University of Camerino

Faculty of Science and Technologies

Class no. 23/S

Second Level Degree Course in Information Technology

COURSE INTRODUCTION

This second level degree course aims to turn out graduates well versed in both the fundamental theory and the applications of the various branches of Information Technology. The knowledge acquired will enable graduates to keep up with, and contribute to, new developments in the sector and will also give the the opportunity to further their academic studies by taking for example a research doctorate.

The curriculum is designed to complement the learning and skills obtained in the three year first level degree course in Information Technology which is taught in our university. The 180 ECTS which students have received as a part of the previous course will automatically be accepted for enrolment on this course.

The exact weighting to be given to the 180 ECTS which graduates in the first level degree in Information Technology (Information Technology or I.T and Management) have obtained will be decided by the academic council (Consiglio di Classe) on the basis of students' curricula. Students from other first level degree courses taken in this, or other, universities in Italy or elsewhere may also apply provided they have obtained at least 120 ECTS that are relevant to this second level degree course in Information Technology.

COURSE ORGANISATION

Two years timetable. Every academic year is divided into two semesters (terms) in which periods of lectures and periods of examinations occur.

The first semester starts within the 15th October of each year and terminates at the end of February. It is organized as follows:

- 6 weeks of lectures
- 2 weeks for reading/examinations (approximatively end of November/beginning of December)
- 6 weeks of lectures (Christmas holidays excluded)
- 4 weeks for examinations (February)

The second semester starts at the beginning of March and terminates at the end of July.

It is organized as follows:

- 6 weeks of lectures
- 2 weeks for reading/examinations (in April)
- 6 weeks of lectures (until approximatively the middle of June)
- 6 weeks for examinations (2nd part of June, July)

August is period of vacations and there are 5/6 weeks for a supplementary session of examinations in September/1st part of October .

AVAILABLE COURSES

The student must acquire 120 ECTS (European Credits Transfer System) to obtain the Degree. The courses are distributed over the two years as shown in the following table:

Modules	Year	Semester	Lectures Hours		Individual Study Hours	ECTS
			Lessons	Excercise		
Distributed and Coordinating Calculus	I	I	45	30	225	12
Computer Networks 2	I	I	45	30	225	12
Theoretical Computer Science	I	II	24	18	110	6
Quantum Computation	I	II	36		89	5
Physics of Information Technology	I	II	28		72	4
Quantum Information	I	II	28		72	4
Complex Systems Performance Evaluation	II	I	24	18	110	6
Software Engineering 2	II	I	45	30	225	12
Computational Geometry	II	I	36		89	5
Codes and Cryptography	II	I	36		89	5
Computational Graphics	II	II	36		89	5

Language of Teaching

Currently all courses are in Italian. Courses in English will be activated in the near future. Tutoring and examinations can be currently done in English.

CURRICULUM

The Second Level Degree Course in Information Technology has a standard curriculum. However, a student may present a self-designed curriculum that must be approved by the academic council (Consiglio di Classe). The following table shows the distribution of the ECTS in the standard curriculum.

Year 1	ECTS	Semester
Distributed and Coordinating Calculus	12	1
Computer Networks 2	12	1
Complex Systems Performance Evaluation	6	1
Theoretical Computer Science	6	2
Quantum Computation	5	2
Free choices (***)	8	2
Physics of Information Technology (*)	4	2
Quantum Information (*)	4	2
Seminars	3	1, 2

(*) The student chooses one of these courses.

Year 2	ECTS	Semester
Software Engineering 2	12	1
Combinatorial Optimization	6	1
Computational Geometry (**)	5	1
Codes and Cryptography (**)	5	1
Seminars	2	1, 2
Free choices (***)	10	2
Final Thesis	20	2

(**) The student chooses one of these courses.

(***) Students may choose between available courses activated in the Academic Year in the Faculty (refer to http://www.cs.unicam.it/en/).

USEFUL INFORMATION

President of the Degree Course

Prof. Emanuela Merelli

Phone + 39 0737 402567

Mobile: +39 320 4381108

e-mail: emanuela.merelli@unicam.it

Tutoring Service

Dott. Pasini Leonardo

Phone: +39 0737 402562

e-mail: pasini.leonardo@unicam.it

Course Internationalization staff

Prof. Emanuela Merelli

Phone + 39 0737 402567

Mobile: +39 320 4381108

e-mail: emanuela.merelli@unicam.it

Dott. Luca Tesei

Phone: +39 0737 402572

Email: luca.tesei@unicam.it

Internationalization Office

ASSICOS

Via Le Mosse 22

62032 Camerino (MC)

Phone: +39 0737 404610 - +39 0377 404600

Email: relazioni.internazionali@unicam.it

How to enrol

Applications for enrolment should be addressed to the Rector and be written on official stamped paper and should be sent or delivered to the "Area Accademica e Didattica dell'Università" between the 15 th of July and the31st of October of each Academic Year.

