UNIVERSITY OF MOLISE

DEPARTMENT OF BIOSCIENCE AND TERRITORY

Bachelor in Computer Science

Course overview

July 2015

Course Learning Outcomes

The Computer Science course has a single branch of study, where students can acquire experimental and applied theories and methodologies in the fundamental areas of Computer Science. Such knowledge forms the conceptual and technological basis for the design, organization and management of large software systems. Students will acquire skills that could be useful not only for industries in the area of information systems and networking, but also in public administrations and, more in general, in all the organizations making use of IT technologies.

The course also allows students to obtain specific skills related to (i) the maintenance and evolution of software systems; (ii) geographic information systems; and (iii) computer security. Knowledge and skills about the maintenance and evolution of software systems play an important role in modern societies. It is estimated that the maintenance cost of a software system exceed by more than 50% the development cost. The acquired skills will allow students to (i) design software systems that are easily to be extended; and (ii) effectively plan maintenance activities on critical and large software systems. Emphasis will also be given to the migration of "legacy systems" towards new technologies, namely web and mobile technologies. Regarding the latter, special attention will be also given to the development of "green" applications, *i.e.*, applications with limited power consumption.

In the context of the course, a fundamental role is also played by computer security. Besides knowledge on how to design and implement defensive solutions through firewalls and intruder detectors, the course will also provide students with knowledge on creating preventive solutions through technology intelligence. All the acquired skills will allow students to implement solutions that can identify dangers and threats in the context of information flows that characterize communication infrastructure, such as the World Wide Web.

Finally, the course will also provide students with knowledge about legal aspects of computer applications and computer crimes. Nowadays, legal informatics plays a key role in the processing of sensitive data.

Summarising, the course aims at training the following professionals: software analysts, software engineers, data administrators, database designers and administrators, and networking designers and administrators.

Expected learning outcomes

Next subsections describe the expected learning outcomes organized by the three areas of study that characterize the course, *i.e.*, area of "Mathematics, Physics, and Statistics", area of "Computer Science", and area of "Legal Informatics". The description of the expected learning outcomes is based on the Dublin Descriptors.

Area of Mathematics, Physics, and Statistics

Knowledge and understanding. Students will acquire knowledge on and understanding of basic elements of Mathematics, Physics and Statistics. These skills are essential to achieve the other learning objectives of the course. Specifically, Computer Science graduates will be able to demonstrate knowledge on and understanding of

- scientific language and methodologies;
- basic mathematical elements;
- linear algebra and geometry tools;
- historical and epistemological mathematical elements;
- physical phenomena;
- basic elements of statistics;
- evolution, from the physical and electronic point of view, of the automatic calculation;
- statistical methods of machine learning;
- numerical methods.

Applying knowledge and understanding. Computer Science graduates will use the acquired knowledge to achieve the following goals:

- formalize and solve a mathematical problem;
- study and describe a physical phenomenon with scientific rigor;
- use a specific machine learning technique for the design of a decision support system;
- design and implement mathematical algorithms for the efficient resolution of scientific computational problems.

Area of Computer Science

Knowledge and understanding. Students will acquire knowledge on and understanding of conceptual foundations of Computer Science. These skills are essential to acquire more specific knowledge and professional skills. In particular, Computer Science graduates will demonstrate knowledge on and understanding of

- fundamental principles of Computer Science, related to programming languages, algorithms and systems;
- database management;
- methods and tools for the development of software systems;
- methods for the maintenance and the evolution of software systems;
- software development technologies;
- architecture of modern computer networks;
- artificial intelligence methods and tools;
- geographic information systems.

Applying knowledge and understanding. Graduates will use the acquired knowledge to analyse, design and develop a software system. In particular, Computer Science graduates will be able to

- understand the feasibility and complexity of Computer Science problems and select appropriate methods for their analysis and design;
- formalize real problems where the computer is part of the solution, and identify appropriate solution patterns;
- apply appropriate methodologies both for the development of new software systems and the maintenance of existing ones;
- apply techniques and tools for the migrations of legacy systems towards new technologies, namely web and mobile technologies;
- design user interfaces of software applications that meet usability standards;
- evaluate and design computer systems security solutions;
- exploit artificial intelligence techniques to solve complex problems;
- design and implement a geographic information system.

Area of Legal Informatics

Knowledge and understanding. Students will acquire knowledge on and understanding of legal aspect related to Compute Science. Such knowledge is nowadays fundamental to properly manage sensitive data. Specifically, Computer Science graduates will be able to demonstrate knowledge on and understanding of

- regulations related to information technology;
- general principles relating to data processing.

Applying knowledge and understanding. Graduates will use the acquired knowledge to properly manage legal aspects related to the development of a software system. In particular, Computer Science graduates will be able to

- apply information technology in compliance with the related regulations;
- manage sensitive data according to the related regulations;
- apply appropriate measures for the security of sensitive data.

Making judgements, Communication, Lifelong learning skills

Making judgements. Students will acquire knowledge on how to collect and interpret data aiming at formulating a subjective judgement. This relates to the ability of graduates to combine and abstract their technical skills to solve problems that include aspects in a wide technological context. The graduates will be able to use appropriate methods aiming at usefully immerge themself in a professional context. In particular, Computer Science graduates will demonstrate

- ability to manage both theory and practice to solve Computer Science problems;

- understanding the state-of-the-art of technologies in their area of expertise and their applications;
- skills related to professional responsibilities and legal regulations related to information technology.

In the context of the course, such capabilities are acquired during Computer Science specific courses and additional activities, with particular reference to the courses of software engineering, computer security, geographic information systems, and legal informatics.

Communication. Students will acquire knowledge on how to communicate with specialists and non-specialists, honing the professional skills needed to communicate information, ideas, problems and solutions. In particular, Computer Science graduates will demonstrate ability to

- work effectively as an individual and as a member of a working group;
- communicate effectively with colleagues and potential users about issues and problems related to their area of expertise;
- present ideas and suggest convincingly solutions in both written and oral forms;
- use effectively, in written and oral forms, at least one European language - other than Italian - in the specific field of expertise and to exchange general information.

In the context of the course, such capabilities are acquired during computer science specific courses, especially those having laboratory sessions that include the development, in working groups, of software projects. Communication skills are also acquired by students in the English course.

Lifelong learning skills. Students will acquire not only abilities to learn, but also (and more important) abilities to apply the acquired knowledge, starting to orientate in a theoretical and/or professional context that complements the training course completed. In particular, Computer Science graduates will demonstrate ability to

- set up and properly solve new theoretical and applied problems;
- respond positively to the various work assignments entrusted as part of internship/working experience;
- hone skills autonomously choosing additional areas of learning.

In the content of the course, such capabilities (and the refinement of such skills) are acquired during specific activities chosen by students, as well as through experiences of internship.

Curriculum - Academic year 2015/2016

	First year		
N.	**Course**	**ECTS**	**Semester**
1	Mathematics	12	I II
2	Programming	12	I II
3	Computer law	10	I II
4	Evolution of automatic computation	6	I
5	Formal languages and compilers	6	I
6	Computer architecture	6	II
7	Operating systems	9	II
	English language (pass mark)	3	I
	Total ECTS – 1st year	**64**	
	Second year		
N.	**Course**	**ECTS**	**Semester**
8	Statistics for technology	6	I
9	Algorithms and data structures	10	I II
10	Software engineering	10	I II
11	Database and information systems	10	I II
12	History of Mathematics	6	I
13	Numerical computation	6	II
14	Physics	7	I
	Total ECTS – 2nd year	**49**	
	Third year		
N.	**Course**	**ECTS**	**Semester**
15	Computer networks and security	12	I II
16	Mobile and web programming	10	I II
17	Geographical Information Systems	6	I
18	Software evolution	6	I
19	Artificial intelligence	6	II
20	Optional courses	12	I II
	Traineeship	5	
	Final essay (thesis)	4	
	Total ECTS – 3rd year	**61**	

Total ECTS	**180**
Total exams	**20**

Optional courses - Academic year 2015/2016			
First year			
N.	**Course**	**ECTS**	**Semester**
1	Communication skills for Computer Scientists	3	II
2	Visual communication	3	II
3	Computer ethics	3	I
4	Computational methods for optimization	3	II
5	Mathematical methods in Science	6	II
6	Fundamental concepts of Chemistry and new materials	3	II
7	Semantic web	3	II

MATHEMATICS

Lecturer	Giovanni Capobianco
ECTS	12

Learning outcome and their consistency with the objectives of the course of study

To review the basics of Mathematics. To provide tools of linear algebra and geometry that are essential for scientific degree course students. To provide essential knowledge in mathematical analysis for scientific degree course students with particular emphasis on computer science applications.

To enforce abstraction, *formality, and rigor of mathematical reasoning.*

Content of the Program/Course:

I Module

(1) Sets, Relations, Functions: The sets, representations and operations; The numerical sets: N, Z, Q, R; Cartesian product. Binary relations. Order relations. Logic elements.

(2) Equations, inequalities, combinatorics: equations and inequalities: algebraic, logarithmic, exponential; Factorial and binomial coefficients. Permutations with and without repetitions. Combinations.

(3) The real functions: The Cartesian plane. Real function of a real variable; Properties and graphs of elementary functions.

(4) Complex Numbers: The set of complex numbers. Algebraic form, geometric representation, trigonometric form, exponential form. Operations between complex numbers. Fundamental theorem of algebra.

(5) Lines, circles, conic sections: Line, Circle, Ellipse, Hyperbola and Parabola in the Cartesian plane.

(6) Matrices and Linear Systems: Matrices and matrix operations. Determinants. Diagonal matrices, identity, transpose, symmetrical. Invertible matrices and inverse matrix. Rank of a matrix. Linear equations in n unknowns. Systems of linear equations. Cramer's rule. Gaussian elimination.

II Module

(7) Limits and continuous functions: Numerical sequence. Limit of a sequence; Limit of a function; Continuous functions; Asymptotes.

(8) Derivatives: Definition, physical meaning and geometrical interpretation; Properties and rules; Derivatives of elementary functions; Application of the derivative. Taylor formula. Differential of a function.

(9) Integration: Definite integrals; Properties; Integral function; Fundamental Theorem and Formula of calculus; Primitive of a function, The indefinite integral: definition and properties; Integration methods.

(10) The series: Numerical series; Series in non-negative terms; The geometric series; The harmonic series; Convergence criteria; Alternating series; Series of functions.

(11) Differential equations: differential equations of the first order; Bernoulli equations; Equations with separation variables; Differential equations of the second order with constant coefficients.

(12) Reference systems in plane and in space. Functions of several variables: Polar coordinates, spherical, cylindrical. Domain of a function of two variables, Cartesian representation. Limits and continuity. Partial derivatives and gradient. Successive derivatives. Schwarz theorem. Maxima and minima.

Suggested text-books

Theory:

Lecturer's slides.

Marcellini, Sbordone: ELEMENTI DI CALCOLO, Liguori editore;

*Exercises***:**

Marcellini, Sbordone: ESERCITAZIONI DI MATEMATICA, 1° VOLUME, PARTE PRIMA E PARTE SECONDA, Liguori editore.

Research material:

Fiorenza, Greco: LEZIONE DI ANALISI MATEMATICA I e II,

Demidovic: ESERCIZI E PROBLEMI DI ANALISI MATEMATICA

Prerequisites / Co-requisites:

The students, having passed a state exam at the end of their studies in high school, should be aware of elementary arithmetic and algebra necessary to complete the course.

Organization of teaching: (lectures, tutorials, laboratory, etc.)

Lectures. During the tutorials theories and examples are presented and students have the chance to complete exercises using scientific computing software.

Language

Italian

Methods and assessment criteria

A written exam and an oral exam on each module which students can take in the same exam session or in two different periods. To gain access to the oral examination the students must have obtained at least 16/30 in the written exam.
Written exam: two or three exercises lasting a maximum of 80 minutes.
Oral exam: questions about definitions, statements, demonstrations, concepts, graphics, calculations on the topics covered in the course.

