systems*, 3rd Edition, McGraw-Hill, 2003.

Teaching methods: Lectures (3 hours weekly), and Laboratory sessions (1.5 hours weekly).

Assessment: Midterm, final exam and homework (assignments and presentation).

CS651 Data Management for Mobile Computing

Type: Specialization Course for MIC and PM and Restricted Choice for MCS, MIntS

Level: Graduate

Semester: Spring

Credit: 8 ECTS units

Instructor: G. Samaras

Objectives: Introduction to fundamental concepts, applications and limitations of mobile computing. Familiarization with practical applications and research topics of current interest in the field of Mobile Computing.

Content: Introduction (wireless technologies, architectures, applications, limitations). Software architectures for mobile computing. Theoretical models for mobile computing. Support for information recovery. Information Management. Dynamic redirection of computations. Indicative applications and open problems.

Prerequisites: Undergraduate course equivalent to the CS446 (Advanced Topics in Databases) and undergraduate course equivalent to the CS324 (Communications and Networks)

Bibliography:

1. E. Pitoura and G. Samaras, *Data Management for Mobile Computing*, Kluwer Academic Publisher, 1998.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).

Assessment: Final exam, midterm exam and homework.

CS653 Computer Games Software Technology

Type: Specialisation Course for MIntS, PM and Restricted Choice for MIC, MCS

Level: Graduate

Semester: Spring

Credit: 8 ECTS units

Instructor: ------------

Objectives: Study in depth of the technologies of Electronic Commerce. Introduction to the software technology of client/server systems of e-Commerce and to Business Models of e-

Commerce.
Content: Game structure and design, computer animation, movement and deformation, interactive cameras, visual simulation of physically-basedmodels, special effects using particle systems, collision detection, articulated characters, navigation and other behavioural models for autonomous characters._
Prerequisites: ---------
Bibliography:

1. R. Parent, *Computer Animation: Algorithms and Techniques*, Morgan Kaufmann, 2002.
2. A. Watt and M. Watt, *Advanced Animation and Rendering Techniques*, Addison-Wesley, 1992.
3. I. Millngton, *Artificial Intelligence for Games*, Morgan Kaufmann, 2006.

Teaching methods: Lectures (3 hours weekly) and Laboratory (1,5 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS655 Advanced Parallel Processing II

Type: Specialisation Course for PM and Restricted Choice for MCS, MIC, MIntS
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: P. Trancoso
Objectives: Introduction to fundamental architectural concepts and methods for parallel program execution. Examine the challenges and difficulties for parallel computing.
Content: Support for parallel program execution, parallel architectures, different types of multiprocessor interconnection networks, compilation of parallel programs, and performance analysis of various parallel applications.
Prerequisites: Undergraduate course equivalent to the CS370 (Computer Architecture) or undergraduate course equivalent to the CS605 (Advanced Computer Architecture I) or the consent of the instructor.
Bibliography:

1. J. Henessy and D. Patterson, *Computer Architecture: A Quantitative Approach*. Third Edition, Morgan Kaufmann, 2002.
2. D. E. Culler and J. P. Singh. *Parallel Computer Architecture: A Hardware/Software Approach*, Morgan Kaufmann, 1998.
3. Selected research articles from the international literature.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (group project and exercises).

CS656 Computer Graphics: Modeling and Realism

Type: Specialisation Course for PM and Restricted Choice for MCS, MIC, MIntS
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: Y. Chrysanthou
Objectives: This course goes beyond the basics of digital image synthesis, looking at issues such as photo-realistic rendering, modeling and animation. A big component for this are the creation of realistic and detailed models as well as the faithful simulation of light transport. We will see how these can be applied to virtual and augmented reality. Students will acquire both the theoretical foundations as well as practical skills since a significant part of the course is the student project.
Content: Modeling, parametric and implicit surfaces, camera specification, projections of primitives. Graphics Pipeline. Local and global illumination, shadows, ray tracing and radiosity. Real-time rendering of large environments. Acceleration techniques.
Prerequisites: C Programming and basic Linear Algebra.

Bibliography:

1. M. Slater, A. Steed and Y. Chrysanthou, *Computer Graphics and Virtual Environments: From Realism to Real-Time*, Addison-Wesley, 2001.
2. A. Watt, *3D Computer Graphics*, 3rd Edition, Addison-Wesley, 2001.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (group project and exercises).

CS657 Wireless Networks

Type: Specialisation Course for MIC and Restricted Choice for MCS, MIntS, PM
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: A. Pitsillides
Objectives: Introduction to wireless networks (mobile/local/cellular/Ad-hoc/Sensor) with an emphasis on the fundamental concepts and principles of the technologies which are important for the design, application, evaluation and development of these systems. The course will also cover new architectures and topologies, existing and proposed standards, as well as open research issues.
Content: Wireless environment, Interference and other problems in wireless communications, basic principles of wireless local and metropolitan area networks, and cellular wireless networks. New architectures and technologies of wireless networks and wireless communication (e.g., ad-hoc and sensor networks, VANETS). Resource management techniques, Next Generation wireless networks, design and planning of wireless networks, protocols for wireless and mobile networks. Internet/Web of Things.
Prerequisites: Undergraduate course equivalent to the CS324 (Communications and Networks)
Bibliography:

1. H Karl and A. Willing, *Protocols and Architectures for Wireless Sensor Networks*, Wiley, 2005.
2. K. Sohraby, D. Minoli and T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*, Published Online, 2006.

   K. Sohraby, D. Minoli and Taieb, *Wireless Sensor Networks: Technology, Protocols, and Applications*, 2006,

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (2 hours weekly).
Assessment: Final exam and homework (including Individual or Group Project and laboratory exercises).

CS658 Digital Video Processing

Type: Specialisation Course for MIntS and Restricted Choice for MCS, PM, MIC
Level: Graduate
Semester: ----------
Credit: 8 ECTS units
Instructor: C. Pattichis
Objectives: Basic familiarization with the application of current technology for processing and coding video.
Content: Basic of analog and digital video. Frequency domain analysis of video signals, spatial and temporal frequency response of the human visual system. Scene, camera, and motion modeling, 3D motion and projected 2D motion, models for typical camera/object motions. 2D motion estimation. Basic compression techniques. Waveform-based coding. Video compression standards (H.261 and H.263, MPEG-1, MPEG-2, MPEG-4, MPEG-7, MPEG-21).
Prerequisites: ----------
Bibliography:

1. Y. Wang, J. Ostermann and Y. Q. Zhang, *Video Processing and Communications*, Prentice Hall, 2002.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory exercises, additional

exercises, final study).

CS659 Design on Embedded Systems
Type: Specialisation Course for PM and Restricted Choice for MCS, MIC, MIntS
Level: Graduate
Semester: ----------
Credit: 8 ECTS units
Instructor: P. Evripidou
Objectives: ---------
Content: A review of embedded system processors. Organization of embedded systems: CPUs, RAM, ROM, buses, peripherals, sensors, actuators, interfacing. Examples of widely used processors buses and peripherals. Interfacing with peripherals: sampling, interrupts, advantages and disadvantages. Process distribution between hardware and software. Tools for the development of embedded systems
and real-time operating systems. Hands-on experience with the development and implementation of embedded systems.
Prerequisites: Knowledge on the subjects of Digital Systems, Computer Organization and Assembly Programming
Bibliography:

1. F. Vahid and T. Givargis, *Embedded System Design: A Unified Hardware/Software Introduction*, John Wiley & Sons, 2002.
2. W. Wolf, *High-Performance Embedded Computing: Architectures, Applications and Methodologies*, Morgan Kaufman.
3. W. Wolf, *Computers as Components: Principles of Embedded Computing System Design*, Morgan Kaufman.
4. P. Raghavan, A. Lad, S. Neelakandan, *Embedded Linux System Design and Development*, Auerbach Publications

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly) Laboratory sessions (2 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS660 Information Retrieval and Search Engines
Type: Specialisation Course for MIC, MIntS and Restricted Choice for MCS, PM
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: G. Pallis
Objectives: The objective of this course is to examine the main computer science principles that lie behind Google and other search engines. To this end, the course will focus on basic and advanced techniques for text-based information systems: efficient text indexing; Boolean and vector space retrieval models; evaluation and interface issues; text classification and clustering. The course will also focus on Web search including crawling, link-based algorithms, and Web metadata.
Content: Introduction to Information Retrieval. Boolean Retrieval.Text encoding: tokenisation, stemming, lemmatisation, stop words, phrases. Dictionaries and Tolerant retrieval. Index Construction and Compression. Scoring and Term Weighting. Vector Space Retrieval. Evaluation in information retrieval. Relevance feedback/query
expansion. Text classification and Naive Bayes. Vector Space Classification. Flat andHierarchical Clustering.Web Search Basics.Web crawling and indexes. Link Analysis.
Prerequisites: Algorithms, Data Structures, Internet Technologies and Linear Algebra
Bibliography:

