UNIVERSITÀ
DEL SALENTO

CORSO DI LAUREA LM55 –

Computer Engineering
SCHEDE INSEGNAMENTI DIDATTICA EROGATA
a.a. 2020/2021

UNIVERSITÀ DEL SALENTO

SCHEDA INSEGNAMENTO

COMPUTER VISION

Corso di studio di riferimento	LM55 - CdL Magistrale in Computer Engineering
Dipartimento di riferimento	Dipartimento di Ingegneria dell'Innovazione
Settore Scientifico Disciplinare	ING-INF/03
Docente	In attesa di assegnazione
Crediti Formativi Universitari	9
Ore di attività frontale	81
Ore di studio individuale	144
Anno di corso	I anno
Semestre	I
Lingua di erogazione	Inglese
Percorso	PERCORSO COMUNE

Prerequisiti	No prior experience with computer vision is assumed, although previous knowledge of visual computing or signal processing will be helpful. The following skills are necessary for this class: - Math: Linear algebra, vector calculus, and probability. Linear algebra is the most important. - Data structures: Students will write code that represents images as feature and geometric constructions. - Programming: A good working knowledge. All lecture code and project starter code will be Python, and Pytorch for Deep Learning, but student familiar with other frameworks such as tensorflow is ok.
Contenuti	Computer Vision today is everywhere in our society and images have become pervasive, with applications in several sectors; just to mention some in: apps, drones, healthcare and precision medicine, precision agricolture, searching, understanding, control in robotics and self-driving cars. The course introduces the basics of image formation, reconstruction and inferring motion models, as well as camera calibration theory and practice. Recent developments in neural networks (Deep Learning) have considerably boosted the performance of the visual recognition systems in tasks such as: classification, localisation, detection, segmentation etc. Students will learn the building blocks of a general convolutional neural network, the way how it is trained and optimized, how to prepare a dataset and how to measure the final performance.
Obiettivi formativi	Upon completion of this course, students will: - Be familiar with both the theoretical and practical aspects of computing with images; - Have described the foundation of image formation, measurement, and analysis;

	- Have implemented PERCORSO COMUNE methods for robust image matching and alignment; - Understand the geometric relationships between 2D images and the 3D world; - Have gained exposure to object and scene recognition and categorization from images; - Grasp the principles of state-of-the-art deep neural networks; and - Developed the practical skills necessary to build computer vision applications.
Metodi didattici	Teaching is based on theoretical and practical lectures. The student will write in python algorithms taught in class
Modalità d'esame	Oral session. The student will explain the developed project and shall answer two or more questions regarding theoretical aspects of the studied topics
Programma	Introduction to Computer Vision Image Formation 2D and 3D geometric primitives - Projections image enhancement LAB Introduction to Python and Operations with images Color perception, color spaces and processing Image Filtering image pyramids and blending Local feature detector LAB SIFT with MatLab Find image rotation scale SURF Object Detection Image Alignment I- warping, homography estimation direct linear transform Image Alignment II- robust motion estimation with Ransac - perspective n point problem. Registration examples: face recognition, medical imaging Camera Calibration - distortion models and compensations - linear methods for camera parameters. Calibration with a checkerboard LAB Mosaicking with SURF Face Detection and Tracking (Nose Skin) Face Detection and Tracking withKanade - Lucas - Tomasi feature tracker Motion Analysis and background modelling, application to intelligent videosurveillance Multiview geometry - Epipolar geometry, position error estimation, stereo rig, Essential matrix estimation, rectification, Reconstruction, correspondense problem, weak calibration and ransac estimation of fundamental matrix LAB - Camera calibration Image Classification - Key nearest neighbor, linear classifiers Image Classification - loss functions, optimization with stochastic gradient descent LAB - Stereo calibration and reconstruction Image Classification - backpropagation and neural networks, computational graphs and gradient estimation Image Classification - Convolutional Neural Network architecture Image Classification - CNN activation functions, data preprocessing, weight normalization, batch normalization, monitoring the learning process, hyperparameter optimization, Regularization (Dropout, drop connect,