Assessment of the knowledge, skills and competences according to Dublin Descriptors:

Descriptor	**Topics**	Assessment Methods
Descriptor 1 - Knowledge and understanding;	Numerical sets, combinatorics, elementary functions, matrices, complex numbers, loci of the plan, limits, derivatives, integrals	Questions on the issues in the oral exam
Descriptor 2 – Applying knowledge and understanding;	Solving equations in the field of complex numbers, behaviour of elementary functions, remarkable points, measures, trends loci of the plan, matrix calculation, calculations of limits, derivatives and integrals of both direct and inverse problems or questions when instrumental to answer to a problem.	Exercises to do in the written exam and possibly also in the oral exam
Descriptor 3 – Making judgements;	Solving exercises in a different way from how learned a lesson. Preference of a proof of a theorem performed differently from the way shown by the teacher.	Exercises to do in the written exam and / or questions on the issues in the oral exam
Descriptor 4 – Communication skills;	Using the appropriate language in enunciating theorems, demonstrations, specify definitions.	Questions on the issues in the oral exam
Descriptor 5 – Learning skills.	Having the ability to identify the mathematical tool useful to solve a given problem.	Exercises to do in the written exam and / or questions on the issues in the oral exam

PROGRAMMING

Lecturer	Rocco Oliveto
ECTS	12

Learning outcome and their consistency with the objectives of the course of study

The student, in line with the objectives of the course, will develop skills related to the realization of simple programs through the use of procedural and object-oriented programming.

Knowledge and understanding

The student, at the end of the first module of the course, will know the theoretical and practical aspects related to the design and coding of programs in C language based on linear data structures (programming in the small), with particular emphasis to the principle of functional abstraction and the definition of abstract data types. At the end of the second module, the student will know the theoretical and practical aspects related to the design and coding of programs written in Java with graphical interfaces.

Knowledge and understanding applied

The student will be able to design and develop through the use of the C and Java language simple programs.

Making judgments

The student will be able to analyze a set of functional requirements and to adopt an implementation strategy for these requirements different from those learned during the course.

Communication skills

The student will be able to describe with sufficient level of formalism and appropriate language a solution to a specific problem.

Ability to learn

The student will be able to evaluate the different strategies and choose the most suitable solution in specific circumstances, while being aware of the limitations and strengths of the selected solution.

Content of the Program/Course:

First part - 6 ECTS

Credit 1. Introduction to problem solving. Design of simple algorithms. Elements of computer architecture.
Credit 2. Introduction to the C language. Operators and basic data types. Flow control.
Credit 3. Functional abstraction in C. Arrays, pointers and strings.
Credit 4. Sequential and binary search. Sorting problem.
Credit 5. Linear data structures in C. Abstract data types.
Credit 6. Storing data on file.

Second part - 6 ECTS

Credit 7. Introduction to object-oriented programming. Inheritance and polymorphism. The Liskov's substitution principle.
Credit 8. Introduction to the Java language. Differences between C and Java.
Credit 9. Comparison of objects in Java. Linear data structures in Java: the Collection framework.
Credit 10. Event-driven programming. The Swing library of Java.
Credit 11. Multi-threading programming.
Credit 12. Principles of human-computer interaction. Design and implementation of simple video-games.

Suggested text-books

Al Kelley e Ira Pohl, C-Didattica e programmazione, Pearson Education, IV edizione, 2004.
Cay Horstmann S e Gary Cornell, Core Java - I fondamenti, Prentice Hall, VIII edizione, 2008. Horstmann S e Gary Cornell, Core Java – Tecniche avanzate, Prentice Hall, VIII edizione, 2008.

Organization of teaching: (lectures, tutorials, laboratory, etc.)

Lectures and tutorials.

Attendance is not mandatory, but it is strongly recommended in order to effectively achieve the objectives of the course, as it allows the terminology and basic concepts to be learned more easily, enhancing cognitive capabilities needed for self-study, the theoretical content and their possible implementation.

Language
Italian.
Methods and assessment criteria
The examination is organized in two parts in line with the division into two modules of the course. The first part of the exam consists of a practical test and an oral exam. The duration of the practical test, carried out at the computer, is 3 hours. In this test the student is required to implement a simple program (with no more than 5 functional requirements) through the use of the C language. This test is designed to assess knowledge and ability to understand and apply the student's independent judgment. Students who get a grade higher than 18/30 in the practical test can have access to the oral exam, which will take place on the same day of the practical test, to emphasize that exam preparation is global, and does not distinguish between practical teat and oral exam preparation. The oral exam consists of the discussion of the strategies implemented by students during the practical test. The aim is to evaluate the students' communication skills and their ability to learn. If a positive result is achieved in the oral exam, the student will be given an overall grade (practical test and oral exam). This grade represents the overall assessment for the first part of the exam. Students who have obtained an overall grade on the first part of the exam of greater than 18/30 can have access to the second part of the exam. Similar to the first part, the second part consists of a practical test and an oral exam. The practical test consists of the development of a GUI-based program through the use of Java language. The development of a simple video game is highly recommended. During the oral exam, students will discuss the strategies implemented in their projects with the aim of enhancing communication skills and their ability to learn. If the project and the oral examination both receive positive evaluation, the student will be assigned a grade of between 0 and 4 points. This grade will be added to the grade achieved on the first part of the exam in order to attain the overall grade.

COMPUTER LAW

Lecturer	Barbara Troncarelli
ECTS	10

Learning outcome and their consistency with the objectives of the course of study

This course objectives are:
first module:
1) knowledge of the link between computer science and law, in order to understand regulatory implications of information technology (legal informatics and ICT law: origins and development; protection of intellectual property; computerization and law in Public Administration; Internet governance; computer crimes);
2) applying knowledge, so as to be able to operate proper management of information systems in conformity with legal provisions;
second module:
1) formation of a suitable knowledge base about principles and rules of the right to privacy related to activities of automatic processing of personal data in public or private sector, as well as spamming, data retention, social network, smartphone and tablet, cloud computing, video surveillance, biometrics;
2) ability to apply knowledge in order to manage various aspects concerning the rules on privacy and security, with particular attention on implementation of appropriate security measures for the correct use of information technology in the field of sensitive personal data processing.
Through knowledge acquisition and understanding of the legal implications of information technology, this course aims to be consistent with the educational objectives of the study program and of the "Dublin Descriptors", according to which it is necessary to develop a good theoretical background in the various areas, including the legal area. Moreover, it is essential to promote the application of acquired knowledge, that is to say, formation of operational capabilities such as to be able to deal with the prescriptive contents related to the legal regulation of new technology, especially regarding protection of personal data. This reflects the aim of the whole Course to create professionals in step with the complex dynamics of technological development and with the growing demand for interdisciplinary expertise, also relating to rules of law and not only to technical information.

Content of the Program/Course:

First module

Credit 1. Introduction to legal informatics: origins and development; theories and perspectives; information systems in legal field; law and information society.
Credit 2. Technology and regulation, with particular attention to: intellectual property issues; free software and open source; copyright and Internet.
Credit 3. Public Administration: computerization and law, especially about digital firm; Digital Agenda for Europe; Italian Digital Agenda.
Credit 4. Internet, social networks and legal implications.
Credit 5. Strategy to combat cybercrime in European Union; law on computer crime in Italy.

Second module

Credit 1. Legal protection of personal data within the EU.
Credit 2. Protection of personal data according to Italian legal system.
Credit 3. General principles regarding treatment of personal data; persons involved in treatment; rights of interested parties; rules for treatment of personal data.
Credit 4. Security of data and systems. Minimum security measures. Appropriate security measures. Provisions relating to treatment of personal data in public and private sectors.
Credit 5. Regulatory issues with regards to spamming, data retention, social network, smartphone and tablet, cloud computing, video surveillance, biometrics.

Suggested text-books

Reference book:
M.G. Jori, *Diritto, nuove tecnologie e comunicazione digitale*, Giuffrè, Milano 2013.

Slides and papers of interest presented in the lessons.

Further bibliographic references:

G. Taddei Elmi, *Corso di informatica giuridica*, Simone, Napoli 2010;

M. Sirimarco (edited by), *Info-ius: problemi e prospettive dell'informatica giuridica*, Nuova Cultura, Roma 2010;

C. Di Cocco, G. Sartor, *Temi di diritto dell'informatica*, Giappichelli, Torino 2013;

European Union Agency for Fundamental Rights, *Handbook on European data protection law*, Publications Office of the European Union, Luxembourg 2014.

Organization of teaching: (lectures, tutorials, laboratory, etc.)

- Lectures.
- Multiple choice tests during the course.
- Brief presentations in powerpoint, optionally performed by students in classroom, individually or in groups, on topics proposed by the teacher.
- Students attending class will have the possibility to carry out, at the end of the course, a written exercise on topics of the lessons, with multiple choice and open questions, as useful feedback activity and incentive to a good preparation for the final exam.

These methods will make use of the following tools to support teaching: 1) "virtual classroom", used by the teacher to provide students with detailed indications about program and bibliography, to transmit on-line slide and other material, to indicate some useful links, and to give information on the teaching activities; 2) "Moodle" web platform used to propose tests during the course, to increase the level of student interest, and to facilitate the learning process.

Attending class is not mandatory, but it is strongly recommended in order to achieve the educational objectives, because this allows terminology and fundamentals of computer law to be learnt more easily, by encouraging capabilities needed to tackle conceptual contents and their possible implementation.

Language

Italian.

Methods and assessment criteria

The final exam is oral for all students: they will have to answer some questions on themes of the course. The exam is aimed at evaluating the results, also through questions on topics chosen by the student.

The oral examination appears to be a suitable way to achieve the following objectives and expected results: 1) knowledge and understanding of theoretical contents; 2) correlative ability to apply them at normative level. Moreover, the oral verification, in which the student may be required to supplement the answers with a topic of his/her choice, is a way to pursue two further aims reaffirmed by the Course of study and formulated by the "Dublin Descriptors": 1) to encourage judgment-making as the ability to perform plausible decisional acts and assessments; 2) ability to expose (communication skills) as indispensable capabilities, especially in the professional world today. It is thus possible to face future work experiences with the aid of greater preparation.

EVOLUTION OF AUTOMATIC COMPUTATION

Lecturer	Giovanni Maria Piacentino
ECTS	6

Learning outcome and their consistency with the objectives of the course of study

Obiettivi formativi e loro coerenza con gli obiettivi del corso di studio (max 3800 caratteri s.i.)

The course aims at providing students with knowledge on how calculation methods and, more in general, information processing have evolved through time, from the very beginnings to present days.

Knowledge and understanding

At the end of course, students will know the operating principles and modes of several mechanical and electronic computers as well as of the calculation methods that have been developed for data processing.

Applied knowledge and understanding

Students will understand the historical evolution of problems and solutions achieving cultural maturation useful to broaden professional horizons and preparation.

Making judgments

Students will achieve the necessary skills to evaluate the pros and cons of each of the solutions to the problem of data processing proposed in the literature and they can project themselves into the future by imagining new solutions to new problems.

Communication skills

Students will be able to express complex technical concepts, such as those involved in the structure and components of many systems of information processing and computing, through plain language that is understandable also to non-experts.

Ability to learn

Students will be able to understand how the equipment presented operate.

Content of the Program/Course:

Credit 1. Automatic calculation of astronomical events; analog computation, Astrario of de Dondi, Astrolabe, a even older calculator, from Al-Khoresmi to Luca Pacioli, Adelardo of Bath, Robert of Chester, Gerard da Cremona, and Abacisti snf alfebristi, Leonardo Pisano and Liber Abaci, Sacrobosco, Hauksbók, Paul Dagomari, Luca Pacioli.

Credit 2. The Roman abacus, Japanese abacus, Russian abacus, Chinese double abacus, Neper's sticks, The Schickard's machine, Pascal's machine, Leibnitz's machine, Aritmometro of Thomas de Colmar, Braunsweig Odnher, Facit, the first items for analog computation, Galileo's compass, The Slide Rule, The Planimeters.