1. C. D. Manning, P. Raghavan and H. Schutze, *An Introduction to Information Retrieval,* Cambridge University Press, 2008.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS662 Machine Learning and Data Mining
Type: Specialisation Course for MIntS, PM and Restricted Choice for MCS, MIC

Level: Graduate
Semester: --------
Credit: 8 ECTS units
Instructor: C. Pattichis
Objectives: Introduction of the fundamental principles, algorithms and techniques that support the development and implementation of data mining systems leading in the extraction of knowledge.
Content: Data Warehouse and OLAP Technology for Data Mining. Data Processing. Data Mining Primitives, Languages, and System Architectures. Concept Description: Characterization and Comparison. Mining Association Rules in Large Databases. Classification and Prediction. Cluster Analysis. Mining Complex Types of Data. Applications and Trends in Data Mining.
Prerequisites: --------
Bibliography:

1. J. Han and M. Kamber, *Data Mining – Concepts and Techniques*, Morgan Kaufmann, 2000.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (case studies, exercises, oral presentation of a case study).

CS663 Computational Logic
Type: Specialisation Course for MIntS and Restricted Choice for MCS, MIC, PM
Level: Graduate
Semester: ----------
Credit: 8 ECTS units
Instructor: A. Kakas
Objectives: Familiarization with fundamental concepts and applications of Computational Logic. Familiarization with current research trends in Computational Logic.
Content: Historical introduction. Review of Classical Logic. Abduction and induction. Knowledge representation and knowledge. Reasoning about Actions and Change. Application of Computational Logic. Declarative Programming. Autonomous Agents. Knowledge-based Robotics. Intelligent Information Integration.
Prerequisites: Undergraduate course equivalent to the CS324 (Communications and Networks)
Bibliography:

1. Selected research articles from the international literature.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam and homework.

CS664 System Analysis and Verification
Type: Restricted Choice for MCS, MIC, MIntS, PM
Level: Graduate
Semester: -----------
Credit: 8 ECTS units
Instructor: A. Philippou
Objectives: Familiarization with current technologies for modeling, analysis and verification of computer systems.
Content: Formal methods for system specification and analysis. Concurrent systems and interleaving and partial-order semantics. Transition systems and Kripke structures. Temporal logic (linear and branching). Automatic verification and model-checking. Process algebras: syntax, semantics, equivalence relations and axiom systems. Real-time system analysis (timed automata, timed process algebras and timed temporal logic). The tools SPIN and Concurrency Workbench.
Prerequisites: ----------
Bibliography:

1. D. Peled, *Software Reliability Methods*, Springer-Verlag, 2001.
2. C. Baier and J.-P. Katoen, Principles of Model Checking. MIT Press, 2008

3. L. Aceto, A. Ingólfsdóttír, K. G. Larsen and J. Srba, *Reactive Systems: Modelling, Specification and Verification*. Cambridge University Press, 2007.
4. Selected research articles from the international literature.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam and homework.

CS665 Constraint Solving Methods

Type: Specialisation Course for MIntS and Restricted Choice for MCS, MIC, PM
Level: Graduate
Semester: --------
Credit: 8 ECTS units
Instructor: Y. Dimopoulos
Objectives: A significant number of problems in Computer Science over a wide spectrum ranging from Computer Vision and Artificial Intelligence to the Management of Computer Networks and Scheduling are special cases of Constraint Satisfaction problems. This course covers advanced techniques for solving such problems and describes how they can be used in tackling real-life applications.
Content: Review of basic concepts from Constraint Satisfaction over Finite Domains. Advanced Consistency Techniques in Binary and non-Binary problems. Constraint Satisfaction and Propositional Logic: new algorithms and reduction techniques. Logic Programs with negations and the systems SMODELS and DLV. Relation between Constraint Satisfaction and Propositional Satisfiability. Redundant Constraints. Planning and Constraint Satisfaction. Satisfaction of Temporal Constraints. Introduction to Distributed Constraint Satisfaction. Problem solving with CHIP.
Prerequisites: CS604 or CS433
Bibliography:

1. R. Dechter, *Constraint Processing*, Morgan Kaufmann, 2003.
2. Selected research articles from the international literature.

Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory sessions (1 hour weekly).
Assessment: Final exam, midterm exam and programming exercises.

CS667 Neuroinformatics

Type: Specialisation Course for MIntS and Restricted Choice for MCS, MIC, PM
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: Ch. Christodoulou
Objectives: Neuroinformatics or Computational Neuroscience is an emerging and dynamically developing field aiming to elucidate the principles of information processing by the nervous systems well as applying information technology to the processing of neuroscientific data. This course aims to develop and apply computational methods for studying brain and behaviour as well as understanding the dynamics of the conscious mind.
Content: Introduction to Neuroinformatics; basic neurobiology: from the brain to single neurons; biophysics of single neurons; synapses; dendrites and axons. Conductance-based neuron models: the generation of action potentials and the Hodgkin and Huxley equations. Dendritic trees, the propagation of action potentials, cable theory, compartmental models. Modelling synapses. Spiking neuron models and response variability: leaky integrator and integrate-and-fire type neuron models, spike time variability. Current topics in Neuroinformatics including (a) understanding of the neural code (b) Synaptic Plasticity. Bottom-up/top-down modeling of the brain: modeling of self-control behaviour as an example of top-down modeling. Modelling consciousness. Applications of Neuroinformatics; Neuroinformatics vs Bioinformatics.
Prerequisites: Linear Algebra, Differential Equations
Bibliography:

1. P. Dayan and L. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modelling of Neural Systems*, MIT Press, 2001.

2. D. Sterratt, B. Graham, A. Gilles and D. Willshaw, *Principles of Computational Modelling in Neuroscience*, Cambridge University Press, 2011.
3. W. Gerstner and W. M Kistler, *Spiking Neuron Models: Single Neurons, Populations and Plasticity*, Cambridge University Press, 2002.
4. C. Koch, *Biophysics of Computation: Information Processing in Single Neurons*, Oxford University Press, 1998.
5. E. M. Izhikevich, *Dynamical Systems in Neuroscience: the Geometry of Excitability and Bursting*, MIT Press, 2007.

Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory sessions (1.5 hours weekly).

Assessment: Final exam, midterm exam and laboratory exercises /oral presentations of selected research papaers.

CS668 Mechanical Vision

Type: Specialisation Course for PM
Restricted Choice for MCS, MIC, MIntS

Level: Graduate

Semester: Spring

Credit: 8 ECTS units

Instructor: C. Pattichis / Y. Chrysanthou

Objectives: The objective of this course is to understand the basic issues in mechanical vision and the major approaches that address them. Through the duration of the course, vision is treated as a process of inference from noisy and uncertain data in order to answer the question of how computers can understand the visual world of humans.

Content: Basic concepts and methodologies relating to the subject of Computer Vision. Image information, image processing, feature extraction. Image segmentation, clustering, multiple-image processing, case studies.

Prerequisites: -----------

Bibliography:

1. D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice-Hall, 2003.
2. R. Hartley and A. Zeisserman, *Multiple View Geometry,* Cambridge University Press, 2003.
3. C. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag, 2007.
4. O. Faugeras and Q.T. Luong, *Geometry of Multiple Images*, MIT Press, 2001.
5. B. Horn, *Robot Vision*, MIT Press, Cambridge, Massachusetts, 1986.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1 hour weekly).

Assessment: Final exam, midterm exam and homework (programming exercises).

CS670 Research Methodologies and Professional Practices in Computer Science

Type: MCS, MIC, MIntS, PM

Level: Graduate

Semester: Spring

Credit: 4 ECTS units

Instructor: ……………..

Objectives: Introduction to the methods and tools of Computer Science Research and technological culture. Familiarization with reading, reviewing and presenting of relevant literature. Technical writing. Literature surveying.

Content: Seminars/lectures in Computer Science and practice. Research or technical literature reviewing. Presentation of technical study.

Prerequisites: -------------

Bibliography:

1. Selected research articles from the international literature.
2. Course Presentation Slides (introductory and research).