	fractional pooling, cotout, mixup) Image Classification - CNN activation functions, data preprocessing, weight normalization, batch normalization, monitoring the learning process, hyperparameter optimization, Regularization (Dropout, drop connect, fractional pooling, cotout, mixup) Image Classification - Object detection and Image segmentation with Convolutional neural networks. Introduction to auto-encoders LAB - Introduction to Pytorch framework LAB - Deep learning applications to object detection (Yolo and Faster R-CNN) LAB - Deep Learning application to segmentation with mark R-CNN
Testi di riferimento	There is no requirement to buy a book. The goal of the course is to be self contained, but sections from the following textbooks will be suggested for more formalization and information. The primary course text will be Rick Szeliskis draft Computer Vision: Algorithms and Applications; we will use an online copy of the <a href="http://research.microsoft.com/en-us/um/people/szeliski/Book/">June 19th draft. A copy and link will be provided in website. The secondary text is Forsyth and Ponce, Computer Vision: A Modern Approach (new Edition coming out in 2020) <a href="http://www.deeplearningbook.org/front_matter.pdf">Deep Learning, MIT Press book, Ian Goodfellow and Yoshua Bengio and Aaron Courville
Altre informazioni utili	For the LAB practice, students may use for the deep learning development the Google Colab or Cloud Platform.

UNIVERSITÀ DEL SALENTO

SCHEDA INSEGNAMENTO

DECISION SUPPORT SYSTEMS

Corso di studio di riferimento	LM55 - CdL Magistrale in Computer Engineering
Dipartimento di riferimento	Dipartimento di Ingegneria dell'Innovazione
Settore Scientifico Disciplinare	MAT/09
Docente	Gianpaolo GHIANI
Crediti Formativi Universitari	9
Ore di attività frontale	81
Ore di studio individuale	144
Anno di corso	I anno
Semestre	I
Lingua di erogazione	Inglese
Percorso	PERCORSO COMUNE

Prerequisiti	Calculus. Probability and Statistics. Linear Algebra. Programming skills.
Contenuti	The course provides the theoretical foundations, the practical skills and the development tools to design and deploy intelligent systems that support or automate complex decisions. Applications include motion planning in robotics, designing non-player characters in video games, machine scheduling in the manufacturing sector, portfolio optimization in the financial industry, timetabling and crew rostering in transportation, Methodologies and algorithms taken from Operations Research, Statistics and Artificial Intelligence are analyzed and compared.
Obiettivi formativi	Knowledge and understanding . The course describes methods and models to design decision support/automation systems. - Students will acquire the basic cognitive tools to think analytically, creatively, critically and in an inquiring way, and have the abstraction and problem-solving skills needed to cope with complex systems. - They will have solid knowledge of decision support/automation systems. - They will be able to design and develop complex systems to improve decision-making processes. Applying knowledge and understanding. After the course the student should be able to: - describe and use the main decision support/automation techniques; - understand the differences among several algorithms solving the same problem and recognize which one is better under different

	conditions; - tackle decision support/automation problems by selecting the appropriate methods and justifying his/her choices; - tackle new decision support/automation problems by designing suitable algorithms and evaluating the results; - explain experimental results to people without a computer science background. Making judgements . Students must have the ability to assess a decision support/automation system and must arrive at original and autonomous ideas and judgments.. The course promotes the development of independent judgment in the appropriate choice of techniques/models and the critical ability to interpret the goodness of the results of the chosen models/methods. Communication. It is essential that students are able to communicate with a varied and composite audience, not culturally homogeneous, in a clear, logical and effective way, using the methodological tools acquired and their scientific knowledge and, in particular, the specialty vocabulary. Students should be able to organize effective dissemination and study material through the most PERCORSO COMUNE presentation tools, including computer-based ones, to communicate the results of data analysis processes, for example by using visualization and reporting tools aimed at different types of audiences. Learning skills. Students must acquire the critical ability to relate, with originality and autonomy, to the typical problems of data mining and, in general, cultural issues related to other similar areas. They should be able to develop and apply independently the knowledge and methods learnt with a view to possible continuation of studies at higher (doctoral) level or in the broader perspective of cultural and professional self-improvement of lifelong learning. Therefore, students should be able to switch to exhibition forms other than the source texts in order to memorize, summarize for themselves and for others, and disseminate scientific knowledge.
Metodi didattici	The course consists of lectures, classroom exercises and home assignments. Lectures aim at providing the methodological foundations. They are given using slides and/or a blackboard. Students are invited to participate by asking questions and presenting examples. The exercises and home assignments are about the solution of practical problems with software tools.
Modalità d'esame	The exam consists of two parts: - a written test made up of 10 questions [10 marks]; - an oral exam in which students must: - discuss a presentation of their own on an advanced course topic [10 marks]; - show their ability to use the software tools presented in the course (Python libraries for machine learning, STRIPS, AMPL, ...) [10 marks].