Credit 3. The Swedish branch of Odhner, First Brunsviga, American Developments: Burroughs Adding Machine Company, The Comptometer Facit and Friden, the Double calculator, Marchant and calculation of military shooting, Marchant Figurematic, Monroe and Mercedes, Olivetti's revolution, Divisumma 14 and Class 14, Class 24 and Tetractys, the late Swedish competition, Class 26, the Summit of the mechanical calculator 27. The logos 27. The competitors of Logos 27 Marchant Monroe, Friden and Precise (The best despite being worst than Olivetti's)

Credit 4. The history of mechanographics, Joseph-Marie Jacquard invented the loom punch cards, Charles Babbage built the first computer. The Difference Engine is not completed for the inadequacy of the technology, Babbage in Turin, first programmer Ada Lovelace, Herman Hollerith reinvents the perforated cards, mechanographic centers, Birth of IBM, Dehomag D11, IBM and the Holocaust, Univac and Remington, The Olivetti Bull. An example of process inventory and billing.

Credit 5. The machine of Zuse, Atanasoff-Berry Computer, Eniac. Components and internal logic by Relay, Valves, Transistors, Integrated circuit, The first computers: Harvard Mark 1, ENIAC and EDVAC IAS by Von Neuman, IBM Business Computers, Temington, Olivetti ELEA, Honeywell II 370 IBM, components and increasing scale of integration supercomputer.

Credit 6. The first electronic calculators: the Anita, The logos 328, 270 and the RPN by Olivetti to Hewlet-Packard. Program 101 of the first personal calculator and Hewlett Packard on Italian patent. 8080 and CP/M, Motorola and Apple PC IBM and Microsoft, the competition by Olivetti M24. The birth of the MacIntosh History of Programming Languages, History of Digital.

Suggested text-books
Notes by the lecturers distributed to students during the course.
Prerequisites / Co-requisites:
Knowledge of English language.
Organization of teaching: (lectures, tutorials, laboratory, etc.)
Lectures and instruction in the computer laboratory.
Language
Italian
Methods and assessment criteria
A term paper (that can be produced by a group of students) and an oral exam (individual). The term paper aims to assess theoretical knowledge acquired during the course and the ability to apply algorithms studied during the course to specific situations. Students can have access to the oral exam after obtaining a minimum score of 18/30 in the term paper. The oral examination aims at assessing (i) the ability to argue the choices made in the term paper; and (ii) the ability to apply acquired knowledge. Assessment of knowledge, skills and competences according to the Dublin Descriptors: Knowledge and understanding Student's ability to understand the historical evolution of technology development will be assessed. Theoretical and practical knowledge will also be evaluated during the oral exam. Application of knowledge and understanding Student's ability to clearly express original historical considerations will be evaluated. Making judgments Student's ability to describe and compare processing tools shown during the course will be evaluated. Communication skills Student's ability to express complex technical concepts through the use of plain language, understandable to non-experts, will be evaluated. Learning skills Learning skills will be evaluated through the production of the term paper and the discussion on the course topics.

FORMAL LANGUAGES AND COMPILERS

Lecturer	TBD
ECTS	6

Learning outcome and their consistency with the objectives of the course of study

The student, in line with the objectives of the course, will develop skills related to programming languages, with particular reference to (i) the interpretation of the instructions and the compilation; (ii) the different types of automata used in various stages of compilation; and (iii) the formalization of algorithmic solutions to specific problems.

Knowledge and understanding

The student, at the end of the course, will know the theoretical and practical aspects on the principles of programming languages and automata.

Applied knowledge and understanding

The student will be able to design, through the use of automata and/or formal languages, algorithmic solution to simple problems.

Making judgments

The student will be able to solve a problem in a different way from how learned during the lesson.

Communication skills

The student will be able to describe with sufficient level of formalism and inappropriate language a solution to a specific problem.

Ability to learn

The student will be able, given a problem, to evaluate the different solution strategies and choose the most suitable in specific circumstances, being aware of the limitations and strengths of the selected solutions.

Content of the Program/Course:

Credit 1. The concept of system; algorithm; program; algorithm representation; flow diagrams; pseudo-coding.

Credito2. Languages and grammars; equivalence of grammars; Classification of grammars according to Chomsky hierarchy.

Credit 3. Deterministic finite automata.

Credit 4. Deterministic Pushdown automata.

Credit 5. Turing machine; an overview of algorithmic complexity.

Credit 6. Interpreters and compilers.

Suggested text-books

Gabbrielli M., Martini S. *Linguaggi di Programmazione: Principi e paradigmi,* McGrawHill.

Acciaro, Marengo, Roselli. *Analisi e progettazione di algoritmi*. Adriatica Editrice Bari, 2002.

Organization of teaching: (lectures, tutorials, laboratory, etc.)

Lectures and exercises.

Attendance is not mandatory, but it is strongly recommended in order to effectively achieve the objectives of the course, as it allows to learn faster the terminology and basic concepts, enhancing the cognitive capabilities needed for self-study the theoretical content and their possible implementation.

Language

Italian

Methods and assessment criteria

A written and oral exam.

The duration of the written test is 2 hours and consists of practical exercises that require the formalization of a solution to a specific problem, through the use of automata or formal languages. This test is designed to assess the knowledge and ability to understand and apply the independent judgment of the student.

Students who get a grade greater than 18/30 in the written test access to the oral exam. The oral exam will take place on the same day of the written test, to emphasize that the exam preparation is unique, and does not distinguish between a preparation for written test and oral exam. The oral exam consists of the

discussion of the exercises solved during the written test in order to evaluate both the communication skills of the students and their ability to learn.

COMPUTER ARCHITECTURE	
Lecturer	Mario Petrone
ECTS	6

Learning outcome and their consistency with the objectives of the course of study

The goal of the Computer Architecture course is to provide students basic knowledge of computer organization and the principles underlying their functioning. We will study digital circuits theory (combinational and sequential machines analysis and synthesis), as well as some more advanced aspects of computer systems organization and architecture. In addition, the course will provide essential tools for the understanding and use of assembly language.

Content of the Program/Course:

Credit 1. Introduction. Processing systems introduction. Abstraction levels of the architecture of a processing system. Historical perspective on computer architecture. Processing system's base models. Parallel processing architectures.
Credit 2. Information representation. Binary arithmetic. Numbers, texts, images and sound coding.
Credit 3. Logic networks. Basic concepts of combinatorial and sequential synthesis. Logic ports and Boolean algebra. Functions, minimal forms, normal forms. Network and combinatory module design. Minimization technics. Arithmetic and logic unit (ALU).
Credit 4. Synthesis of sequential networks. Combinational and sequential circuits. Macro computer architecture.
Credit 5. System design. Von Neumann's architecture. Data path and instructions execution. Instructions and format. Assembly language.
Credit 6. Peripherals and memorization units. Peripherals, memorization units and interconnections. Primary memory. Secondary memory. I/O and bus.

Suggested text-books

M. Petrone, R. Caruso, Sistemi di elaborazione delle informazioni, Milano, Franco Angeli, 2008.
A.S. Tanenbaum, Structured Computer Organization, Prentice-Hall.
Lecturer supplied material. Online material

Organization of teaching: (lectures, tutorials, laboratory, etc.)

Lectures and exercises.

Language

Italian

Methods and assessment criteria

The exam consists of a written test lasting two hours and a mandatory oral test, if the written test scores between 15/30 and 17/30, or optional if the written test scores at least 18/30. This exam's vote is the sum of the written test and the possible oral exam scores.

Assessment of the knowledge, skills and competences according to Dublin Descriptors:
Knowledge and understanding
Acquiring fundamental knowledge of computer organization and functioning.
Applying knowledge and understanding
Ability to evaluate computer system performances based on a full understanding of possible technical solutions.
Making judgments
Ability to analyze and evaluate a computer's architecture according to its components.
Communication skills
Ability to describe computing system solutions by analyzing technical specifications provided.
Learning abilities
Development of self-learning skills through consultation of advanced literature on the subject. Ability to adequately understand materials developed for master's degree as well as first-level master's programs using acquired knowledge.

OPERATING SYSTEMS

Lecturer	Fausto Fasano
ECTS	9

Learning outcome and their consistency with the objectives of the course of study

The course aims at providing the basic concepts related to modern operating systems. The student will understand the role of the operating systems, the management of resources by the operating system and the main differences between the various modern operating systems. The student will address some of the problems typical of the shared management of multiple resources, from a theoretical point of view as well as a practical point of view. Finally, the student will become familiar with the various distributions and with the command interpreter for Linux.

Knowledge and understanding

The student, at the end of the course, will know the theoretical and practical aspects underlying the implementation and use of operating systems with particular reference to the GNU-Linux.

Applied knowledge and understanding

The students will be aware of the different modern operating systems and different distributions. They will be encouraged to customize the environment to suit their specific needs. In order to experience the different distributions a virtualization software will be used.

Making judgments

The student will achieve the necessary skills to evaluate the pros and cons of each of the tested solutions not only from an end user perspective, but also in terms of an experienced systems engineer.

Communication skills

The student will be able to express complex technical concepts, such as those involved in the structure and components of a modern operating system through clear and understandable language even to non-experts.

Ability to learn

The student will be able to understand the operations of an operating system accessing the source code of a real system and experiencing the different functions. The student will also be able to access the main sources of online documentation and understand the contents based on the basic understanding of the foundations of a modern operating system.

Content of the Program/Course:

Credit 1. Introduction to Operating Systems. Background. Activities and structure of an operating system. The kernel and modules of an operating system.

Credit 2. Linux and the major distributions. Installing and configuring the operating system. The command interpreter.

Credit 3. Software processes. Properties of the processes. Operations on processes. States of a process. Context switching. Creation and termination of a process. Sequential, concurrent, and real-time processes. Lightweight processes (threads).

Credit 4. Management of processes and threads in Linux.

Credit 5. Cooperation and synchronization. The mutual exclusion problem. The semaphores. Communication: shared memory, message passing. Deadlock.

Credit 6. Management of the central processing unit. Scheduling criteria and algorithms.

Credit 7. Memory management. Address spaces. Static and dynamic relocation. Virtual memory and swapping. Memory allocation, paging and segmentation. Management of secondary and tertiary memory .

Credit 8. The file system. File system structure. File attributes, operations and access methods. File allocation. Free space management.

Credit 9. Concurrent programming in C.

Suggested text-books

A. Silberschatz, P. Galvin, G. Gagne, Sistemi operativi – Concetti ed esempi 9/ed. Pearson 2014 – ISBN 9788865183717.

Prerequisites / Co-requisites:

Knowledge of computer architecture. English proficiency.
Organization of teaching: (lectures, tutorials, laboratory, etc.)
The course is structured in classroom lectures and laboratory sessions.
Language
Italian with some materials in English.
Methods and assessment criteria
Written test and oral discussion. The written test is designed to assess the theoretical knowledge acquired during the course and the ability to apply algorithms studied during the course to specific situations. Access to the oral exam is allowed to students that obtain a minimum score of 18/30 in the written test. During the oral discussion, besides assessing the ability to explain the answers given during the written test, the ability to apply the acquired knowledge to solve a specific solution to problems that the teacher will submit to the student will be assessed. Assessment of the knowledge, skills and competences according to Dublin Descriptors: Knowledge and understanding The teacher will assess the student's ability to understand a real problem simulated by the teacher in relation to the use of a modern operating system and to explain the reasons in relation to knowledge acquired during the course. Theoretical and practical knowledge will also be evaluated during an oral discussion. Application of knowledge and understanding in terms of the ability to know how to deal with cases of argumentation and problem solving This skill will be assessed by evaluating the clarity with which an issue submitted to the student is contextualized as well as the ability to find a solution to the problem through the use of specific tools or shell scripts. Making judgments (also with reference to social, scientific and ethical problems) The teacher will assess the ability to describe the different aspects of a real problem and evaluate the advantages and disadvantages of the adopted solution. Communication skills The student’s ability to express complex technical concepts will be evaluated through the use of language clear and understandable to a non-expert user. Learning skills These skills will be evaluated during the written test and the oral discussion on the topics of the course.
Further information
Students can practice for the final test, after the registration and authorization request at: http://dibt.unimol.it/gequiz

STATISTICS FOR TECHNOLOGY

Lecturer	Fabio Divino
ECTS	6

Learning outcome and their consistency with the objectives of the course of study

This is a basic course in statistics for a first-level degree in Computer Science and introduces the student to the applications of statistics in machine learning and data mining problems. The approach is mostly conceptual and application-oriented but without neglecting basic mathematical aspects. In order to improve computational skills, computer sessions in data analysis with R software are organized.