Teaching methods: Lectures, research seminars and atomic assignments (summary of research seminars) and group study of a research subject under the supervision of a faculty member.

Assessment: Attendance and participation in lectures and a number of research seminars, written atomic studies, group study of a research subject and technical presentation of the group study. The course grade is Pass/Fail.

CS673 Algorithmic Game Theory

Type: Specialisation Course for MIC and Restricted Choice for MCS, MIntS, PM
Level: Graduate
Semester: -----------
Credit: 8 ECTS units
Instructor: M. Mavronicolas
Objectives: ----------
Content Strategic games. Pure and mixed strategies, utilities, best responses, equilibrium concepts. Pure and mixed Nash equilibria, their refinements and generalisations. Classical existence theorems of equilibria and their algorithmic aspects. Algorithms and complexity of equilibrium searching. The complexity classes PLS and PPAD and their relation to equilibrium computation. Bimatrix games and
algorithms to compute their approximate equilibria. The Price of Anarchy and its variants. Analysis of the Price of Anarchy for both general and specific games (e.g., selfish routing games, congestion games, security games). Applications to realistic cases (e.g., social networks, Internet formation).
Prerequisites: Undergraduate course equivalent to the CS211 (Theory of Computation and Complexity) and undergraduate course equivalent to the CS436 (Algorithms and Complexity)
Bibliography:

1. M. Mavronicolas and P. Spirakis, *Algorithmic Game Theory*, Springer, 2011, (book draft).
2. Selected research articles from the international literature_

Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory sessions (1 hour weekly).
Assessment: Final exam, midterm exam and homework (theoretical assignments).

CS674 Networks and System Security

Type: Specialization Course for MIC, PM and Restricted Choice for MCS, MIntS
Level: Graduate
Semester: ----------
Credit: 8 ECTS units
Instructor: A. Pitsillides, V. Vassiliou
Objectives: ----------
Content Introduction to security threats and attacks. Cryptographic and cryptanalysis techniques. Key exchange management (PKI). Network and Internet security protocols (IPSec, SSL/TLS). Identification and authentication standards (Kerberos, AAA). Systemsecurity (Firewalls, IDS). Specific threats on end-systems (viruses, worms, trojan horses, stack overflow, rootkits). Identification of security vulnerabilities in software and operating systems. Checking of networks and applications for vulnerabilities, introduction to computer systems forensics. Security policies. Security management, ethical and legal issues in system security.
Prerequisites: Introductory graduate course equivalent to CS606 (Computer Networks and the Internet)
Bibliography:

1. C. P. Pfleeger, S. L. Pfleeger, *Security in Computing*, Fourth Edition, Prentice Hall, 2006.
2. C. Kaufman, R. Perlman, and M. Speciner, *Network Security: PRIVATE communication in a PUBLIC world*, Second Edition, Prentice Hall, 2002.
3. M. Dowd, J. McDonald and J. Schuh, *The Art of Software Security Assessment*, Addison Wesley, 2006.

Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory sessions (1 hour weekly and 2 hours weekly, respectively).
Assessment: Final exam, midterm exam and homework (studies and/or laboratory assignments).

CS675 Web Services and Service Oriented Computing

Type: Specialisation Course for MIC, PM and Restricted Choice for MCS, MIntS
Level: Graduate
Semester: ----------
Credit: 8 ECTS units
Instructor: ----------
Objectives: Understanding of the principles and the design of Web services, the programming of Web services and the protocols used.
Content: Introductory concepts. Relationship and difference between services and other related formalisms (distributed systems, component-based systems, etc). Fundamental architectures and protocols (SOAP, WSDL, UDDI). Fundamental development platforms (J2EE, NET, etc).
Problems and challenges. Information 74odeling and representation (ontologies, RDF and OWL protocols, etc). Cooperative Information Systems and service composition.
Prerequisites: CS603
Bibliography:

1. M. Singh, M. Hunhs, *Service-Oriented Computing: Semantics, Processes, Agents*, Wiley.
2. T. Erl, *Service Oriented Architecture*, Prentice Hall.
3. Selected research articles from the international literature.

Teaching methods: Lectures (3 hours weekly) and Discussions/Presentations (1 hour weekly).
Assessment: Final exam, midterm exam and homework (studies and/or laboratory assignments).

CS679 Electronic Health

Type: Specialisation Course for MintS and PM and Restricted Choice for MCS and MIC
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: Ch. Schizas
Objectives: To introduce the student to the medical and clinical environment from the perspective of medical informatics and exploit the possibilities of using information technologies for modeling, prototyping, presenting and using the relevant data. To study and develop practical skills in building relevant intelligent information systems.
Content: Information retrieval from medical databases, data, medical records, live signals, and data mining using intelligent techniques. Study of application systems that are currently in use for managing medical data and suggest ways for better handling and building, medical knowledge bases, electronic health record, and decision support systems for the medical profession.
Prerequisites: --------
Bibliography:

1. J. H. van Bemmel and M. Musen (Edts), *Handbook of Medical Informatics*, Springer, 1997.
2. E.H. Shortliffe (Editor), L. M. Fagan, G. Wiederhold, L.E. Perreault, *Medical Informatics: Computer Applications in Health Care and Biomedicine*, Springer Verlag, Second, 2000.
3. L.Burke, B. Weill, *Information Technology for the health professionals*, Pentice Hall, 2000.

Teaching methods: Lectures (3 hours weekly) and Discussions/Presentations (1 hour weekly).
Assessment: Final exam, midterm exam and homework (studies and/or laboratory assignments).

CS699 Special Topics in Computer Science: Online Social Networks

Type: Restricted Choice for Ph.D. cndidates (and conditionally for MCS, MIC, MIntS, PM)
Level: Ph.D. and Advanced M.Sc.

Semester: Spring
Credit: 8 ECTS units
Instructor: M. Dikaiakos
Objectives: Survey of recent scientific literature of Online Social Networking Systems. Study of basic concepts, principles, techniques and scientific results that shape the analysis and evolution of Online Social Networking Systems.
Content: Introduction to Social Networks. Design Principles of Online Social Networking Systems. Overview of popular Online Social Networking Systems: Facebook, Twitter, LinkedIn. Social Networks analysis from a Complex Networks Perspective. Data Mining of Online Social Networks data. Modeling of Online Social Networks.
Prerequisites: (for M.Sc. students) CS646, CS660 or the consent of the instructor.
Bibliography:

1. Vakali and Jain (Eds), *New Directions in Web Data Management 1*, Springer, 2011.
2. Easly and Kleinberg, *Networks, Crowds and Markets*, Cambridge, 2010.
3. Selected articles from scientific journals and conferences.

Teaching methods: Lectures (3 hours weekly).
Assessment: Active participation in the class, paper assignements, authoring of scientific survey paper, semester team-project.
Language of Instruction: English.

Short Biographical Notes of Academic Staff

Chris Christodoulou, Associate Professor. Undergraduate studies at Queen Mary and Westfield College, University of London, UK (BEng in Electronic Engineering, 1991) and Birkbeck College, University of London, UK (BA in German, 2008). Graduate studies at the Kings College, University of London, UK (Ph.D. in Electronic Engineering, 1997). He has worked as a Postdoctoral Research Associate at the Kings College, University of London, UK (1995-97), and he has taught as a Lecturer at Birkbeck College, University of London, UK (1997-2005). His research interests include Computational and Cognitive Neuroscience as well as Neural Networks.

Yiorgos Chrysanthou, Associate Professor. Undergraduate and graduate studies at Queen Mary and Westfield College of University of London, UK (B.Sc. in Computer Science and Statistics, 1990, and Ph.D. in Computer Science, 1996). He has worked and taught at University College London as Research Fellow (1996-98) and Lecturer (1998-2001). He has been a Program Committee Chair for international conferences (VAST 2004, ACM VRST 2005 and ACM VRST 2006, ECMS 2008, MIG 2010) and has been the local or overall coordinator of many research projects funded through various sources. His research interests are in the general area of 3D Computer Graphics, including real-time rendering, reconstruction of urban environments, character simulation, with applications to serious games and cultural heritage.