UNIVERSITÀ DEL SALENTO

Programma	PART I DECISION-MAKING PROCESSES (4 hours) 1.1 Introduction. Data, information, knowledge, decisions. Taxonomy of decisions. Decision support methodologies. (2hours) 1.2 Intelligent agents. (2 hours) PART II SIMULATION (10 hours) 2.1 Evaluation: experimentation, simulation and analitical methods (1 hour) 2.2 Pseudo-random number generation. (3 hours) 2.3 Monte Carlo simulation. Discrete-event simulation. Variance reduction techniques. (6 hours) PART III - KNOWLEDGE, REASONING AND PLANNING (28 hours) 3.1 Search. Uninformed and informed search. A* algorithm. (3 hours) 3.2 Basics of optimization. Optimization model review. Convex Optimization. Linear Optimization. (10 hours) 3.3 Local search. Simulated Annealing. Genetic Algorithms. (4 hours) 3.4 Adversarial search. Basics of Game Theory. (4 hours) 3.5 Propositional and first-order logic (recap) (4 hours) 3.5 Planning. The STRIPS language (3 hours) PART IV - PLANNING IN UNCERTAIN ENVIRONMENTS (13 hours) 3.1 Decision making under uncertainty (1 hour) 3.2 Decision making under risk (2 hours) 3.3 Sequential decision processes (4 hours) 3.4 Dynamic Programming (6 hours) PART IV LEARNING (16 hours) 4.1 Introduction (1 hour) 4.2 Supervised learning: linear and polynomial regression, naive Bayes classifier, classification and regression trees, linear classification with hard threshold, linear classification with logistic regression, basics of neural networks (8 hours); non parametric classification; model selection (8 hours) 4.3 Unsupervised learning: clustering: k-means algorithm, determination of the number of clusters; rule mining: the a-priori algorithm (4 hours) 4.4 Reinforcement learning (3 hours)
Testi di riferimento	Handouts (available on FormazioneOnLine at <a href="https://formazioneonline.unisalento.it/course/view.php?id=487">https://formazioneonline.unisalento.it/course/view.php?id=487). For consultation: - Russell, Stuart J., and Peter Norvig. Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited, 2016.
Altre informazioni utili	Office Hours By appointment. As a rule, on Thursdays at 11:00. Please contact the instructor by email or at the end of the lectures.

UNIVERSITÀ DEL SALENTO

SCHEDA INSEGNAMENTO

SYSTEM AND NETWORK PROGRAMMING

Corso di studio di riferimento	LM55 - CdL Magistrale in Computer Engineering
Dipartimento di riferimento	Dipartimento di Ingegneria dell'Innovazione
Settore Scientifico Disciplinare	ING-INF/05
Docente	Francesco Tommasi
Crediti Formativi Universitari	12
Ore di attività frontale	108
Ore di studio individuale	192
Anno di corso	I anno
Semestre	I
Lingua di erogazione	Inglese
Percorso	PERCORSO COMUNE

Prerequisiti	All the concepts presented in the "Sistemi Operativi" course in the first level degree "Ingegneria dell'Informazione". Namely, a good knowledge of: UNIX® basic concepts, the UNIX® bash shell, bash scripting, main UNIX® commands
Contenuti	The course introduces the student to the main UNIX system calls and APIs
Obiettivi formativi	The course aims at starting the students off on programming system applications (e.g. a server) on a UNIX® System. Learning Outcomes; after the course the student should * Know the most important functionalities and facilities offered by a UNIX® system, the System Calls (and, more generally, the APIs) offered to access them. * Be able to write efficient CLI (Command Line Interface) system and network applications in the C language. * Know how to write interoperable applications by complying with the UNIX® standards (SUSv3, SUSv4). * Know which are the main differences between the MacOS and the Linux varieties and how to cope with them.
Metodi didattici	The course is strongly oriented towards an hands-on methodology. Students must follow lectures in front of a computer which must be used to reproduce and test what is explained by the teacher
Modalità d'esame	Writing a C program aimed at solving a given problem within a given time. Students are free to consult (paper and digital) texts and to use Internet search engines.
Programma	UNIX System Overview UNIX Standardization and Implementations File I/O Files and Directories System Data Files and Information