Content of the Program/Course:

Credit 1. Introduction to statistics and application in machine learning and data mining. Introduction to explorative statistics: position, dispersion and shape. PC-lab with R software.

Credit 2. Introduction to probability theory: Kolmogorov's axioms and basic results. Independence and conditional probability. Bayes theorem. Introduction to random variables: probability function, density and cumulative distribution. Moments: expectation and variance. PC-lab with R software.

Credit 3. Linear correlation and linear regression. PC-lab with R software.

Credit 4. Generalized linear models: logistic regression and log-linear regression. PC-lab with R software.

Credit 5. Categorical data analysis: contingency tables and dependence. The ANOVA model. PC-lab with R software.

Credit 6. Introduction to statistical machine learning: supervised and unsupervised methods. PC-lab with R software.

Suggested text-books

D.M. Levine, T.C. Krehbiel, M.L. Berenson, Statistica, Pearson, 2010.

G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning with Application in R, Springer, New York, 2013.

Organization of teaching: (lectures, tutorials, laboratory, etc.)

Lectures and PC-lab with R software

Language

Italian

Methods and assessment criteria

The evaluation test consists in a written test and computer work assessing the level of knowledge acquired by each student in coherence with the learning targets of the course and the mission of the degree program.

ALGORITHMS AND DATA STRUCTURES	
Lecturer	Maurizio Giacci
ECTS	10
Learning outcome and their consistency with the objectives of the course of study	
Acquisition of theoretical and methodological knowledge associated with algorithm design and analysis. The course introduces the fundamental techniques for designing and analyzing algorithms, including computational complexity and correctness concepts, by studying the sorting algorithms, the linear time selection algorithm, the Strassen's matrix multiplication algorithm, the priority queue implementation and the longest common subsequence problem. It also illustrates effects of data organization in algorithm design by studying both the simple data structures (list, queue, stack, graph and tree) and the advanced data structures used to implement an ordered dictionary, to solve the set union problem, to find the minimum spanning tree and to solve the shortest path problem.	
Content of the Program/Course:	
First module – 5 ECTS Credit 1. Algorithm and Computer Programs. Algorithm, problems and computer program. Abstract model of computation. Algorithm computational complexity. Asymptotic notation O, Θ, Ω. Asymptotically optimal algorithms. Complexity of algorithm pseudo-code descripted. Asymptotic notations rules. Simple data structure: list, stack, queue, graph, tree. Credit 2. Recursive algorithm. Recursion. Divide-and-Conquer. Merge Sort. Strassen's algorithm. Recurrence equations. Methods to solve recurrence equations. The master theorem. Credit 3. Heaps. Heap structure. Heap operators. Priority queue. Heapsort. Hash tables. Common hash functions. Handling the collisions: open addressing and separate chaining. Credit 4. Bucket Sort. Lower bound of exchange and comparison-based sorting algorithm. The bucket sort algorithm. The selection problem. The linear time selection algorithm. Credit 5. NP-Completeness. P and NP classes, Reduction algorithm, The NP-C class, The main question: P=NP?, Identification technique for NP-C problems, NP-C problems Second module – 5 CFU Credit 6. Searching problem. Binary search trees, AVL Trees, 2-3 Tree, B-Tree Credit 7. The Set Union problem. The Quick Find algorithm, The Quick Union algorithm, Weighted Heuristic, The off line min problem Credit 8. The Minimum Spanning Tree problem. Introduction. The greedy algorithm. Kruskal's algorithm. Implementing Kruskal's Algorithm using Union-Find data structure. Credit 9. The shortest path problem. Introduction. Dijkstra algorithm. Dijkstra algorithm implementation based on simple queue. Dijkstra algorithm implementation based on min-priority queue. Credit 10. Dynamic algorithms. Introduction, The Longest Common Subsequence	
Suggested text-books	
Appunti di Analisi e Progettazione di Algoritmi – Acciaro, Roselli, Marengo Algoritmi e Strutture Dati – Demetrescu, Finocchi, Italiano – McGraw Hill Introduzione agli Algoritmi e Strutture Dati, sec. ediz. – Cormen, Leiserson, Rivest, Stein–McGrawHill	
Prerequisites / Co-requisites:	
Programming	
Organization of teaching: (lectures, tutorials, laboratory, etc.)	
Lectures. Practice exercises. Laboratory work.	
Language	
Italian	
Methods and assessment criteria	
Students have to pass written test, in order to verify knowledge, understanding and learning skills, and oral examination, in order to verify the ability to apply acquired knowledge.	

SOFTWARE ENGINEERING

Lecturer	Fausto Fasano
ECTS	10

Learning outcome and their consistency with the objectives of the course of study

The student, in line with the objectives of the course of study,will develop specific skills essential for the formation of a professional profile needed to operate at the engineering level in the field of software industry. The course provides an overview of the problems, theory, models, techniques, and technologies that characterize the production and life of the software, with particular focus on the design of object-oriented software.

Knowledge and understanding

The student, at the end of the course, will be aware of the theoretical and practical aspects on the principles of definition, modeling, design, implementation and verification of software systems of medium to high complexity.

Applied knowledge and understanding

Students will broaden their knowledge in a simulated cooperative environment. During the early stages of the course, for the entire duration of the course, each student will be allocated to a team which will be assigned tasks of understanding, analysis, modeling and planning within a software project.

Making judgments

The student will achieve the necessary skills to understand and analyze documents of high and low level projects. Particular emphasis will be paid to the ability of verification, through static and dynamic techniques, the accuracy, completeness, clarity and ambiguity of deliverables produced by their own team as well as the various teams.

Communication skills

Communication and organization as part of a software project are essential elements of a successful project. Therefore, students will be encouraged to organize frequent meetings, both within their team (brainstorming, review meeting, issue resolution, etc.) and publicly, i.e., in the presence of the teacher and other students (final presentation of the project, project status review, etc.). They will also explore opportunities to support project activities provided by communications and collaboration tools that make use of information technologies such as wikis, blogs, social networks, mailing lists, chat and cloud services.

Ability to learn

The student will expand transversal skills related to the application of technological, methodological, organizational and communication knowledge in order to complete a real software project, and will have full autonomy in the choice of tools and technologies that he considers necessary to complete the project and will be encouraged to experiment with new languages and platforms that can improve the success of the project and specialize the personal as well as work team body of knowledge.

Content of the Program/Course:

Credit 1. Basics of Software Engineering. Software engineering. principles, methods, techniques, methodologies and tools.

Credit 2. Software development process. Software life cycle models.

Credit 3. Object-oriented modeling and Unified Modeling Language (UML).

Credit 4. Requirements analysis and specification.

Credit 5. Design of a software system and software architectures.

Credit 6. Design Patterns.

Credit 7. Low level design of a software system. Object Design and model transformations, refactoring, forward and reverse engineering.

Credit 8. Software testing. Black-box testing techniques. Equivalence classes and category partition techniques.

Credit 9. Static review of a software product: software inspections.

Credit 10. Change management in software projects: Software configuration management and rational management.

Suggested text-books

B. BRUEGGE, A.H. DUTOIT, Object Oriented Software Engineering - Using UML, Patterns and Java, 3rd ed., Prentice Hall.

Other recommended textbooks:
R. S. PRESSMAN, Principi di Ingegneria del Software, Mc Graw-Hill Italia.
I. SOMMERVILLE, Ingegneria del Software, 8 ed., Addison Wesley.

Prerequisites / Co-requisites:

Programming.

English proficiency.

Organization of teaching: (lectures, tutorials, laboratory, etc.)

The course is structured in classroom lectures and laboratory sessions.

A team project will also involve students in a simulation of the modeling and development of a software system for small to medium-high complexity projecst. During the project the teacher will assume the role of project manager organizing workgroups and coordinating their activities.

Language

Italian with some materials in English.

Methods and assessment criteria

Presentation of the final project and oral discussion.

The theoretical and practical knowledge acquired during the course will be assessed, with particular focus on the software life cycle phases.

The course involves the construction of a project, generally a group project, during which the students must implement a process of modeling and development of a software system of medium-high complexity. The project will be organized in three phases. The first phase will involve the elicitation and analysis of the requirements of the system to be implemented and must be completed between the end of the first semester and the beginning of the second one. The second phase will involve the design of the final system and will be completed by the end of the second semester. Finally there will be a phase of implementation and testing to be completed by the due date for the final exam. This last step is optional as the course will mainly assess the ability to understand the problem, analyze requirements, model and design the software system as well as the planning of the development, verification and validation phases.

Before the oral discussion, each team must prepare a final presentation of the project to be carried out in the presence of the teacher and eventually the other interested students.

Assessment of the knowledge, skills and competences according to Dublin Descriptors:

Knowledge and understanding

The student's ability to understand the problem and to apply the knowledge to formalize the problem in a requirements specification analysis document of the system will be assessed.

Theoretical and practical knowledge will also be evaluated during the oral discussion.

Application of knowledge and understanding in terms of the ability to know how to deal with cases of argumentation and problem solving

The clarity of the problem is described in terms of the document of the previous point and the ability to find design solutions appropriate and consistent with functional and quality requirements of the system will be assessed.

Making judgments (also with reference to social, scientific and ethical problems) will be evaluated during the whole development process of the project through appropriate project meetings and during the final discussion.

Communication skills will be evaluated at the final presentation stage of the project.

Learning skills will be evaluated during a discussion on the topics of the course.

DATABASES AND INFORMATION SYSTEMS

Lecturers	Rocco Oliveto (1st module) – Remo Pareschi (2nd module)
ECTS	10

Learning outcome and their consistency with the objectives of the course of study

The student, in line with the objectives of the course, will develop essential skills related to fundamentals, concepts, methods and techniques for the design and use of databases and database management systems.

Knowledge and comprehension

The student, at the end of the first module, will be able to understand and use basic concepts and tools related to the design and implementation of relational databases. At the end of the second module, the student will know the extension of the relational data model based on computational logic and in particular its implementation through Datalog and Prolog as well as the use of ontologies in order to represent knowledge in complex domains and their complementarity compared to the relational representation of data, and will have an understanding of the main features of NoSQL databases and of the motivations behind their use, especially in contexts of "Big Data".

Applied knowledge and comprehension

The student will be able not only to design and develop relational databases, but will be able to manage and implement NoSQL databases in "Big Data" contexts.

Autonomy in the capability of judgment

The student will be able to analyze a set of requirements that a database needs to satisfy and adopt an implementation strategy for such requirements that may diverge from those learned in class.

Communication skills

The student will be able to describe a solution to a specific problem with sufficient level of formalism and by using appropriate language.

Ability to learn

The student will be able to evaluate different solution strategies for a given problem and choose the most suitable one in the specific circumstances, while maintaining full awareness of the limitations and strengths of the approach.

Content of the Program/Course:

First module – 5 CFU

Credit 1. Modelling and conceptual design. Entity-relationship data model. Conceptual design.

Credit 2. From conceptual schemata to relational schemata. Creating intermediate schemata. Generation of basic relations.

Credit 3. Normalization of relational DB. Functional dependencies. Normal forms. Information management systems. Databases (DBs) and Database Management Systems (DBMS).

Credit 4. Formal query models on relational DBs. Relational algebra.

Credit 5. SQL. Data Definition Language (DDL) e Data Manipulation Language (DML). Transactions and their properties. Queries of databases in Java: JDBC.