Marios D. Dikaiakos, Professor. Undergraduate studies at the National Technical University of Athens (Dipl.-Ing. In Electrical Engineering, summa cum laude, 1988). Graduate studies at Princeton University (Ph.D. 1994, M.A. 1991 - both in Computer Science). He has worked and taught as Research Associate at the University of Washington, Seattle, USA (1994-1995), the University of Cyprus (visiting Assistant Professor 1996; Assistant Professor 1998-2004; Associate Professor 2004-2011; Professor from 2012), the University of Crete, Greece (Socrates/Erasmus visiting Professor, 10/2004), and Rutgers University, USA (visiting Research Professor, spring semester 2005). He also spent a pre-doctoral internship at the Paris Research Lab of Digital Equipment Corporation (1991). He established and is the Principal Investigator of the Laboratory for Internet Computing at the Department of Computer Science, University of Cyprus, where he directs several research projects funded by the European Union and the Research Promotion Foundation of Cyprus. Dr. Dikaiakos has served in the program committees of several international scientific conferences and has consulted with the European Commission and several organizations of European countries in evaluating research proposals, research projects, and academic work. His research interests include Network-Centric Computing with an emphasis on Grid Computing systems (Performance, Dependability, Information Services), Web Technologies (Search Engines, Web Characterization and Mining), and Mobile Computing (Application Protocols).

Yannis Dimopoulos, Associate Professor. Undergraduate and graduate studies at the Athens University of Economics (B.Sc. in Computer Science 1987, Ph.D. in Computer Science, 1992). He has worked as Researcher at the Max-Planck-Institute for Computer Science, Germany (1992-1994), University of Cyprus (1995-1996) and University of Freiburg, Germany (1996-1998). His research interests are in the area of Artificial Intelligence with emphasis on Knowledge Representation and Reasoning, Planning, Logic Programming, Constraint Satisfaction and Machine Learning.

Paraskevas Evripidou, Professor. Undergraduate studies at the Higher Technical Institute, Cyprus (H.N.D. in Electrical Engineering, 1981) and University of Southern California, USA (B.Sc. in Electrical Engineering, 1985). Graduate studies at the University of Southern California USA (M.Sc., 1986 and Ph.D., 1990, both in Computer Engineering). He has taught at the University of Southern California, USA (part-time faculty, 1989-1990) and at the Southern Methodist University, USA (Assistant Professor, 1990-1994). He has been the Program Co-Chair for the 8th Panhellenic Conference on Informatics (2001) and the Program Chair for the 3rd South Eastern European Conference on E-Commerce (2002). His research interests include Parallel Processing and Computer Architecture, Data Flow Systems, Functional Programming and Parallelizing Compilers, and Electronic Commerce.

Chryssis Georgiou, Associate Professor. Undergraduate studies at the University of Cyprus, Cyprus (B.Sc. in Mathematics, 1998). Graduate studies at the University of Connecticut, USA (M.Sc., 2002; Ph.D., 2003, both in Computer Science and Engineering). He has worked as a Teaching and Research Assistant at the University of Connecticut, USA (1998-2003) and as a Visiting Lecturer (2004), as a Lecturer (2005-2008) and as an Assistant Professor (2009-2013) at the University of Cyprus. His research interests span the Theory and Practice of Distributed and Parallel Computing; in particular, Fault-tolerance and Dependability, Algorithms and Complexity, Communication Networks and Paradigms, Distributed Atomic Storage, Algorithmic Game Theory and Dynamic Computing Environments. He has published in top journals and conference proceedings in his area of study and he has co-authored a book on Distributed Cooperative Computing. He served on Program Committees of top conferences in Distributed Computing and he is on the Steering Committee (2008-2010, 2010-2012) of the International Symposium on Distributed Computing (DISC).

Antonis C. Kakas, Professor. Undergraduate studies at the Imperial College, University of London, UK (B.Sc. in Mathematics, 1980). Graduate studies at the Imperial College, University of London, UK (M.Sc. in Computer Science, 1987 and Ph.D. in Theoretical Physics, 1984). He has worked as Research Fellow at Kings College, UK (1984-1986), at the University of Zurich, Switzerland (1987-1988), and at Imperial College, UK (1988-1992). He is a member of the editorial board of the international journals *AI Communications*, *Journal of Applied Logic* and *Journal of Theory and Applications of Logic Programming.* His research interests include Knowledge Representation and Reasoning in Artificial Intelligence, with applications to Machine Learning, Planning, Diagnosis, Congitive Agents Computational Bioscience.

Georgia Kapitsaki, Lecturer. Undergraduate studies at the National Technical University of Athens (Dipl. Electrical and Computer Engineering, 2005). Graduate studies at the National Technical University of Athens (M.Sc., 2008, co-organised, and Ph.D., 2009, School of Electrical and Computer Engineering). She has worked as a Research Associate at the National Technical University of Athens (2005-2009), as a Laboratory Assistant at the Technical Institute of Pireaus (2007-2009), as well as a software and telecommunications engineer in Germany (2005, 2009-2010). She has also worked as a visiting researcher at the Otto von Guericke University of Magdeburg (2013) and has participated in the organization and in the program committees of international conferences (e.g., WISE 2013, WEBIST 2012). Her research interests include Open Source Software, Mobile Internet and Context-aware services, Model-driven engineering, Aspect-oriented programming, Web 2.0 technologies and Privacy protection.

Elpida Keravnou-Papailiou, Professor. Undergraduate studies at Brunel University, UK (B. Tech. in Computer Science, 1982). Graduate studies at Brunel University, UK (Ph.D. in Cybernetics, 1985). She has taught at University College London, UK (Lecturer, 1985-1991; Senior Lecturer, 1991-1992; Director of the M.Sc. Program in Computer Science). She is Associate Editor of the journal *Artificial Intelligence in Medicine* (AIM) and she is on the Editorial Board of the journal *Systems Research and Information Science.* She has been the Program Committee Chair for the 6th Artificial Intelligence in Medicine Europe (AIME 1997) Conference. Her research interests include Artificial Intelligence, Expert Systems and their Applications in Medicine, Diagnostic Models, Temporal Reasoning, Abductive Reasoning, Intelligent Data Analysis, and Hybrid Decision-Support Architectures. She served as Vice-Rector for Academic Affairs at the University of Cyprus (2002-2006) and as President of the Governing Board of the Cyprus University of Technology (2009-2010). She is Vice-Chair of the Evaluation Committee for Private Universities and a Member of the National Team of Bologna Experts.

Marios Mavronicolas, Professor and Vice-Rector for International Affairs, Finance and Administration. Undergraduate studies at the National Technical University of Athens, Greece (Diploma in Electrical Engineering, 1985; Summa cum Laude). Graduate studies at Harvard University, USA (M.A., 1988; Ph.D., 1992, both in Computer Science). He has taught at University of Crete, Greece as Visiting Assistant Professor (1992-93), and at the University of Connecticut, USA as Assistant Professor (1999). He is on the Editorial Board of the journals *Theoretical Computer Science*, *Journal of Interconnection Networks* and *Networks*, and he is the Distributed Computing Column Editor of the *Bulletin of the European Association for Theoretical Computer Science.* He has been the Program Committee Chair for the 11th International Workshop on Distributed Algorithms (WDAG 1997), the Program Committee Co-Chair for the 2nd International Workshop on Internet and Network Economics (WINE 2006), and the Program Committee Chair for the 2nd International Symposium on Algorithmic Game Theory (SAGT 2009). He is the Chairman of the Cyprus Council for the Evaluation of Higher Degrees (ΚΥ.Σ.Α.Τ.Σ.) and the Vice-Chair of the Assembly of Rectors of Cypriot Universities. His research interests span the area of Algorithms and Complexity, with emphasis on Algorithimic Game Theory, Distributed and Parallel Computing, Networking and the Internet.

George Pallis, Lecturer**.** He received his BSc (2001) and Ph.D. (2006) degree in Department of Informatics of Aristotle University of Thessaloniki (Greece). Currently, he is lecturer at the Computer Science Department, University of Cyprus. Previously, he was visiting lecturer and Marie Curie Fellow at the same department. His research interests include Content Distribution Networks, Cloud computing, Web data caching, information retrieval, online social networking, vehicular networking and Web data clustering. He has published in top journals and conference proceedings in his area of study and he has co-edited a book on Web data management. He is member of the editorial board in the IEEE Internet Computing magazine and the International Journal on Internet and Distributed Computing Systems.

George A. Papadopoulos, Professor. Undergraduate studies at Aston University, UK (B.Sc. in Computer Science and Mathematics, 1982). Graduate studies at Aston University, UK (M.Sc. in Computer Science, 1983) and at the University of East Anglia, UK (Ph.D. in Computer Science, 1989). He has worked as a Researcher at NCR Demokritos, Greece (1981-1982), as a Research Assistant at the Aristotelian University of Thessaloniki, Greece (1983-1985) and as a Research Associate at the University of East Anglia, UK (1987-1990). He is a recipient of an 1995 ERCIM Fellowship visiting GMD in Germany and CWI in the Netherlands. His research interests include Advanced Software Engineering, Parallel and Distributed Systems, Adaptive and Context Aware Systems, e-Health and e-Learning.