Location

Department of Mathematics and Information Technology

Polo Informatico

Via Madonna delle Carceri 9

62032 Camerino (MC)

Phone: +39 0737 402565

Fax: +39 0737 402561

E-mail: emanuela.merelli@unicam.it

Web site: **`http://www.cs.unicam.it`**

Department

Department of Mathematics and Informatics

Via Madonna delle Carceri, 9

62032 Camerino (MC)

Phone: +39 0737 402549

Email: dipartimento.matematica@unicam.it

President: Prof. Flavio Corradini

Phone: +39 0737 402564

Mobile: +39 320 4381177

Email: flavio.corradini@unicam.it

Faculty Secretariat

Via Gentile III da Varano

62032 Camerino (MC)

Phone: +39 0737 402126

Mobile: +39 320 8604294

Email: segreteria.scienze@unicam.it

Students' Secretariat

Via Le Mosse 69

62032 Camerino (MC)

Phone: +30 0737 633517

COURSES DESCRIPTION

Codes and Cryptography

Course goals:

- Introducing problems of classical cryptography and public key cryptography
- Presenting the main public key cryptographic systems and their related theoretical foundations, either in Number Theory and in Complexity
- Listing the possible attacks and the corresponding security techniques

Needed knowledge for reaching the goals:

- Classical cryptography: Julius Caesar, Vigenère, Enigma. Perfect crypto-systems. Outline of the Shannon theorem. Public key cryptography. One way functions. Possible one way function sources in Number Theory and in Theoretical Computer Science. Recall on P = NP problem and on NP-complete languages. The bag problem. Prime numbers and factorization in Number Theory
- Prime numbers algorithms: Miller-Rabin and Solovay-Strassen probabilistic procedures, Agrawal-Kayal-Saxena deterministic polynomial procedure; outline of Shor quantum algorithm
- Factorization algorithms: p-1 method, rho method, Fermat factorization
- On public key cryptography: Diffie-Hellman cryptosystem, Massey-Omura crypto-system and ElGamal crypto-system, crypto-systems based on the bag problem, RSA crypto-system, digital signature methods. Zero knowledge protocols, telephonic toss, telephonic poker. Possible attacks to the RSA crypto-system: random hardware/software errors, coding and decoding time acceleration, smart cards usage, errors caused by an hacker

Combinatorial Optimization

Courses goals:

- Classifying and relating different modelling and algorithmic approaches to problems of combinatorial optimization, given their computational cost and the accuracy of solutions
- Showing the main characteristics of heuristics and meta-heuristics for combinatorial optimization problems
- Identifying valid inequalities
- Showing the main characteristics of network optimization algorithms

Needed knowledge for reaching the goals:

- Combinatorial optimization models
- Recalls of computational complexity
- Classical problems of combinatorial optimization

- Valid inequalities
- Classes of approximated algorithms for combinatorial optimization “hard” problems
- Examples based on classical problems
- Optimization problems on a network

Complex Systems Performance Evaluation

Courses goals:

- Acquiring the competence of modelling complex systems by networks of extended queues and the competence of constructing the corresponding simulators

Needed knowledge for reaching the goals:

- An overview of queueing network modelling
- Conducting a network study
- Foundation Laws
- Queueing network model input and output
- Bounds on Performance
- Models with one job class
- Models with multiple job classes
- Flow equivalence and hierarchical modelling

Computational Geometry

The course is meant to provide an overview of the main computational geometry techniques, characterized by a wide range of applications, starting from the “classical” combinatorial ones to the most innovative ones, either of analytic type or of digital type, involving students in experimental activities. The course topics, which are chosen also according to the students’ interests, may include:

- Classical algorithms (convex wrappers, Voronoi diagrams triangulations)
- Geometric interpolation (polynomial parametrization and spline, Bezier curves and surfaces, B-spline)
- Geometric modelling (representation and manipulation of curves, surfaces, solids
- Mesh generation (structured and unstructured, for plane regions and spatial surfaces)
- Configuration optimization (distribution of points, graphs, curves, surfaces, mesh)

Computer Networks 2

Courses goals:

- Recalling the general concepts of computer networks and performing an in-depth study on certain issues through programming exercises and research proposals

Needed knowledge for reaching the goals:

- Application level: HTTP, FTP, SMTP
- Socket programming and construction of a Web server
- Transport level: principles of reliable data transportation and of congestion control
- Network and routing level: routing algorithms, routing in Internet (autonomous systems), Ipv6
- Links and local area networks level: search and recovery of errors, multiple access protocols, asynchronous transfer (ATM), X.25 and frame relay
- Network and multimedia application. Network management

Distributed and Coordination Calculus

Course goals:

- Designing software applications in modern distributed systems
- Using the fundamental concepts that characterize the various models of distributed and coordination calculus
- Classifying, evaluating and applying some coordination languages
- Autonomously tackling distributed systems design.