	Process Environment Process Control Process Relationships Signals Threads Thread Control Daemon Processes Advanced I/O Interprocess Communication Network IPC: Sockets Terminal I/O Cybersecurity (Disassembling an executable Following the execution of a process at machine code level Buffer overflows Shellcode)
Testi di riferimento	Stevens, Rago, Advanced Programming in the UNIX Environment, 3rd Edition, Addison-Wesley, 2013 ISBN 978-0321637734 Stevens, Fenner, Rudoff, Unix Network Programming, Volume 1: The Sockets Networking API (3rd Edition), Addison-Wesley, 2003 ISBN 978-0131411555 Kerrisk, The Linux Programming Interface, NO STARCH PRESS, 2010 ISBN 978-1593272203 Handouts delivered by the teacher through http://moodliis.unisalento.it/
Altre informazioni utili	

UNIVERSITÀ DEL SALENTO

SCHEDA INSEGNAMENTO

Advanced Control Techniques

Corso di studio di riferimento	LM55 - CdL Magistrale in Computer Engineering
Dipartimento di riferimento	Dipartimento di Ingegneria dell'Innovazione
Settore Scientifico Disciplinare	ING-INF/04
Docente	Gianfranco Parlangeli
Crediti Formativi Universitari	9
Ore di attività frontale	81
Ore di studio individuale	144
Anno di corso	I anno
Semestre	II
Lingua di erogazione	Inglese
Percorso	PERCORSO COMUNE

Prerequisiti	conoscenze di Algebra Lineare, teoria dei Sistemi
Contenuti	Questo corso offre un'ampia panoramica di argomenti fondamentali ed emergenti nell'area del controllo e della teoria dei sistemi. Le applicazioni sono illustrate nei settori della robotica, dei sistemi multi-agente e dei sistemi cyber-fisici. Ha lo scopo di fornire principi e strumenti per affermare e risolvere problemi di controllo ottimali alla fine alla ricerca di architetture di controllo distribuite in diversi sistemi tecnologici, e la soluzione è ricercata sia analiticamente attraverso il calcolo diretto sia numericamente con l'aiuto di un software adeguato (viene utilizzato Mathworks Matlab nel corso).
Obiettivi formativi	Risultati di apprendimento; dopo il corso lo studente dovrebbe essere in grado di: (Conoscenze e comprensione) Descrivi e spiega le principali peculiarità (sia vantaggi che svantaggi) della teoria del controllo classica e moderna considerata nel corso. (Capacità di applicare conoscenze e comprensione) + (Abilità comunicativa) + (Autonomia di giudizio) Essere consapevoli, descrivere e spiegare i problemi pratici del controllo di sistemi complessi e come superare questi inconvenienti utilizzando approcci moderni. (Capacità di applicare conoscenze e comprensione) + (Capacità di apprendimento) Per un dato problema pratico a portata di mano, lo studente dovrebbe essere in grado di affermare un problema di controllo in un ambiente matematico naturale, alla fine alla ricerca di architetture distribuite, basate sulle ipotesi del problema. (Capacità di applicare conoscenze e comprensione) + (Abilità comunicativa) + (Autonomia di giudizio) Partendo da una formulazione teorica di un problema, lo studente dovrebbe essere in grado di costruire un framework di simulazione per trovare una soluzione computerizzata del problema matematico dichiarato con