Module 2 – 5 CFU

Credit 6. Summary of the basic assumptions of the SQL world, with particular emphasis on transactions and ACID properties.

Credit 7. Introduction to logic preparatory to the extended representation of data and knowledge. Closed world assumption.

Credit 8. Logical Databases (Datalog and Prolog). Moving beyond relational algebra through the management of recursion and of negation. Multi-mode query execution (backward chaining and forward chaining).

Credit 9. Ontologies and tools for their creation and use. OWL, Protégé, RDFS, SPARQL. NoSQL databases. History and motivation. Key-value databases, column-oriented databases, document databases, graph-oriented databases.

Credit 10. Data processing in the NoSQL world. Basic constructs for parallel cloud processing in the cloud

(MapReduce, HADOOP). Problem of consistency in the face of the continued availability of "big data."
Suggested text-books
Ramez Elmasri, Shamkant Navathe, Sistemi di basi di dati, Pearson/Addison Wesley, 2006. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence (Pramod J. Sadalage, Martin Fowler) Datalog and Recursive Query Processing (Todd J. Green, Shan Shan Huang, Boon Thau Loo) Semantic Web Programming (John Hebeler, Matthew Fisher, Ryan Blace , Andrew Perez-Lopez)
Methods and assessment criteria
Lectures and exercises. The lectures will be used to discuss the general issues related to the design and development of a relational and logical databases, ontological knowledge representations, and NoSQL databases. Class attendance is not mandatory, but is strongly recommended so as to achieve the learning objectives of the course, as it allows faster learning of the terminology and of the basic concepts, thus fostering the cognitive potential needed for self-study of the theoretical contents and of their possible implementation.
Language
Italian.
Methods and assessment criteria
The examination is organized in two parts in accordance with the division of the course into two modules. The first part of the exam consists of a written test and an oral examination. The duration of the written test is 3 hours. In this test the student is required to design a database in SQL language and to implement three queries. This test is designed to assess knowledge and ability to understand and apply the student’s independent judgment skills. The grade of the written test is out of thirty. Students who achieve a grade higher than 18/30 in the written test will gain access to the oral exam, which will take place on the same day of the practical test, in order to emphasize that exam preparation is global, and does not distinguish between preparation for the practical test and another for the oral test. The oral examination consists of the discussion of the design strategies adopted by the student during the written test in order to evaluate communication and learning skills. In case of positive evaluation at the oral exam, the student will be assigned an overall grade (written and oral) out of thirty. This vote represents the overall assessment on the first part of the exam. Students who have obtained an overall grade for the first part of the course of at least 18/30 will gain access to the exam for the second module of the course. This part consists in the completion of a project (on the topics of the second module) and an oral exam. In case of positive evaluation of the project and of the oral exam, the student will be assigned an overall grade (project and oral) out of thirty. This mark represents the overall assessment for the second module of the course. The overall grade of the examination is the average of the grades obtained in the two parts.

HISTORY OF MATHEMATICS

Lecturer	Giovanni Ferraro
ECTS	6

Learning outcome and their consistency with the objectives of the course of study

The goals of the course are to:

- develop the capacity to understand the contemporary world in the larger framework of tradition and history
- interpret the past and relationship between past and present critically
- introduce students to research methods in the history of mathematics

I) Knowledge and understanding.

Knowledge of the main aspects in the history of calculus

II) Applying knowledge and understanding.

At the end of the class, the student will be able to analyse a historical mathematical texts

III) Making judgements.

The student will be able to make judgements about remarkable aspects in history and epistemology of mathematics

IV) Communication skills.

Students will be required to communicate sophisticated theories in both formal and intuitive formats.

V) Learning skills.

The student will be able to understand the nature of the historical process in mathematics.

Content of the Program/Course:

Credit 1. Integration and Differentiation in the 18th and 19th century

Credit 2. The Rise and Development of the Theory of Series up to 1900

Credit 3. Differential equations and Existence theorems from an historical perspective

Credit 4. Quantity, numbers and the process of arithmetization of mathematics

Credit 5. Set theories and foundational problems around 1900

Credit 6. The Bourbakist movement and the mathematics of structures

Suggested text-books

L. Corry, *Modern Algebra and the Rise of Mathematical Structures*, Birkhauser, 2013

V. J. Katz, *A History of Mathematics, an Introduction*, Addison-Wesley, 2005

G. Ferraro, *The rise and development of the theory of series up to the early 1820s*, Springer, New York, 2008

Prerequisites / Co-requisites:

Good knowledge of calculus and algebra

Organization of teaching: (lectures, tutorials, laboratory, etc.)

Lectures

Language

Italian

Methods and assessment criteria

Oral exam.

NUMERICAL COMPUTATION

Lecturer	Giovanni Capobianco
ECTS	6

Learning outcome and their consistency with the objectives of the course of study

Providing knowledge of numerical methods and implementation techniques for the construction of algorithms and codes for efficient resolution of scientific computing problems.

Motivating students to learn about scientific computing environments that may be useful in further studies of computer science.

Content of the Program/Course:

Credit 1. Introduction to Numerical Calculation. Solving real world problems: mathematical models and numerical calculations. Calculation procedure errors: truncation errors and round off errors; Mathematical problem F (x, d) = 0. Well-posed, well-conditioned problems: qualitative and quantitative definitions. Conditioning numbers. Stability of an algorithm. Floating point arithmetics. Epsilon machine. Rounding errors, queuing and cancellation.

Credit 2. Linear systems: bad conditioning; direct and iterative methods; Gaussian elimination method, Jacobi and Gauss-Seidel methods; stability, convergence; computational complexity.

Credit 3. Fitting of data: The interpolation problem; Lagrange interpolating polynomial. Discretization error and propagation. Polynomial and Chebishev nodes; The divided differences and the Newton polynomial; limits of polynomial interpolation; Trigonometric interpolation: construction of the trigonometric polynomial interpolation (dim). Discrete Fourier Transform. FFT: basic idea and algorithm. Curves for graphics; Cubic spline interpolation. Parametric splines. Bezier curves. BSplines; Nurbs

Credit 4. Nonlinear equations: Conditioning of $f(x)=0$: relative and absolute number of conditioning (dim.). Bisection method: formula, study of the convergence (with dim.), Stopping criteria. Linearization of the problem of root finding: chord, tangent and secant methods, regula falsi method. Fixed point iteration. Geometric meaning. Fixed point theorem (with dim.). Linearization methods such as fixed point methods. Convergence conditions for the chord method (dim.), secant (dim.) and Regula falsi. Tangent method (Newton): local convergence theorem, global convergence theorem, criteria for the choice of the first approximation. Error analysis: order of convergence of iterative methods (definition). Order of convergence and number of correct digits. Order of convergence of the fixed point method (with proof). Order of convergence of Newton's method (dim.) and secant method (dim.). Newton's method in presence of multiple roots. Newton-Raphson method for systems.

Credit 5. Least Squares Approximation: the discrete case. Linear regression. Construction. System of normal equations. Linearization of $y=ae^{bx}$; $y=ax^{b}$; $y=ax/(b+x)$.

Credit 6. Scientific computing environments: Scilab / Matlab and Mathematica; language, work environment, scripts and notebooks, functions, graphs, libraries

Suggested text-books

A. Murli: Matematica numerica: metodi, algoritmi e software, Ed. Liguori
A. Quarteroni, R.Sacco, F.Saleri: Matematica Numerica. Ed. Springer
V. Comincioli - Analisi Numerica - Ed. Mc Graw Hill
R.Bevilacqua, D.Bini, M.Capovani, O.Menchi, Metodi numerici, Zanichelli.

Prerequisites / Co-requisites:

It is desirable, but not required, that the student has already studied / passed the Mathematics exam in order to understand the topics covered in the course.

Organization of teaching: (lectures, tutorials, laboratory, etc.)

Lectures. During the lessons theories are proposed and numerous examples and exercises conducted using Scilab.

Language

Italian

Methods and assessment criteria

A written exam and an oral exam which can be done in the same exam session or in two different periods.
To gain access to the oral the student must have obtained at least 16/30 in the written exam.
Written exam: two or three exercises lasting a maximum of 80 minutes.
Oral exam: questions about definitions, statements, demonstrations, concepts, graphics, calculations on the topics covered in the course.

Assessment of the knowledge, skills and competences according to Dublin Descriptors:

Descriptor	Topics	Assessment Methods
Descriptor 1 - Knowledge and understanding;	Sources and errors propagation. Numerical methods for solving linear and nonlinear equations. Numerical methods for the approximation of data and functions. Matlab / Scilab.	Questions on the issues in the oral exam
Descriptor 2 – Applying knowledge and understanding;	Linear systems with direct and iterative methods . Calculation of solutions of nonlinear equations with iterative methods, approximation of data and functions, and through exercises both direct and inverse problems or questions when instrumental to troubleshooting. Estimation errors, non-conditioning calculation and stability in the solution of linear systems of nonlinear equations and in the approximation of data and functions. Implementations of methods / algorithms in a Scilab / Matlab	Exercises to do in the written exam and possibly also in the oral exam
Descriptor 3 – Making judgements;	Solving exercises and implementing algorithms even differently from how they have been learned in the lessons. Preference of theorem proof performed differently from the way shown by the teacher.	Exercises to do in the written exam and possibly also in the oral exam
Descriptor 4 – Communication skills;	Using the appropriate language in enunciating theorems, demonstrations, specify definitions.	Questions on the issues in the oral exam
Descriptor 5 – Learning skills.	Having the ability to identify the numerical method useful in solving a particular problem or to estimate the errors.	Exercises to do in the written exam and possibly also in the oral exam

PHYSICS

Lecturer	Ciro Marmolino
ECTS	7

Learning outcome and their consistency with the objectives of the course of study

The course is an elementary level first-year physics course, aimed at university students in biology, computer science and natural sciences. The goal is to provide the knowledge, skills and tools for understanding simple physical phenomena that occur in nature and for describing some simple technical applications of physics that students will need to continue their scientific studies and, at the same time, to convey the excitement of the physicist's quest to understand nature at its deepest level. It is based on the conviction that all students can and should become familiar with basic ideas of physics and, for this reason, it emphasizes principles still having contact with everyday life and practical situations.

Although in an elementary physics course, like this one, the student brings no mathematics beyond simple algebra, it is becoming increasingly apparent that for maximum insight into physics at any level a student should not be deprived of the help offered by the use of limits. Therefore, during the course, the derivative is defined, illustrated, and used as the limit of a ratio and the definite integral as the limit of a sum. Calculus is developed and explained as the need arises, and it requires only a knowledge of high school algebra.

Physics is also a classic field of learning methodology and scientific language and therefore its study can't be seen only for instrumental purposes but also for educational purposes.

The student should strive for personal mastery over the following knowledge and skills and must become proficient to apply them to real, however simple, situations.

1) Equations of motion for a particle in a uniformly accelerated motion and executing a simple harmonic motion.
2) Newton's laws of motion.
3) Conservation of momentum and energy.
4) Structure and properties of matter.
5) Electric and magnetic field.

Content of the Program/Course:

Credit 1. Precision, measurement and notation in physics. The SI absolute system of units. Other systems and conversion factors. Significant figures. Powers of ten, scientific notation and orders of magnitude. Kinematics of a point mass. Velocity. Acceleration. Uniform and uniformly accelerated motion in straight line. Acceleration due to gravity. Vectors. Composition of motions. The projectile's trajectory. Centripetal acceleration. Earth's satellites

Credit 2. Dynamics of a point mass. Force, the cause of acceleration. Newton's laws. Definition of momentum and Newton's second law in terms of momentum. Isolation of bodies in problem solving. Dynamics of the inclined plane. Atwood's machine. Friction and coefficients of static and kinetic friction. Gravitation: Newton's law of universal gravitation. Kepler's laws. Dynamics of planetary motions-the Newtonian synthesis.