Constantinos S. Pattichis. Professor. Undergraduate studies at the Higher Technical Institute (H.N.D. in Electrical Engineering, 1979), and at the University of New Brunswick, Canada (B.Sc. in Electrical Engineering, 1983). Graduate studies at the University of Texas at Austin, USA (M.Sc. in Biomedical Engineering, 1984), at the University of Newcastle Upon Tyne, UK (M.Sc. in Neurology, 1988), and at the Queen Mary College, University of London, UK (Ph.D. in Electronic Engineering, 1992). He has worked as a Research Associate (1986-87) and as a Senior Research Scientist (1990-92) at the Cyprus Institute of Neurology and Genetics and has taught as Visiting Lecturer at the Higher Technical Institute of Cyprus (1990-1992). He is Associate Editor of the scientific journals *IEEE Transactions on Information Technology in Biomedicine* and the *IEEE Transactions on Neural Networks*. He served as Chairman of the IEEE Cyprus Section, the Cyprus Association of Medical Physics and Biomedical Engineering and of the conferences Medical and Biological Engineering and Computing (1998) and IEEE MELECON 2000. His research interests include the development of Computational Intelligent Systems (based on Neural Networks and Cenetic Algorithms), Data Mining in Medicine, Medical Imaging Analysis and Biosignal Analysis, and Health Telematics.

Anna Philippou, Associate Professor. Undergraduate studies at the University of Oxford, UK (B.A. in Mathematics and Computation, 1992). Graduate studies at the University of Warwick, UK (M.Sc. in Parallel Computers and Computation, 1993; PhD in Computer Science, 1996). She has worked as a Teaching Assistant at the University of Warwick, UK (1993-1996) and as a Postdoctoral Research Fellow at the University of Pennsylvania, USA (1997-1998). Her research interests include Concurrency Theory and its Applications, Specification and Verification of Concurrent Systems, Formal Methods for Safety-Critical Systems and Algorithmic Game Theory.

Andreas Pitsillides, Professor. Undergraduate studies at University of Manchester, Institute of Science and Technology (UMIST), UK (B.Sc. in Electrical and Electronic Engineering, 1980). Graduate studies at Swinburne University of Technology, Australia (Ph.D. in High Speed Multimedia Networks, 1993). He has worked in the industry (1980-1986), and has taught at Swinburne University of Technology, Australia (Lecturer 1987-1991, Senior Lecturer, 1992-1994; Foundation Associate Director of the Laboratory for Telecommunications Research, 1992-1995). His research interests include fixed and wireless Networks, flow and congestion control, resource allocation and radio resource management, Internet technologies and their application in Mobile e-Services and recently the Internet/Web of Things. He established the Networks Research Laboratory (NetRL) at the Department of Computer Science, University of Cyprus, where he leads several research projects, funded by the European Union and the Research Promotion Foundation of Cyprus. He has a particular interest in adapting tools from various fields of applied mathematics such as adaptive non-linear control theory, game theory, computational intelligence, and recently nature inspired techniques, to solve problems in communication networks. He is a member of the International Federation of Automatic Control (IFAC) Technical Committee TC 1.5 on Networked Systems and TC 7.3 on Transportation Systems, and of the International Federation of Information Processing (IFIP) working group WG 6.3: Performance of Communications Systems. He is also a member of the Editorial Board of Computer Networks (COMNET) Journal and the International Journal of Handheld Computing Research (IJHCR).

George Samaras, Professor. Undergraduate studies at the University of Athens, Greece (B.Sc. in Mathematics, 1982). Graduate studies at Rensselaer Polytechnic Institute, USA (M.Sc. and Ph.D. in Computer Science, 1982 and 1989, respectively). He has worked in the Applied Research Program of IBM’s Communications and Networks Centre at Research Triangle Park, North Carolina, USA (1990-1993), and has taught as Visiting Assistant Professor at the University of North Carolina at Chapel Hill, USA (1990-1993). He has also been a member of IBM’s International Standards Committees for issues related to Distributed Transaction Processing (OSI/TP, XOPEN, OMG). His research interests include Mobile Computing, Relational and Object-Oriented Databases, Object-Oriented Programming, and Commit Protocols for Distributed Transaction Processing.

Yanos Sazeides, Assistant Professor. Undergraduate studies in Computer Engineering at Oakland University (B.Sc.,1991). Graduate studies at Cornell University (M. Eng. in Electrical Engineering, 1992) and at University of Wisconsin-Madison (Ph. D. in Electrical Engineering, 1999). He has worked at research/development laboratories for high performance processors at HP, Compaq and Intel. He has taught at the University of Cyprus as Visiting Lecturer (2000-2001). His research interests include Computer Architecture with special emphasis on High Performance Processors, Program Behavior, Execution Paradigms and Power Aware Microarchitectures.

Christos N. Schizas, Professor. Undergraduate studies at the Queen Mary College, University of London, UK (B.Sc. in Electronic Engineering, 1978). Graduate studies at the University of Indianapolis, USA (M.B.A., 1988) and at the Queen Mary College, University of London, UK (Ph.D. in Systems Theory, 1981). He received the 1979 William Lincoln Shelley award from the University of London of excellence in research, and a Fullbright fellowship for collaborative research in the USA in 1993. He was a Postdoctoral Fellow at the University of London (1980-1983), and was Professor of Computer Information Systems at the University of Indianapolis (1989-1991). Since 1991 he has been with the Department of Computer Science, University of Cyprus. He served as Vice Rector of the University of Cyprus (2002-2006). His research interests include computational intelligence, medical informatics, eHealth, diagnostic and prognostic systems, system modeling and identification of brain activity. He edited conference proceedings and served as associate editor of the journal Technology and Health Care, area editor of the journal, IEEE Transactions on Information Technology in Biomedicine, and member of the editorial board of the journal, Intelligent Systems. He is the founder of the Computational Intelligence laboratory of the University of Cyprus. He has taken part in European Commission initiatives for promoting the Information Society, especially the Euro-Mediterranean partnership.

Pedro Trancoso, Associate Professor. Undergraduate studies at the Technical University of Lisbon, Portugal (B.Sc. in Computer and Electronics Engineering, 1992). Graduate studies at the University of Illinois at Urbana-Champaign, U.S.A. (MSc. and Ph. D. in Computer Science, 1995 and 1998, respectively, Thesis: Optimizing Memory-Resident Decision Support System Workloads for Cache Memories). He has worked at IBM T.J. Watson Research Center, U.S.A. as a Researcher (1997), at the University of Illinois at Urbana-Champaign, U.S.A. as a Visiting Scholar (2000) and at Intercollege Limassol as an Assistant Professor (1998-2001). His research interests focus on Computer Architecture with a focus on Memory Hierarchy and Technologies, Architecture-Aware Optimizations for Database Workloads, Power-Aware Optimizations, Multi-core Architectures,, Graphics Processor Architectures, Performance Evaluation and Benchmarking.

Vasos Vassiliou, Assistant Professor. Undergraduate studies at the Higher Technical Institute (H.N.D. in Electrical Engineering, 1993) and the University of South Florida (B.Sc. in Electrical Engineering, 1997). Graduate studies at the Georgia Institute of Technology, USA (M.Sc., 1999 and Ph.D., 2002, both in Electrical and Computer Engineering). He has worked as Research and Teaching Assistant at the Georgia Institute of Technology, USA. He has taught as a Visiting Lecturer at the Department of Computer Science, University of Cyprus (2004-2005) and as an Assistant Professor at the Intercollege, Cyprus (2002- 2004). His research interests include High Speed Network Architectures, Mobile Computing, Wireless Communications, Quality of Service (QoS) and Traffic Engineering for Computer and Telecommunication Networks.