	l'uso di un software adatto. (Abilità comunicativa) + (Capacità di apprendimento) Gli studenti volontari possono sviluppare un progetto su un'applicazione di interesse in cui applicare le metodologie sviluppate lungo il corso.
Metodi didattici	Lezioni frontali svolte in aula dal docente tramite l'ausilio di gesso e lavagna. Nel corso delle lezioni saranno occasionalmente illustrati e discussi software commerciali.
Modalità d'esame	L'esame è una prova scritta e una discussione orale e ha lo scopo di determinare in che misura lo studente ha: 1) la capacità di identificare e utilizzare i dati per formulare risposte a problemi ben definiti, 2) capacità di problem solving per cercare un soluzione analitica. Inoltre, gli studenti disponibili possono avere un seminario o un progetto su un'applicazione di interesse in cui vengono applicate le metodologie del corso.
Programma	Introduction. Mathematical background and connections with other courses. Background on Systems theory and linear algebra. Jordan form of a matrix. Linear systems, unforced response and forced response. Exponential and raise to a power of a square matrix. Stability of a linear system and Lyapunov Equation. Linear systems controllability and observability. Eigenvalues placement through state feedback: Rosenbrock theorem. Kalman decomposition of a linear system. Introduction to optimal control. Extremum seeking techniques. Functionals. Normed vector spaces. Weak and strong extremum. Differentiable functionals and first variation. Calculus of variations, Euler equation: derivation, comments, examples. The Bellman's optimal principle: statement, examples. Cost to go. Costate variables. The optimal control problem solved using the Bellman approach for continuous time systems: HJB equation. Derivation. Examples. The optimal control problem in the presence of saturation: the Pontryagin's maximum principle. The linear quadratic optimal control problem. Statement and solution using the variational approach. Discussion on the issues of extending the horizon to infinity. Main theorems. Riccati and Lyapunov equations. Nonsingular solutions of the Riccati Equation. Multi agent systems: an introduction. Examples, main definitions. Centralized architectures vs decentralized ones. Supervisory control, distributed control. Some notions of Graph theory. Dynamical systems over graphs. The importance of consensus in various emerging fields. Consensus protocols. Consensus networks. Analysis of consensus within a multi-agent dynamical system. Consensus problems for directed graphs. Leader-follower multi-agent systems. Symmetries and equitable partitions. Directed weighted graphs: a model for consensus networks and cyber-physical systems. Analysis, properties. Differences between directed weighted graphs and undirected weighted graphs. Examples. Misbehaving nodes and intruders in a collaborative network. System zeros and output-nulling inputs. Rosenbrock's system matrix. Unobservable zeros and transmission zeros.
Testi di riferimento	[1] Antsaklis, P. J., & Michel, A. N. (2006). Linear systems. Springer Science & Business Media. [2] Anderson, Brian DO, and John B. Moore, Optimal control: linear quadratic methods, Courier Corporation, 2007. [3] Bullo, F. Lectures on Network Systems, with contributions by J. Cortes, F.

	Dorfler and S. Martinez, Kindle Direct Publishing, 2018.
Altre informazioni utili	

UNIVERSITÀ DEL SALENTO

SCHEDA INSEGNAMENTO

Internet of Things

Corso di studio di riferimento	LM55 - CdL Magistrale in Computer Engineering
Dipartimento di riferimento	Dipartimento di Ingegneria dell'Innovazione
Settore Scientifico Disciplinare	ING-INF/05
Docente	Luigi Patrono
Crediti Formativi Universitari	9
Ore di attività frontale	81
Ore di studio individuale	144
Anno di corso	I anno
Semestre	II
Lingua di erogazione	Inglese
Percorso	PERCORSO COMUNE

Prerequisiti	Reti di Calcolatori, Principi di Ingegneria del Software, Fondamenti di Informatica
Contenuti	+ Description of the course + Introduction to Internet of Things and Web of Things + Introduction to WoT through use cases and practical approach + Introduction to Typescript and Node.js applications + WoT stack + RFID technology and Traceability + Bluetooth Low Energy and its evolution + Wireless Sensor Networks: IEEE 802.15.4, 6LowPAN, RPL, IPv6 + Embedded Systems: Raspberry Pi, MT3620 and STM32 + Layer 1 of the WoT stack: Access Layer + REST, CoAP, MQTT + Layer 2 of the WoT stack: Find Layer (Semantic Web) + Layer 3 of the WoT stack: Share Layer + Security in IoT and WoT, Blockchain and IOTA + Layer 4 of the WoT stack: Compose Layer (Physical Mashup) + Introduction to Cloud Computing and Edge/Fog Computing + Domotics: KNX standard and practical use cases + Discussion of several use cases regarding smart environments
Obiettivi formativi	The Internet of Things course aims to offer a complete vision on how to design and develop smart objects and smart services based on hardware and software technologies enabling the Internet of Things. Particular attention will be paid to the creation and testing of the so-called smart environments. The Web of Things approach will be adopted which allows a total abstraction from the main physical technologies adopted in modern networks. The extended WoT protocol stack, composed of four layers,