Credit 3._Elastic constants-Hooke's law. Simple harmonic motion. Period of simple harmonic motion: the reference circle. The simple pendulum. Kinetic energy. Work and work done against a variable force. Theorem of the kinetic energy. Conservative forces and potential energy. Conservation of energy. Power; efficiency. Gravitational potential energy. Escape velocity. Center of mass. Momentum and kinetic energy in collision. Elastic and inelastic collisions in one dimension.

Credit 4._The structure of matter and the electric charge. Electrification of bodies. Conductors and insulators. Electrostatic induction. Coulomb's law. The electric field. Lines of force. Gauss' law and the electric field of different charge distributions. Electric potential energy. Potential difference. Capacitance. The use of capacitors in circuits.

Credit 5. The energy method in electricity. Electromotive force. The electric circuit-Joule's law. Conventional current. Ohm's law. Resistivity. Microscopic view of the electric current. Electric circuits in direct current. Kirchhoff's laws. Resistors in series and in parallel. Emf's in series and in parallel. RC Circuit.

Credit 6._Magnetic force. Magnetic field. Force on a current segment in a magnetic field. Direction of

magnetic force on a moving charge. Mass spectrometer. Sources of magnetic fields. Ampere's law. Magnetic field due to a long straight wire and inside a solenoid. Theory of magnetism. Magnetic flux. Faraday's law and induced electromotive force. Inductance. RL circuits. Displacement current and Maxwell's equations. Credit 7. Numerical exercises for the preparation of the written test.
Suggested text-books
Textbook: GIANCOLI: *Fisica* (seconda edizione), Casa Editrice Ambrosiana. Other books at, more or less, the same level and useful for references are: HALLIDAY D., RESNICK R., WALKER J., *Fondamenti di Fisica*, Casa editrice Ambrosiana, Milano. JEWETT J.W. & SERWAY R. A., *Principi di Fisica* Vol. 1 (IV Edizione), EdiSES, Napoli KESTEN P.R., TAUCK D. L., *Fondamenti di Fisica* Vol. 1, Zanichelli, Bologna.
Prerequisites / Co-requisites:
It is advisable to consider the course in mathematics as a prerequisite.
Organization of teaching: (lectures, tutorials, laboratory, etc.)
Formal teaching, with lectures and numerical exercises.
Language
Italian
Methods and assessment criteria
2-hour written test with multiple-choice questions and an oral exam. The written and oral exam are taken on the same day, to emphasize that exam preparation is global, and does not distinguish between preparation for the written and one for the oral exam. Knowledge and understanding of the topics discussed are achieved only when the student can discuss (showing his/her communication and learning skills) and apply effectively the basic ideas of physics to simple situations (showing his/her ability to make judgments and to apply his/her knowledge and understanding). The written test is selective and, to be admitted to the oral exam, it is necessary to get a minimum score of 18/30. During the oral section, the student will be asked to discuss one of the problems assigned during the course to practice the art of problem solving.

COMPUTER NETWORKS AND SECURITY

Lecturer	Mario Petrone
ECTS	12

Learning outcome and their consistency with the objectives of the course of study

The first module of the course of Computer Networks and Security aims to make students understand and learn computer architecture and key design issues of modern computer networks. For this purpose we will explore the features and functioning of the most common network protocols. Students will also acquire network logic design capabilities as well as analysis and application development capabilities.

The aim of the second module of the Computer Networks and Security course is to make students understand and acquire the main issues of computer network security.

The most important categories of vulnerability will be discussed, by analyzing management strategies and considering limits and requirements to which these strategies have to conform. In particular, the second module aims to provide a capacity for large-scale analysis of network security through the combination of low-level technical details and considerations on a larger scale.

Content of the Program/Course:

First part – 6 ECTS

Credit 1. Introduction to Computer Networks. Layered architecture: layering, protocols, reference model ISO/OSI, Internet architecture and TCP/IP model.

Credit 2. Physical layer and data-link. The transmission of signals; The linking function; Channel-access techniques; Ethernet (IEEE 802.3), WI-FI and Point to Point connections.

Credit 3. Network layer. The routing function; Static routing, dynamic routing, and hierarchical routing; adaptive routing algorithms; Internet Protocol: IPv4 and IPv6; routing protocols: RIP, OSPF, BGP, Multicast.

Credit 4. Transport layer. The transport function; reliability management; flow and congestion control; transport layer in the Internet: UDP and TCP.

Credit 5. Application layer. Functions of session, presentation and application; the Domain Name System (DNS); some protocols (HTTP, FTP, SMTP and POP3); systems content delivery: P2P.

Credit 6. Programming of network services. The Unix socket API (BSD socket); structures and basic functions.

Second part – 6 ECTS

Credit 1. Introduction to Network Security. Services, mechanisms and attacks. A model for network security.

Credit 2. Cryptography. Conventional encryption and message confidentiality. Public key encryption and message authentication.

Credit 3. Authentication. Email security. IP Security: IP Security Architectures, IP Security Protocol, Virtual Private Network

Credit 4. Web security: Secure socket layer, Transport layer security, Secure electronic transaction.

Credit 5. Security of network management: Basic Concepts of SNMP, Facility SNMP community.

Credit 6. Security System: Intrusions and virus, Firewall, Trusted Systems.

Suggested text-books

J.F. Kurose, K.W. Ross –"Reti di Calcolatori e Internet", Pearson Education Italia, 2013

W. Stallings, Sicurezza delle reti, Milano, Prentice Hall, 2007.

Teacher supplied material

Online material

Organization of teaching: (lectures, tutorials, laboratory, etc.)

Lectures, tutorials

Language

Italian

Methods and assessment criteria

The exam consists of a written test lasting 4 hours and a mandatory oral test, if the written test scores between 15/30 and 17/30, or optional, if the written text scores at least 18/30. This exam's overall score is the sum of the written test and the possible oral exam scores.

Assessment of the knowledge, skills and competences according to Dublin Descriptors:

Knowledge and understanding

Acquisition of fundamental knowledge about architectures and key issues of the design and management of modern computer networks also with regard to security.

Ability to apply knowledge and understanding

Ability to assess the operational mechanisms of computer networks and security levels required on the basis of a full understanding of the technical solutions used.

Making judgments

Ability to analyze and evaluate the architecture of a computer network according to the components that constitute it.

Ability to analyze and evaluate the levels of risk and security of a computer network.

Communication skills

Ability to describe design solutions for a computer network through the analysis of the technical specifications provided. Ability to describe design solutions for the security of a computer network through the analysis of the technical specifications provided.

Learning ability

Self-learning skills by consulting advanced literature. Ability to attend master degree courses as well as first-level master's programs using acquired knowledge

WEB AND MOBILE PROGRAMMING

Lecturers	Fausto Fasano (1st module) – Rocco Oliveto (2nd module)
ECTS	10

Learning outcome and their consistency with the objectives of the course of study

The course introduces students to basic and advanced concepts for the construction of professional level multimedia applications with special focus on both server and client technologies for the Web through the J2EE platform. The student will understand basic aspects of the main markup languages. The set of technologies underlying DHTML and AJAX will be presented.

Students will be introduced to basic and advanced concepts related to the creation of applications for mobile devices. The set of technologies underpinning the development of applications for devices based on iOS and Android will be presented. Finally, an overview on the implementation of web services and communication between mobile devices and web services will be provided.

Knowledge and understanding

The student, at the end of the course, will know the theoretical and practical aspects related to the technologies used to build web applications as well as native applications for mobile devices.

Applied knowledge and understanding

The student will be able to put into practice the acquired knowledge in a software project assigned by the teacher.

Making judgments

The student will achieve the necessary skills to understand and analyze third-party applications. They will learn how to evaluate the differences between the multiple technologies available and will be able to choose the best one for the solution of a specific problem.

Communication skills

Communication and organization are essential elements for successful software projects. Therefore, students will be encouraged to organize frequent meetings, both with the other students and the teacher. They will also explore opportunities to support the project activities provided by the many communication and collaboration tools that make use of information technologies such as wikis, blogs, social networks, mailing lists, chat and cloud services.

Ability to learn

The student will develop transversal skills related to the application of technological, methodological, organizational and communication knowledge to the realization of a real software project, and will have full autonomy in the choice of tools and technologies considered necessary to complete the project and will be encouraged to experiment with new languages and platforms that can improve the success of the project as well as expand personal as well as work team body of knowledge.

Content of the Program/Course:

Credit 1. The Web and the HTTP protocol. Markup languages. HTML, CSS, XML and XML Schema, XPath and XSLT

Credit 2. The server side programming with Java. Java Servlet, sessions, cookies and context, Expression Language, Custom Tags, Property files, Resource Bundles

Credit 3. The client-side programming with Java Server Pages and Javascript. jQuery

Credit 4. AJAX (Asynchronous JavaScript and XML). The MVC model and the web 2.0

Credit 5. Design of multimedia applications. Extensions of the UML modeling language to the web

Credit 6. Introduction to mobile programming.

Credit 7. iOS Programming: Introduction to Objective-C. Programming in Android: Java Mobile.

Credit 8. Interface design for mobile devices.

Credit 9. Using APIs for interacting with the mobile device.

Credit 10. Web services realization. Communication between mobile devices and web services.

Suggested text-books

Anders Moller, Michael Schwartzbach, *Introduzione alle tecnologie web*, Addison Wesley, 2007, ISBN 9788871923741

Maximiliano Firtman, *Programmazione per il web mobile*, Tecniche Nuove, 2011

Core Servlets and JSP, liberamente scaricabile online all'indirizzo:
http://csajsp-chapters.corewebprogramming.com

Prerequisites / Co-requisites:

English proficiency.

The student should follow the course after having acquired basic knowledge on programming, in general, with particular reference to the Java language. It is also advisable to have acquired knowledge related to software engineering.

Organization of teaching: (lectures, tutorials, laboratory, etc.)

The course is structured in classroom lectures and laboratory sessions.

The students will be assigned an individual or team project, which makes use of the technologies studied during the course.

Language

Italian with some materials in English.

Methods and assessment criteria

Presentation of the final project and oral discussion.

The theoretical and practical knowledge acquired during the course will be assessed, with particular focus on the use of the technologies discussed during the course.

The course involves the construction of an individual or team project, during which the student will have to design and implement a software system using some of the technologies studied during the course.

Before the oral discussion, each student will prepare a presentation of the final project to be presented to the teacher and possibly to other interested students.

Assessment of the knowledge, skills and competences according to Dublin Descriptors:

Knowledge and understanding

The student's ability to understand the problem will be assessed, by applying the most appropriate technologies among those acquired during the course. The project must be accompanied by system requirement specification and analysis document.

Theoretical and practical knowledge will also be evaluated during the oral discussion.

Application of knowledge and understanding in terms of the ability to know how to deal with cases of argumentation and problem solving

The clarity with which the problem is described will be assessed as well as the ability to find design solutions appropriate and consistent with the functional and quality requirements of the system.

Making judgments (also with reference to social, scientific and ethical problems)

will be evaluated during the internal development of the project at specific project meetings and during the final discussion.

Communication skills will be evaluated during the final presentation of the project.

Learning skills will be evaluated during a discussion on the topics of the course.

GEOGRAPHIC INFORMATION SYSTEMS

Lecturer	Paolo Di Martino
ECTS	6

Learning outcome and their consistency with the objectives of the course of study

Learning outcomes

The student, according to the objectives of the bachelor's course, will develop specific skills by integrating knowledge from other modules, practise management of digital geographic data, and apply appropriate methodologies and analysis based on project-work and GIS applications.

Knowledge and understanding

The student will be able to gain the theoretical and practical knowledge on the fundamentals of cartography, the use of vector and raster data, models of spatial analysis and open source GIS geo-database.

Applying knowledge and understanding

The student will work on individual and team based project work, planning and organizing in different steps: collection of geo-database from government open source data, understanding the complexities of the sources and management of spatial database, spatial analysis and the implementation of the results on WEB GIS sites.

Making judgements

The student will achieve the necessary skills to evaluate the complexity of vector and raster cartographic source methodologies on different study cases.