Demetris Zeinalipour, Assistant Professor. Undergraduate studies at the University of Cyprus, Cyprus (B.Sc. in Computer Science, 2000). Graduate studies at the University of California - Riverside, USA (M.Sc., 2003; Ph.D., 2005, both in Computer Science and Engineering). He has worked as a Teaching and Research Assistant at the University of California - Riverside, USA (2000-2005), a Visiting Lecturer at the University of Cyprus (2005-2007), a Lecturer at the Open University of Cyprus (2007-2009) and a Lecturer at the University of Cyprus (2009-2012). He has also been a Visiting Researcher at the Network Intelligence Lab of Akamai Technologies, in Cambridge USA, during 2004. He has founded and directs the Data Management Systems Laboratory (DMSL) at the

Department of Computer Science at the University of Cyprus. He has been a PC Chair for the 11th Intl. Conference on Mobile Data Management (IEEE MDM'10), the PC Chair for the 7th Intl. Workshop on Data Management in Sensor Networks (with VLDB'10) and the PC Chair for the 8th Intl. ACM Workshop on Data Engineering for Wireless and Mobile Access (ACM MobiDE'09), collocated with ACM SIGMOD'10. His research focuses on Data Management in Systems and Networks, in particular, Distributed Query Processing, Storage and Retrieval Methods for Sensor, Smartphone and Peer-to-Peer Systems, Mobile and Network Data Management.

Appendix A: Rules for Diploma Projects

1. GENERAL

A *Diploma Project* is prepared by a final-year student, usually during the seventh and eighth semesters of study in accordance with the Programme of Studies of the Department. The part of Diploma Project prepared in the first semester is called *Diploma Project I* and the part corresponding to the second semester *Diploma Project II*. These two parts are prepared and completed in *EPL400 - Diploma Project I* and *EPL401 – Diploma Project II*, respectively.

A Diploma Project corresponds to seventeen and a half (17.5) *Credit Units ECTS*, which are credited to the student upon successful completion.

The student registers for the Diploma Project with the approval of the Academic Advisor.

The Department Council appoints a member of the Faculty of the Department, who ensures and coordinates the entire process of developing and assessing the Diploma Projects. This Faculty member is known as the *Diploma Projects Coordinator*.

All the forms and guides mentioned in this section are digitally available on the website of the Department. The Department maintains a *Digital Library for the Diploma Projects* for archiving purposes.

2. SUBSTANCE, FORM AND EVALUATION CRITERIA

2.1 Substance

Each *Diploma Project* must contain sufficient information that reflects the student's initiative, independent study and productivity (originality, in the broad sense).

The Diploma Project may be of theoretical or practical nature or a combination of both. A Diploma Project may include, for example, an application of existing techniques, extension of known methods in the theory, software, hardware or applications areas, development of a prototype system, address theoretical problems, a survey review or study of a theoretical or practical area, etc.

2.2. Form

A *Diploma Project* must be a comprehensive document structured in chapters and must follow the rules of the technical guidance report called *Standards for Preparation the Diploma Project*.

The Diploma Project should include an introduction to the subject, an analysis of the importance of project, a description of the related work, a review the work in the area of the topic, a description of the methodology used, listing, classification and evaluation of the results of the work and finally conclusions and suggestions for possible extension of the work.

Diploma Projects where software was created and/or used should also include the code of the software in a specific Annex, a description and analysis of the software a separate section, and instructions for the use of the software. The code of such software may not be used as the Thesis document.

2.3 Evaluation Criteria

The main criteria for the evaluation of the Diploma Project are the following:

[a] Quality of work (e.g. accuracy and completeness of analysis, appropriateness of methodology, validity of theoretical results, software quality, implementation, consistency of material presented and association of ideas).

[b] Degree of the objectives achieved of the Diploma Project.

[c] Degree of understanding by students of the area of the Diploma Project topic.

[d] Quality of the written language of Diploma Project (e.g. structure and organization, clarity, ease of reading and understanding).

[e] Quality of Presentation of the Diploma Project. (e.g. oral speech, the adequacy and suitability of multimedia used (such as slides), proper utilization of the allowed time of presentation, and most importantly to point the contribution of Diploma Project through the presentation).

3. PROCEDURE

3.1 Preparation

3.1.1 Submission and Announcement of Topics

Each member of the Faculty submits to the Department, end of March of each year, a number of Diploma Project topics greater than the ratio of the number of students per Faculty member. Each Faculty member reserves the right not to supervise a number of Diploma Projects larger than the nearest integer above the ratio of students per Faculty member.

Each topic has a title. It is expected that a brief description of each topic is made available to the students by the proposing faculty. (It is also expected that the faculty members will update and renew appropriately the description of the projects on their personal websites).

The *Diploma Projects Coordinator establishes a list of the submitted Diploma Project topics*, and announces the Diploma Project list to the faculty members of the Department and to the final-year students approximately in middle of April.

3.1.2 Choice of Topic

Each student should choose a Diploma Project topic. To this end, the students shall discuss with the corresponding faculty members that have offered the topics, in meetings during a specified period of time. Throughout the process of submission of preferred topics, the students obtain the agreement of the faculty member for the supervision of the corresponding project. This Faculty member will be the *Diploma Project Supervisor* for that student.

With the selected choice of the topics, the student completes a special *Registration Form*, available electronically on the website of the Department which includes the title, description, and any specialized software / hardware or other resources necessary for the preparation of the Diploma Project. The *Registration Form* is signed by the Faculty member, the Academic advisor of the student and it is deposited to the Department no later than the period of registration of students for the semester the student prepare the *first part of the Diploma Project*.

In case of not being able to select topic, the student returns the *Registration Form* to the Department, signed by him/her and the Academic Advisor. In this case, with the filing of the form, the *Diploma Project Coordinator* contacts the Faculty members of the Department, to ensure that every student who submitted the registration form without topic to select a topic before the end of the submission period. In case this is not possible, a special meeting of the Department Council is called to assure that each student is given a topic. The student must choose the topic selected by the Council meeting

Concurrently with the submission of the registration form, the student must register for the EPL 400 course.

3.1.3 Change of Subject

Changing topic of the Diploma Project (with the same or another supervisor) is possible within the first three (3) weeks of the semester the student enrolled for the *first part of the Diploma Project* (Diploma Project I). To this end, using the same procedure that was followed for the submission of the original Diploma Project topic, the student fills in the *Special Registration* Form to the Department in order for the *coordinator of Diploma Project* to approve it.

3.2 Preparation

3.2.1 Supervision

The supervision of a student who prepares a Diploma Project is the responsibility of the Diploma Project Supervisor. The monitoring and controlling of the Diploma Project progress is done through regular meetings between the student and the supervisor. In some cases a co-supervisor may be indicated. The supervisor must be a PhD holder and may either come from the Department of Computer Science or from an institute or organization outside the Department.

3.2.2 Interim Evaluation

During the examination period of the first semester in which the student enrolled in the Diploma Project, the student submits to the *Supervisor* a brief Progress Report. Following the submission of the Progress Report, the Supervisor shall submit a written assessment of the progress of the student to the Department, which is also sent to the student. The possible grades for this assessment are Success or Failure. The grade of the Diploma Project I is the student's grade for the course EPL400 which has been enrolled for the current semester.

In the second semester the student can enroll in the Diploma Project II, EPL401, only if the student has succeeded with EPL400 course. In case of a Failure grade results in the student enrolling again in the first part of the Diploma Project (Diploma Project I: EPL400) in the same or different topic (with the same or different supervisor).

The coordinator of the Diploma Projects submits the student's grade for the Diploma Project I to the Academic Affairs and Student Welfare in time.

3.3 Evaluation

The final evaluation of the Diploma Project takes place towards the end of the semester where the student is registered for *Diploma Project II*.

3.3.1 Second Assessor

The Diploma Project is assessed by the *supervisor*, and the co-supervisor (if assigned), together with another member of the Faculty, known as the *Second Assessor*. The Diploma Project Coordinator with the collaboration of the Supervisors, publishes a *list of Assessors and Program of Diploma Project presentations*.

In special cases, the Coordinator of the Diploma Project may approve as a *Second Assessor* a Visiting Faculty Member of the Department or a Faculty member from other Department of the University or/and other University in Cyprus or abroad. The approval may be justified in cases of close affinity of the Second Assessor Diploma on the subject of the Diploma Project.

3.3.2 Presentations

The Diploma Project Coordinator publishes the *Program of Diploma Project presentations* for three days during the week immediately following the examination period. The Program of Diploma Project presentations must be made publicly available to all faculty members of the department and the presentations are open to the public.

Each student who is expected to complete its Diploma Project II based on the Academic Advisor judgment must be listed in the *Diploma Project Presentation list*. These students present their Diploma Project in public in front of their *Academic Advisor* and their *Second Assessor*.

The students who have been excluded from the *Program of the Presentations* of the diploma Project receive a grade equal to *CD (Continuation of Diploma Project)* and must continue working on the same Diploma Project so in the immediate following semester to complete their Diploma with success. Therefore, these students must reregister in the same course, *EPL401*.