Needed knowledge for reaching the goals:

- Distributed systems
- Distributed and coordination calculus models
- Software architectures and coordination
- Coordination languages
- Distributed applications middle-ware
- Object-oriented technology

Physics of Information Technology

Course goals:

- Showing Information Technology and the physical processes that regulate it.

Needed knowledge for reaching the goals:

- Physics of information management (transistors, converters, PN junctions)
- Physics of information storage (magnetism, CD players, MOS memory)

- Physics of information communication (instrumentation, shielding, light detectors, light emitters, displays, printers)

Quantum Computation

Course goals:

- Providing a new computational paradigm based on the re-examination from a “physical” point of view of Church-Turing principle
- Presenting the main quantum algorithms
- Evaluating the potentiality of the quantum approach to computation with respect to the classical one

Needed knowledge for reaching the goals:

- Introduction and historical outline
- Outline of quantum mechanics: classical mechanics and probabilistic systems, Hilbert spaces, quantum mechanics postulates, states superimposition and states separability, Pauli matrices
- Outline of classical computation theory: deterministic and non-deterministic Turing machines, circuital models, computational complexity, reversibility
- Quantum theory of computation: quantum Turing machines, quantum complexity classes, quantum circuits, universality of logical ports, quantum logical operations and functions evaluation, random numbers generation
- Quantum Fourier transform and its applications: quantum Fourier transform, order-finding problem, period-finding problem, Shor algorithm, hidden subgroup general problem
- Search algorithms: Deutsch algorithm, Grover algorithm, quantum search and simulation, quantum counting, search and NP-complete problems, quantum computation limits. Possible implementation of quantum processors and physical limits of computation: photon devices, ion traps, nuclear magnetic resonance, solid state devices, Bekenstein limit

Quantum Information

Course goals:

- Introducing a “physic approach” to the concept of information
- Giving the basic knowledge concerning the manipulation of information, both in classic theory framework and in quantum theory framework
- Understanding the potentialities and the limitations of information processes that are bound to present theories

Needed knowledge for reaching the goals:

- Introduction and historical notes
- Hints on quantum mechanics, Pauli matrixes, overlap and separability, Schmidt decomposition and purification

- Theory of the quantum noise: superoperators and quantum maps, maps for different types of noisy channels, relationship with other approaches
- Hints on the theory of error recovering: redundancy, hints on classical theory of error recovering, distance measures, conditions for error recovering, linear codes, discretization of quantum errors, stabilizers theory, hints on fault-tolerance
- Entropy and information: Shannon entropy and its properties, von Neumann entropy and its properties, distinguish-ability of quantum states and accessible information, data compression and Shannon theorems, Schumacher theorem, Holevo-Schumacher-Westmoreland theorem, dense coding, teleportation
- Transformations of entanglement: distillation and dilution of entanglement, quantification of entanglement. Cryptography: historical notes, private keys and public keys, Vernam cipher and RSA system, quantum distribution of keys, private amplification, information reconciliation, relationship with error recovery codes, protocols security

Software Engineering 2

Course goals:

- Studying the main methodologies for formal designing of concurrent, distributed, real-time and embedded systems

Needed knowledge for reaching the goals:

- Introduction to complex systems and their application areas
- Specification and modelling languages of systems
- Graphical and textual notations for high-level modelling
- Verification and validation of functional and non-functional requirements

Theoretical Computer Science

Course goals:

- Measuring the least needed resources (time, space, memory, casualty) to resolve a problem
- Ordering the problems according to their complexity
- Introducing the main complexity classes of problems and studying their relationships

Needed knowledge for reaching the goals:

- Recalls of computability. Turing model
- Evaluation of the complexity of a problem, some measurement criteria. Reduction between problems
- Temporal complexity classes. P and the Edmonds-Cook-Karp thesis, NP. The P = NP problem. NP-complete problems. Cook theorem on NP-completeness of SAT. Examples of NP-complete problems. Discussion of the P = NP problem: co-NP, NP-in-between problems, polynomial hierarchy and the class PH
- Spatial complexity classes. LOGSPACE, NLOGSPACE, PSPACE, NPSPACE, Savitch

theorem

- Probabilistic complexity classes: BPP, RP, ZPP
- Complexity and boolean circuits. Interactive algorithms, class IP, class AM (Artù – Merlin). Shamir theorem