Communication skills

The student will be able to communicate design ideas to experts and non-expert peers correlating theory with practical examples of analysis and representation of geographic phenomena.

Learning skills

The student will develop transversal knowledge on information technology and communication skills for dedicated application and individual and team based project organization (stages, deadlines).

Content of the Program/Course:

Credit 1. Fundamentals of cartography, projection systems and geo-referencing of proximal and remote sensing images.

Credit 2. Elements of geographic information systems and land use, land-cover analysis. Open source GIS software: vector and raster model, database and topology, geo-processing, Digital Elevation Model analysis: slope, aspect, hill-shade, terrain.

Credit 3. Spatial information: data main stages of acquisition and pre-processing.

Credit 4. Application on study cases in national and international projects. Using geo-database to support management and planning with open source software: examples and applications.

Credits 5 and 6. Project work: analysis, processing of geographic data and remote sensing images with open source software. WEB GIS.

Suggested text-books

Materials: slides and papers.

Prerequisites / Co-requisites:

Basic knowledge of the English language.

Organization of teaching: (lectures, tutorials, laboratory, etc.)

The course is structured in lectures and laboratory work.

Language

Italian

Methods and assessment criteria

Oral (discussion or presentation).

Knowledge and understanding

The theoretical and practical knowledge acquired during the course will be assessed, with particular reference to the management and analysis of raster and vector geodatabase and integration with information technology and communication.

The course includes the development of a project, usually in groups, on the implementation of open source GIS based on governmental open data available on national and international websites.

The final assessment will be based on the complexity of project work and the geo-data used, (applying knowledge and understanding), analysis from geo-processing, structuring the database (making judgements and learning skills) and quality of the representation of geographical phenomena (communication skills).

The final exam will be based on a presentation of the project and an oral examination based on questions. The final grade consists of project evaluation (80%) and oral evaluation (20%).

The final assessment for students unable to attend the whole series of module lectures will be based on an oral examination (questions and short exercises) of the course materials. In particular, the integrated knowledge of information technology and communication and GIS systems will be evaluated as well as practical examples of GIS applications, the fundamentals of cartography and projection systems, structure and management of the database, vector and raster models, geo-processing and major sources of open data GIS.

SOFTWARE EVOLUTION

Lecturer	Rocco Oliveto
ECTS	6

Learning outcome and their consistency with the objectives of the course of study

The student, in line with the objectives of the course, will develop skills related to quality and evolution of complex and large software systems.

Knowledge and understanding

At the end of the course, the student will have acquired methodologies and techniques for the efficient and effective management of changes in a software system, with particular emphasis on issues related to source code quality, impact analysis, defect prediction, refactoring and software testing.

Applied knowledge and understanding

The student will be able to analyse the structure of a software system in order to evaluate its quality. The student will also be able to analyse the evolution of a software system in order to evaluate quality grade and to design maintenance activities to improve quality.

Making judgments

The student will be able to analyse the quality of a software system and its evolution with the aim of defining evolution strategy for the system under development that could be different from those learned during the course.

Communication skills

The student will be able to describe the quality of a software system and its maintenance process with a high level of formalism and by using appropriate language.

Learning skills

The student will be able to evaluate the different evolution strategies for a given software system so as to improve its quality and be able to select the most suitable in specific circumstances, while maintaining awareness of the limitations and strengths of the selected solution.

Content of the Program/Course:

Credit 1. Evolution of software systems: definitions and motivations. Legacy system. Change management: versioning systems (e.g., SVN or Git) and bug/issue tracking systems (e.g., Bugzilla).

Credit 2. Impact analysis and traceability. Information Retrieval techniques to support impact analysis.

Credit 3. Metric and anti-pattern. Improving the quality of a software system: refactoring and re-modularization.

Credit 4. Regression Testing: selection and prioritization of test cases. Search-based software testing.

Credit 5. Empirical software engineering. Impact assessment of new features/technologies on the business value of a software system.

Credit 6. Techniques for mining software repositories.

Suggested text-books

Ian Sommerville, Ingegneria del Software, Pearson Education, 8 Edizione, 2007.

Martin Fowler, Refactoring: Improving the Design of Existing Code, Addison Wesley Longman, 1999.

Scientific papers provided by the lecturer during the course.

Prerequisites / Co-requisites:

Programming

Organization of teaching: (lectures, tutorials, laboratory, etc.)

Lectures and tutorials.

Attendance is not mandatory, but it is strongly recommended in order to effectively achieve the objectives of the course, as it allows terminology and basic concepts to be learned more easily, enhancing cognitive capabilities needed for self-study, theoretical content and their possible implementation.

Language

Italian

Methods and assessment criteria

The examination is conducted orally as it aims to evaluate the students' communication skills, their knowledge and understanding through questions regarding the topics covered in lectures. The oral exam will also aim to assess knowledge and ability to understand and apply independent judgment through practical questions that require the practical application of acquired knowledge to real problems.

ARTIFICIAL INTELLIGENCE	
Lecturer	Remo Pareschi
ECTS	6
Learning outcome and their consistency with the objectives of the course of study	
The course aims to give students an introduction to the basic concepts of Artificial Intelligence. Applications will be explored and explained in the domains of games, planning, robotics and natural language understanding.	
Content of the Program/Course:	
Credit 1. Introduction to Artificial Intelligence: symbolic, non-symbolic and hybrid approaches. Credit 2. The search for solutions to explore spaces of states. Credit 3. Problem solving through evolutionary mechanisms of computation. Credit 4. Multi-agent systems and distributed artificial intelligence Credits 5 - 6. Applications: games, planning, robotics, natural language processing.	
Suggested text-books	
Artificial intelligence. A modern approach: 1 di Stuart J. Russell, Peter Norvig, Pearson International, 2010	
Organization of teaching: (lectures, tutorials, laboratory, etc.)	
During the lectures the general theme of artificial intelligence will be introduced and contexts of problem solving will be identified where the diverse AI techniques illustrated within the course are applicable.	
Language	
Italian	
Methods and assessment criteria	
Presentation and discussion of a project-work.	

COMMUNICATION SKILLS FOR COMPUTER SCIENTISTS

Lecturer	Rocco Oliveto
ECTS	3

Learning outcome and their consistency with the objectives of the course of study

The skill of problem solving is an important skill for Computer Science students. But it is equally important to develop strong interpersonal skills. Students must be able to communicate effectively with people as well as with computers. In this course, the student, in line with the objectives of the course, will develop skills related to the effective presentation of a research product or commercial product in information technologies.

Knowledge and understanding

The student, at the end of the course, will have acquired methodologies and techniques for effective oral and written presentations.

Applied knowledge and understanding

The student will be able to design and develop an oral presentation and a written report on a research or commercial product in information technologies.

Making judgments

The student will acquire his own style of presentation and will be able to adapt it to suit the audience.

Communication skills

The student will be able to effectively communicate the results of a research or commercial product.

Ability to learn

The student will be able to design and develop an effective oral and/or written presentation for a specific research or commercial product.

Content of the Program/Course:

Credit 1. Written communication skills.

Credit 2. Oral communication skills.

Credit 3. Project work: the presentation of a product.

Suggested text-books

Materials provided by the lecturer during the course

Organization of teaching: (lectures, tutorials, laboratory, etc.)

Lectures and tutorials.

Attendance is not mandatory, but it is strongly recommended in order to effectively achieve the objectives of the course, as it allows the terminology and basic concepts to be learned faster, enhancing cognitive capabilities needed for self-study, the theoretical content and their possible implementation.

Language

Italian

Methods and assessment criteria

The exam includes a practical test. The student chooses a product or a specific topic of interest, develops a presentation on the selected product/topic and presents it in public. The test will aim to evaluate the student's knowledge, ability and communication skills.

VISUAL COMMUNICATION

Lecturer	Piero Barlozzini
ECTS	3

Learning outcome and their consistency with the objectives of the course of study

Through project experiences, the course aims at introducing the methodologies, the strategies and the design techniques in the field of visual communication.

Knowledge and understanding

The student, at the end of the course, will know methods and design techniques in the field of visual communication. In addition, the student will acquire knowledge on the main topics of graphic design and on widely used computer graphics software.

Applied knowledge and understanding

The student will be able to plan, develop, and implement a simple project in the field of visual communication, such as book layouts, advertising posters, or covers of a software product.

Making judgments

The historical perspective on the graphic design provided by the course aims at expanding the ability of students to (i) investigate the future; and (ii) understand how to adapt their own design method to specific domain.

Communication skills

The student will be able to communicate and argue professionally on the rationale behind their choices and justify them from formal, technical, scientific and theoretical point of view.

Ability to learn

The student will be able to knowingly face the next steps and more complex designs in the field of visual communication, using and implementing the acquired tools.

Content of the Program/Course:

Credit 1. Perception and visual communication
Credit 2. Work methodologies and computer graphics software
Credit 3. Graphical design of a poster

Suggested text-books

R. ARNHEIM, *Pensiero visivo. La percezione visiva come attività conoscitiva*, Einaudi, Torino, 1974
M. BARAGHINI, D. TURCHI, (a cura di), *Farsi un libro, Biblioteca del Vascello, Roma, 1990.*
M. SPERA, *La progettazione grafica tra creatività e scienza*, Gangemi Editore, Roma, 2001
J. TSCHICHOLD, *La forma del libro*, Edizioni Silvestre Bonnrd, Milano, 2003.
EDIMATICA, *Photoshop*, Apogeo, Milano, 2008
L. SANTAPAGA, M., TRASI, M., *AutoCAD*, Apogeo, Milano, 2006

Organization of teaching: (lectures, tutorials, laboratory, etc.)

The course is organized in two parts. In the first part students are engaged in lectures, while in the second part they put into practice the gained knowledge producing some graphics boards.

Language

Italian

Methods and assessment criteria

The exam is based on a written and oral test on the topics of the lectures. Such tests are preceded by a discussion on the projects prepared by the student at the end of course. Thus, the exam will aim to evaluate the level of knowledge acquired by students, their ability to work in team and their communication skills.

COMPUTER ETHICS

Lecturer	Barbara Troncarelli
ECTS	3

Learning outcome and their consistency with the objectives of the course of study

The course aims to achieve the following objectives:
1) knowledge of some ethical and social issues related to information and communication technology, especially about origins and development of the discipline so-called "computer ethics", as well as professional ethics and role of information technology in the global society;
2) autonomous judgments concerning duties and codes of conduct in computer science, according to ethical and legal responsibility for professional sectorial activities.

This course will be in line with the objectives of the "Dublin Descriptors". Through appropriate knowledge of basic principles of computer ethics, it is possible to develop understanding of rules and apply that knowledge for the protection of personal data and observance of professional ethics.

Content of the Program/Course:

Credit 1. "Computer ethics": origins and development; information ethics as a branch of applied ethics concerning the social impact of ICT.

Credit 2. Ethical principles in ICT regulation; professional deontology.

Credit 3. Social changes produced by information technology, so-called "digital divide" phenomenon. Computer ethics and Internet governance.

Suggested text-books

Reference book:

S. Di Guardo, P. Maggiolini, N. Patrignani (edited by), *Etica e responsabilità sociale delle tecnologie dell'informazione. Vol. 1. Valori e deontologia professionale*, Angeli, Milano, 2010 (part II and III).

Slides and papers of interest presented in the lessons.

Further bibliographic references:

K.E. Himma, H.T. Tavani (edited by), *The Handbook of Information and Computer Ethics*, John Wiley & Sons, Hoboken 2008;

N. Patrignani, *Computer ethics. Un quadro concettuale*, in "Mondo digitale", n. 3, 2009, pp. 55-63;

P. Maggiolini, *Computer ethics. Un approfondimento*, in "Mondo digitale", n. 45, 2013, pp. 1-15;

M. Suerz, *Etica del virtuale*, in "Rivista di scienze della comunicazione", n. 1, 2012, pp. 45-53.

Organization of teaching: (lectures, tutorials, laboratory, etc.)