3.3.3 Grading

After the presentation the *Supervisor*, in consultation with the *Second Assessor*, submits the grade for the Diploma Project with written justification comments, in a specific *Assessment* Form, and in accordance with the procedure for filing course grades. The Assessment Form of the Diploma Project contains different *evaluation criteria* that need to be filled with numerical grades.

A *Failure* grade results in the student enrolling again in the first part of the Diploma Project (*Diploma Project I*) in a different topic with a different *supervisor*. A Success grade in Diploma Project receives a numerical grade according to the Rules Studies.

The grade of the Diploma Project is the grade for the courses *EPL401* which is enrolled.

The *Coordinator of the Diploma* Project handles the cases where there is a disagreement on the grade for the Diploma Project between the *Supervisor* and the *second assessor*.

3.3.4 Submit in Digital Format

Within ten days after the presentation and evaluation of the Diploma Project, the student shall submit to the Department its Diploma project in digital form. Failure to submit timely may result in delay of the student's graduation which may cause the graduation of the student impossible in the current semester.

3.3.5 Final Grade Submission

The coordinator of the Diploma Project submits the student's grade for the Diploma Project to the Academic Affairs and Student Welfare in time.

Appendix B: Rules for Graduate Studies

The Department of Computer Science offers *Graduate Degree Programs* that lead to *Master in Advanced Information Technologies* and *PhD* in Computer Science. The programs are published at the Department's Guidebook. The *Graduate Program Committee* coordinates the programs based on the General Rules of the University of Cyprus.

[1] Admission Requirements

I. 1. The department announces once a year the maximum number of graduate students it can accept for the upcoming academic year starting in September. Applications arrived after the submission deadline is accepted only in case the number of graduate students is not met.

2. The applicants must submit an application, a CV, official transcripts from all colleges/universities attended, a personal statement about his/her goals and interests, and two letters of recommendations (possibly by faculty members of their college or university, mailed directly to the department). It is not necessary for the applicant to hold an undergrad or college degree at the time he submits his application. However, the applicant must get / have a degree in Computer Science or related field of study before he joins the graduate program. The average grade (GPA) of the applicant should be 6.5/10 (or equivalent to that) and the degree should be from an accredited college / university (as it is defined by the General Rules of University of Cyprus).

3. Applications are examined by the Graduate Program Committee. The Graduate Program Committee holds the right, based on its judgment, to call the applicant for personal interview or ask more information. The Graduate Program Committee writes an evaluation and admission report and submits it to the Department Council for approval. The Department Council holds the right to accept fewer students than it had announced. The department submits its report to the school it belongs to.

II. The department assigns an Academic Advisor to every new graduate student. To a PhD student the department assigns a Research Advisor. The Research Advisor is assigned by the Department Council, after suggestion of the Graduate Program Committee, and an agreement between the student and a faculty member. The Research Advisor watches over the research or any other work of the student and provides any necessary guidance.

III. For students whose undergraduate degree was not related to Computer Science there is a possibility to be asked to take some undergraduate courses offered by the department. The student must pass these undergraduate courses in order to continue the graduate program.

[2] Master Degree

I. To get a Master Degree in Computer Science, each student must complete successfully courses (a total of 60 ECTS units) from the Graduate Program and complete a Master Thesis of 30 ECTS units, under the supervision of a Research Advisor. The assignment of a Research Advisor is based on the rules of the University Senate and is done before the student submits the Thesis Proposal. The student can transfer up to 15 ECTS units from courses completed successfully from similar graduate programs. Theses completed at other graduate schools are not transferable.

II. The Thesis must deal with a research topic or a technical issue. It must be of some original contribution or show a thorough and clear understanding of some special topic. Full-time students normally complete the thesis in 6 months study time. When the student has completed successfully courses totaling to 30 ECTS units, he can submit a Form (signed also by his Research Advisor) stating his Master Thesis topic.

III. The Master Thesis is submitted at the department and defended within the time period decided by the Department Council.

1. When a Master Thesis is submitted the Chair of the Department appoints a Thesis

Examination Committee of three members. The head of the committee is the student's Research Advisor. It is possible a member of this committee to be faculty member of other department. The Committee can have also external members who do not hold a faculty position. However, the membership must be approved by the Department Council and the external member must hold a PhD Degree or must have a reputation in the field of study.

2. The Master Thesis is defended in a presentation before the Thesis Examination Committee. The head of the Thesis Examination Committee is responsible for the procedures that must be followed during the defense.

3. The Thesis Examination Committee can accept (even with conditions) or reject a Master Thesis. The Thesis Examination Committee writes its decision in a Master Thesis Evaluation Form and submits it to the Department Council for approval. When the Thesis is accepted the department informs the Student Affairs Office for the graduation procedure of the student. When the thesis is rejected, the student can follow the suggestions of the committee and resubmit it for the first and the last time. The submission and thesis defense procedure is followed from the beginning.

IV. The minimum graduation time to get a Master Degree is three semesters and the maximum graduation time is eight semesters.

[3] PhD Degree

I. The basic requirements to get a PhD Degree are:

1. (a) Successful completion of at least 60 ECTS units of graduate level courses. A student with a Master or equivalent degree is partially or fully exempt from this requirement.

   (b) Successful completion of the Comprehensive Examination which must be taken no later than the fifth semester of his/her studies. The student submits to the department a request to take the Comprehensive Examination.

2. The Department assigns a Research Advisor to the student that has completed the ECTS units mentioned above.

3. The structure and the subjects of the Comprehensive Examination are decided by the Department. The Comprehensive Examination is general and attempts to measure the student ability to complete the degree. The topics that are examined are Theory, Software, Hardware, and Applications.

4. The Comprehensive Examination is presented before the Comprehensive Examination committee which is elected by the Postgraduate Committee of the department after it has been requested by the student's academic research supervisor. The formation of the Comprehensive Examination committee must be approved by the Department committee. The academic research advisor of the student is the chair of the Comprehensive Examination committee.

5. The Comprehensive Examination consists of three phases. The student should obtain a passing grade in all of them to ensure success in the examination. The three phases are the following:

   i. Submit to the Comprehensive Examination Committee a written literature review on the student's Research area. The review should have the length and the quality of a published survey article and must demonstrate the student's adequate knowledge and understanding of the topic under investigation and the open problems. To prepare the review, the Research Supervisor may give the student an indicative bibliography. The review is graded by the members of the Comprehensive Examination Committee with a Pass / Fail. When a review receives a passing grade, the student may proceed to an oral examination.
   ii. The student should make an oral presentation of his/her survey to the committee. The presentation lasts 50-60 minutes, including questions addressed by the committee and the audience. The presentation is open to the public and should be announced to the members of the department and the University.
   iii. The oral examination of the student, which is made by the members of committee, is closed to the public. The purpose of the examination is to further investigate whether

the candidate has the skills and capability to conduct Doctoral-level research work in Computer Science.

6. The Comprehensive Examination Committee submits for approval to the department a list of students that have succeeded in the examination.

7. A student that fails to pass the first comprehensive examination must take the examination the next time it is offered. A second fail of the student in the Comprehensive Examination disqualifies the student from the PhD program of the department.

8. Second failure in the comprehensive examination implies exclusion from the PhD candidature in the Department.

II. 1. The Post Graduate Program Committee taking into consideration the suggestions of the Research Supervisor appoints a three member Research Committee for the PhD candidate. The chair of the Research Committee is the Research Supervisor of a student. At most, one member of the Research Committee may be a faculty member of another department.

2. The student is allowed to change his/her Research Supervisor. He/She must submit for approval by the Department Board a request explaining in detail the reasons he/she wishes to change supervisor.

III. 1. When the student has passed the Comprehensive Examination he/she submits a written Doctoral Thesis Proposal to his/her Research Committee. The proposal is also presented orally before the Committee members.

a. The Doctoral Thesis Proposal must have the following structure:

1. Introduction
2. Motivation
3. Problem Statement – Hypothesis
4. Approach
5. Roadmap
6. Related Work
7. Preliminary Work
8. Work to be done
9. Timeline
10. Future work

b. The Research Committee must examine the Doctoral Thesis Proposal before the end of the next semester. The proposal might be accepted or recommended for resubmission. The final acceptance of the proposal must be given before the start of the seventh year of the student study time. Otherwise, the student case (whether to continue or to terminate his PhD candidacy) is discussed at the Department Council.

IV. 1. Every Doctoral student must submit to the Research Supervisor an annual progress report. This report must be submitted before the end of each academic year.