- Lectures.
- Multiple choice tests during the course.
- Brief presentations in powerpoint, optionally performed by students in classroom, individually or in groups, on topics proposed by the teacher.
- Students attending class will have the possibility to carry out, at the end of the course, a written exercise on topics of the lessons, with multiple choice and open questions, as useful feedback activity and incentive to a good preparation for the final exam.

These methods will make use of a tool to support teaching, so-called "aula virtuale", used by the teacher to provide students with detailed information about the program and bibliography, to transmit on-line slides and other materials, to indicate some useful links, and to give information on teaching activities.

Attending class is not mandatory, but it is strongly recommended in order to achieve the educational objectives, because this allows terminology and fundamentals of computer ethics to be learned more easily, by encouraging capabilities needed to tackle conceptual contents and their possible implementation.

Language

Italian

Methods and assessment criteria

The final exam is conducted orally for all students: they will have to answer some questions on the themes of the course. The exam is aimed at evaluating the results, also through questions on topics chosen by the student.

The oral examination appears to be a suitable way to achieve the following objectives and expected results: 1) knowledge and understanding of theoretical contents; 2) proper judgments not only through the study of social and moral impact of information technology, but also through a personal exposition of concepts and evaluations. This tends to increase in the student a more adequate understanding on professional sectorial ethics. In particular, the oral exam, in which the student may be required to supplement the answers with a topic of his choice, is a way to pursue two objectives formulated by the "Dublin Descriptors": 1) to encourage judgments making, plausible decisional acts and assessments; 2) ability to expose (communication skills) as indispensable capabilities, especially in the professional world today. It's thus possible to face future work experiences with the aid of a broader preparation.

COMPUTATIONAL METHODS FOR OPTIMIZATION

Lecturer	Fabio Divino
ECTS	3

Learning outcome and their consistency with the objectives of the course of study

The course provides undergraduate students with an introduction to optimization and computational methods as well as the most relevant operational Research methods. The approach is mostly conceptual and application-oriented but without neglecting basic mathematical aspects. In order to improve computational skills, computer sessions with R software are organized.

Content of the Program/Course:

Credit 1. Introduction to Optimization and Operational Research. Linear Programming: geometrical and algebraic aspects. The Simplex algorithm. PC-lab with R.
Credit 2. Graphs and networks. Flow and path optimizations. PC-lab with R.
Credit 3. Elements of queueing theory, simulation, game theory, decision theory. PC-lab with R.

Prerequisites / Co-requisites:

Mathematics

Suggested text-books

F. S. Hillier, G. J. Lieberman , Ricerca Operativa, McGraw-Hill, Milano, 2010.
C. Vercellis, Ottimizzazione, McGraw-Hill, Milano, 2008.
S. M. Iacus, G.Masarotto, Laboratorio di Statistica con R, MacGraw-Hill, Milano, 2003.

Organization of teaching: (lectures, tutorials, laboratory, etc.)

Lectures and tutorials in PC-lab with R.

Language

Italian

Methods and assessment criteria

The evaluation test consists in a presentation and a computer elaboration assessing the level of knowledge acquired by each student in coherence with the course learning targets and degree program mission.

MATHEMATICAL METHODS IN SCIENCE

Lecturer	Ciro Marmolino
ECTS	6

Learning outcome and their consistency with the objectives of the course of study

The course, an introduction to mathematical methods, is concerned with the construction, analysis, and interpretation of mathematical models that shed light on simple but significant problems in natural sciences. It is intended to provide material of interest to students in biology, computer science and natural sciences at the undergraduate level.

The lectures will focus on: 1) Simple physical problems that arise from natural sciences; 2) The relation of mathematics to sciences and of sciences to mathematics; 3) Differential and integral calculus with special emphasis on differential equations, since without analysis the idea of how mathematics is applied to the sciences is certainly inadequate.

The process of using mathematics to increase scientific understanding will be divided into the following three steps: 1) The formulation of the scientific problem in mathematical terms; 2) The solution of mathematical problems thus created; 3) The interpretation of the solution and its empirical verification in scientific terms.

The student should strive for personal mastery over the following skills and must become proficient at them.

1) Solve algebraic equations of first and second degree, as well as a set of linear equations.
2) Solve problems of analytical geometry.
3) Compute the derivative of elementary functions.
4) Know the most basic methods of evaluating single integrals.
5) Model a simple system to obtain a first order ODE. Visualize solutions using direction fields and isoclines, and approximate them using Euler’s method.
6) Solve a first order ODE by separation of variables and by some substitution methods, solve a linear first order ODE by the method of integrating factors.
7) Calculate with complex numbers and exponentials.
8) Solve a constant coefficient second order linear differential equation by the method of undetermined coefficients.

Content of the Program/Course:

Credit 1. Compendium of Fundamental Principles. Powers. Progressions. Algebraic equations of first and second degree. Systems of equations. Dimensional analysis. Experimental errors and significant digits. The statistics in the counts. Multiplication (division) of experimental data. Addition (subtraction) of experimental data.

Credit 2. Analytic geometry, functions and graphs. Equations of curves in the plane (line, circle, ellipse, hyperbola and parabola). Functions and graphs. Elementary functions. Trigonometric functions. Resolution of a plane triangle. Measurements: applications of triangulation.

Credit 3. Derivatives and integrals. Definition of derivative. Derivation rules. Derivatives of trigonometric functions; derivative of the logarithmic function and the exponential function. Problems of maximum and minimum (the Fermat's principle). The method of least squares. Indefinite integrals. Definite integrals. Fundamental theorem of calculus. Gravitational attraction between extended bodies.

Credit 4._Differential equations of the first order. Differential equation of the first order as directional fields. Isoclines and integral curves. Separation of variables. Linear differential equations of the first order. Bernoulli equation and other simple substitution methods. Introduction to numerical solution of differential equations: Euler method.

Credit 5. Problems that lead to differential equations of the first order. Geometric problems: fluid in rotation. Dynamical problems: motion of a particle in a straight line: free fall and with different models of viscous resistance. Problems of growth and decomposition: Carbon-14 tests. Population models. Malthusian growth and other applications of exponential growth. Logistic model and other applications of the logistic model.

Credit 6. Linear differential equations of order n. Complex numbers. Differential equations of the second order with constant coefficients. Solution by the method of undetermined coefficients. Problems that lead to differential equations of the second order. Interacting populations: the model of Lotka-Volterra.

Suggested text-books

1) Lecture notes.
2) Davidson Ronald C.: Metodi Matematici per un corso introduttivo di fisica, EdiSES (for the first three credits)
3) To a level slightly more advanced, but rich with examples of biological interest
   Gaeta Giuseppe: Modelli matematici in Biologia, Springer.

Prerequisites / Co-requisites:

It is advisable to consider first-year courses in physics and mathematics as a prerequisite.

Organization of teaching: (lectures, tutorials, laboratory, etc.)

Formal teaching, with lectures and numerical exercises, possibly using widely available computer algebra systems.

Language

Italian

Methods and assessment criteria

3-hour written exam.

In reference to the above objectives of the course and to the different knowledge and skills required by students, it consists in the solution of 8 exercises of different levels of difficulty and evaluation:

- 4 exercises, for a maximum score of 10 points, consisting of simple operations of derivation or integration, or of simple applications of concepts of analytical geometry;
- 2 exercises, for a maximum score of 10 points, involving the solution of a differential equation of the first or second order by the analytical techniques discussed;
- 2 problems, for a maximum score of 15 points, that require to model a simple system to obtain a first order ODE to be solved.

<table>
<tr><th colspan="2">FUNDAMENTAL CONCEPTS OF CHEMISTRY AND NEW MATERIALS</th></tr>
<tr><td>Lecturer</td><td>Vincenzo De Felice</td></tr>
<tr><td>ECTS</td><td>3</td></tr>
<tr><td colspan="2">Learning outcome and their consistency with the objectives of the course of study</td></tr>
<tr><td colspan="2">The student acquires fundamental knowledge on Chemistry.
The course is intended to provide necessary information on the following topics:
- Composition of material objects in the surrounding territory;
- Dependence of macroscopic properties of substances on their composition;
- Polymeric materials;
- Nanotechnology.</td></tr>
<tr><td colspan="2">Content of the Program/Course:</td></tr>
<tr><td colspan="2">Credit 1. Atomic structure. Bohr's theory of the hydrogen atom. Quantum numbers and atomic orbitals. Electron configurations, periodic table and periodic properties. Names and formulas of inorganic compounds. The mole. Oxidation numbers.
Credit 2. Nature and types of chemical bonds: covalent, ionic and metallic bonds. Dipole moment and molecular geometry. Valence bond theory and hybrid orbital theory. Classification of solids by type of attraction of units. Acid-base concepts.
Credit 3. Connection between the structure of matter and chemical and physical properties. Polymeric materials. Manomaterials and nanotechnology</td></tr>
<tr><td colspan="2">Suggested text-books</td></tr>
<tr><td colspan="2">Kelter P., Mosher M., Scott A.; CHIMICA - La Scienza della vita- EdiSES, Napoli
Additional material:
Copies of the material used during the lesson in electronic format available on the website www.unimol.it</td></tr>
<tr><td colspan="2">Organization of teaching: (lectures, tutorials, laboratory, etc.)</td></tr>
<tr><td colspan="2">The course includes lectures supported by slide projection.</td></tr>
<tr><td colspan="2">Language</td></tr>
<tr><td colspan="2">Italian</td></tr>
<tr><td colspan="2">Methods and assessment criteria</td></tr>
<tr><td colspan="2">Oral examination: The student must demonstrate knowledge of the topics covered.</td></tr>
</table>

SEMANTIC WEB

Lecturer	Mario Petrone
ECTS	3

Learning outcome and their consistency with the objectives of the course of study

The course aims to provide students with theoretical knowledge and methodological and applicative skills in the Semantic Web field, with specific reference to the methods of knowledge representation (ontologies) and semantic markup languages (XML, RDF (S), OWL, etc.). The student will acquire the necessary skills to develop a Semantic Web application.

Content of the Program/Course:

Credit 1. Semantic Web Fundamentals. Semantic Web fundamental technologies. Relational database model for web (RDF). Metadata, ontologies and rules. Semantic Web application developing. Publishing metadatas. Data mining using structured resources. Data mining using non-structured resources. Methods of obtaining knowledge. Tools of developing knowledge.

Credit 2. Making the Semantic Web. Shared Ontologies. Web shared data as Linked Data. Semantic Web application paradigms. Semantic Web existing applications.

Credit 3. Building a Semantic Web application. User needs analysis. Mesh-up as data integration problem. Requisite analysis. Modelling and designing. Application development and testing

Suggested text-books

E. Della Valle, I. Celino, D. Cerizza, Semantic Web: dai fondamenti alla realizzazione di un'applicazione, PEARSON – Addison Wesley, 2009.

G. Antoniou, F. van Harmelen, Semantic Web Primer, The MIT Press, 2008.

Teacher supplied material

Online material

Organization of teaching: (lectures, tutorials, laboratory, etc.)

Metodi didattici (lezioni frontali, esercitazioni…)

Lectures, tutorials, project developing.

Language

Italian

Methods and assessment criteria

The exam consists in the development of a project and an oral test. The final mark is obtained by summing together project score and oral test score.

Assessment of the knowledge, skills and competences according to Dublin Descriptors:

Knowledge and understanding

Acquiring advanced knowledge to comprehend specific literature on the subject. Technical language usage's skills on the subject.

Applying knowledge and understanding

Ability to independently recognize and organize fundamental topics concerning computing methods of natural language oriented to automated data mining for semantic web usage. The student will have to prove knowledge of acquired skills, specifically related to problems that will arise during the course.

Making judgments

Ability to understand relevance of subject matter and to link theoretical and practical aspects.

Communication skills

Ability to explain semantic web fundamentals to non-experts.

Learning abilities

Development of self-learning skills though consultation of advanced literature on the subject. Using acquired knowledge to enable continuation of studies at a master's degree level.