2.The annual progress report is graded by the Coordinator of the Graduate Studies and the Research advisor of the student with a Pass/Fail. The grade is submitted to Student Affairs and is marked in the student's records to which level the student progress in his/her studies (research, writing).

V. 1. Every PhD candidate completes a Dissertation which must be an original and important contribution to the field of study.

2. The written format of the Dissertation is described in the Department's printed forms. The Dissertation can be submitted no sooner than four semesters before student's admission and completing successfully his Comprehensive Examination.

VI. The student submits to the department six (6) copies of the Dissertation and a request (signed also by his Research Advisor) for the formation of a Dissertation Examination Committee. After this procedure, the graduate Program Committee, with suggestions from the Research Advisor, appoints an Examination Committee (based on the rules of University's Senate). The Graduate Program Committee assures a copy of the dissertation is sent to the members of the Examination Committee. The dissertation is defended by the candidate before the Examination Committee. Note that the three faculty members of the Department are usually the three members of the Research Committee.

VII. The Chair of the Examination Committee decides the date of the Dissertation Defense. The Defense must take place within three months from the Dissertation submission. All members of the Examination Committee must be present at the Defense. The defense procedure is based on the rules of the University’s Senate.

1. The approval of at least four members of the Examination Committee is required to award the degree. In this case, the Examination Committee has the right to request for changes or additions, which considers as necessary. The procedure for monitoring the changes or additions is decided by the Examination Committee and is written clearly in the Evaluation Report.

2. The Examination Committee hands over to the Department Chair a written Evaluation Report of the Dissertation and of the Doctoral Candidate in general, together with its recommendations, in a printed form. The Chair verifies that all the rules and procedures have been correctly followed and hands in the report to the Senate.

VI. A maximum of eight (8) academic years are allowed for the completion of the Doctoral Program.

Approved by the Senate, 149th Meeting, 22/5/2002

Appendix C: Specifications for Preparation of the Master Thesis

Abstract. An abstract is *required*. The body of the abstract may not exceed 400 words in length. Please see the sample abstract page for the format.

Minimum Margins. The minimum acceptable margins for all pages of the thesis and the abstract are 3.8 centimeters (1.5 inches) on left and 2.5 centimeters (1 inch) on the top, bottom, and right.

Paper Requirement. All pages of the thesis must be printed on 21 x 29.7 centimeter (8.27 x 11.69 inch) *white* paper. This is the regular A4 paper format.

Font and Point Size. Recommended fonts include Arial, Times New Roman, and Helvetica with a point size of either 11 or 12 (preferably 11).

Printing. Either laser printing or photocopying of high quality is acceptable. Inkjet printing is ***not*** acceptable as it is water soluble. Only *one side* of the paper is to be used for printing. Printing should be consistently clear and dark.

Spacing. The text of the thesis should be *double spaced*. Long quotations, footnotes, appendices, and references may be single spaced.

Color. It is recommended that you ***not*** rely on color to convey information (e.g., use symbols or labels rather than colors to differentiate the lines on a graph, or use stripes and cross-hatching instead of colors to distinguish areas on a map).

Photographs and Graphics. Photographs and graphics in the dissertation should be printed or photocopied directly on the paper as high quality black and white images. Scanned images must print clearly. If color must be used, ***only*** color laser or color photocopy printing is acceptable.

Use of materials copyrighted by others. IMPORTANT: Any material included that goes beyond "fair use" requires written permission of the copyright owner. It may be useful to include these in the thesis as an appendix.

Pagination. Preliminary pages (i.e., the approval page, acknowledgments, table of contents, and the like) are to be numbered consecutively using lower case *Roman numerals*. All pages of the text, appendices (if any), and references must be numbered consecutively using *Arabic* numerals. The page number of the first page of each Chapter must be centered on the bottom of the page. In all other pages, the page number must be placed on the top of the page, aligned to the right.

Landscape pages. The top of a landscape page should be at the left margin and the bottom at the right margin. The page number is to be in the same relative position as on the portrait pages. An easy way to apply page numbers to landscape pages is to run them through the printer twice – once for the text, table, or figure (landscape orientation) and once for the page number (portrait orientation).

Sequence of the main components of the dissertation. The appropriate order of the major sections of the thesis follows: the title page, the abstract, the copyright page (if needed), the approval page, acknowledgments, table of contents, the text, bibliography/references, and appendices (if any).

Bibliography/References. The ACM (http://www.acm.org/pubs) or the IEEE (http://standards.ieee.org/resources) reference style should be followed.

Footnotes and Endnotes No specific rules. The format that is prescribed by your advisory committee should be followed.

* * *

It is recommended that you use your full legal name on the abstract, the title page, the copyright page (if appropriate), and on the approval page. Make certain that your name appears exactly the same way in all places.

Take a moment to check that all pages in all copies of your thesis are accounted for and in the proper order before submitting final copies to the Departmental Secretary. *One hardcopy* and *an electronic version* (in pdf format) of the thesis should be submitted to the Departmental Secretary.

* * *

For students that plan to write their thesis using the Latex authoring language, they can download the style file “ucythesis.sty” that creates the required Master thesis format automatically from http://www.cs.ucy.ac.cy/~chryssis/master-specs.html.

Appendix D: Specifications for Preparation of the Doctoral Dissertation

Abstract. An abstract is *required*. The body of the abstract may not exceed 400 words in length. See the sample abstract page for the format.

Minimum Margins. The minimum acceptable margins for all pages of the dissertation and the abstract are 3.8 centimeters (1.5 inches) on left and 2.5 centimeters (1 inch) on the top, bottom, and right.

Paper Requirement. All pages of the dissertation must be printed on 21 x 29.7 centimeter (8**.**27 x 11.69 inch) *white* paper that it is *at least 25% cotton and 20lbs. weight*. This is a special A4 paper format that can be obtained at bookstores.

Font and Point Size. Recommended fonts include Arial, Times New Roman, and Helvetica with a point size of either 11 or 12 (preferably 11).

Printing. Either laser printing or photocopying of high quality is acceptable. Inkjet printing is ***not*** acceptable as it is water soluble. Only *one side* of the paper is to be used for printing. Printing should be consistently clear and dark.

Spacing. The text of the dissertation should be *double spaced*. Long quotations, footnotes, appendices, and references may be single spaced.

Color. It is recommended that you ***not*** rely on color to convey information (e.g., use symbols or labels rather than colors to differentiate the lines on a graph, or use stripes and cross-hatching instead of colors to distinguish areas on a map).

Photographs and Graphics. Photographs and graphics in the dissertation should be printed or photocopied directly on the paper as high quality black and white images. Scanned images must print clearly. If color must be used, ***only*** color laser or color photocopy printing is acceptable.

Use of materials copyrighted by others. IMPORTANT: Any material included that goes beyond "fair use" requires written permission of the copyright owner. It may be useful to include these in the dissertation as an appendix.

Pagination. Preliminary pages (i.e., the approval page, acknowledgments, table of contents, and the like) are to be numbered consecutively using lower case *Roman numerals*. All pages of the text, appendices (if any), and references must be numbered consecutively using *Arabic* numerals. The page number of the first page of each Chapter must be centered on the bottom of the page. In all other pages, the page number must be placed on the top of the page, aligned to the right.

Landscape pages. The top of a landscape page should be at the left margin and the bottom at the right margin. The page number is to be in the same relative position as on the portrait pages. An easy way to apply page numbers to landscape pages is to run them through the printer twice – once for the text, table, or figure (landscape orientation) and once for the page number (portrait orientation).

Sequence of the main components of the dissertation. The appropriate order of the major sections of the dissertation follows: the abstract, the title page, the copyright page (if needed), the approval page, acknowledgments, table of contents, the text, the bibliography/references, and appendices (if any).

Bibliography/References. The ACM (http://www.acm.org/pubs) or the IEEE (http://standards.ieee.org/resources) reference style should be followed.

Footnotes and Endnotes No specific rules. The format that is prescribed by your advisory committee should be followed.

* * *

It is recommended that you use your full legal name on the abstract, the title page, the copyright page (if appropriate), and on the approval page. Make certain that your name appears exactly the same way in all places.

Take a moment to check that all pages in all copies of your dissertation are accounted for and in the proper order before submitting final copies to the Departmental Secretary. *Two hardcopies* and *an electronic version* (in pdf format) of the dissertation should be submitted to the Departmental Secretary.

* * *

For students that plan to write their disseretation using Latex or MS Word authoring programms, they can download samples from http://www.cs.ucy.ac.cy/~chryssis/phd-specs.html.