	will be discussed, details on emerging enabling technologies such as RFID, embedded systems, WSN and Bluetooth Low Energy (BLE) will be provided. The REST architectural style and protocols such as CoAP and MQTT will be described. Several practical use cases focused on building smart environments will be discussed.
Metodi didattici	Teaching methodology adopted in the Internet of Things course is based both on theoretical discussion on emerging technologies enabling the IoT and practical discussion of use cases about the design and developing of smart environments. Furthermore, several external seminars focused on specific topics of the IoT will be organized involving important industrial companies.
Modalità d'esame	Discussion of a practical project or a research topic in the IoT field and oral exam on all topics analyzed in the course.
Programma	+ Description of the course + Introduction to Internet of Things and Web of Things + Introduction to WoT through use cases and practical approach + Introduction to Typescript and Node.js applications + WoT stack + RFID technology and Traceability + Bluetooth Low Energy and its evolution + Wireless Sensor Networks: IEEE 802.15.4, 6LowPAN, RPL, IPv6 + Embedded Systems: Raspberry Pi, MT3620 and STM32 + Layer 1 of the WoT stack: Access Layer + REST, CoAP, MQTT + Layer 2 of the WoT stack: Find Layer (Semantic Web) + Layer 3 of the WoT stack: Share Layer + Security in IoT and WoT, Blockchain and IOTA + Layer 4 of the WoT stack: Compose Layer (Physical Mashup) + Introduction to Cloud Computing and Edge/Fog Computing + Domotics: KNX standard and practical use cases + Discussion of several use cases regarding smart environments
Testi di riferimento	+ Building the Web of Things: With Examples in Node.js and Raspberry Pi. Dominique D. Guinard, Vlad M. Trifa + Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security. Perry Lea + Scientific papers + Web links
Altre informazioni utili	All didactic materials (slides, scientific papers, etc..) are available in two repositories of the University of Salento: Google Drive (https://drive.google.com/drive/u/0/folders/0ABxf0yPcEXECUk9PVA) and FormazioneOnline (https://formazioneonline.unisale

UNIVERSITÀ DEL SALENTO

SCHEDA INSEGNAMENTO

Software Engineering

Corso di studio di riferimento	LM55 - CdL Magistrale in Computer Engineering
Dipartimento di riferimento	Dipartimento di Ingegneria dell'Innovazione
Settore Scientifico Disciplinare	ING-INF/05
Docente	Luca Mainetti
Crediti Formativi Universitari	9
Ore di attività frontale	81
Ore di studio individuale	144
Anno di corso	I anno
Semestre	II
Lingua di erogazione	Inglese
Percorso	PERCORSO COMUNE

Prerequisiti	The prerequisites for attending the corse are the knowledge of structured programming languages (Java) and the fundamentals of computer science.
Contenuti	After the course the student should be able to: a. Apply main software engineering principles and control software qualities (both internal and external); b. Design and implement software following industrial standards (UML) and structured software production processes; c. Manage the software engineering i.e. execute tasks as planning, organizing, staffing, controlling, estimating (software cost and size); d. Design the software adopting standard software architectures; e. Select and adopt software design patterns (creational patterns, structural patterns, behavioral patterns); f. Verify the software exploiting standard tools and adopting well-known metrics; g. Develop complex model-view-controller web and mobile software systems, exploiting at the back end the Spring framework, and at the front end the Angular framework, connecting them through REST/JSON web services; h. Manage the fundamentals of modern cloud computing and cloud service deployment; i. Use the main open source tools for the software testing and refactoring, and for the software configuration management.
Obiettivi formativi	The main goal of the course is to deepen students' knowledge on modern design and development techniques for interactive software systems. In particular, methods and tools for automated software testing, agile processes organization and design patterns selection will be analyzed. All concepts will be experimented by students designing, developing and testing a software prototype of a service based web application with a mobile extension (app). The software prototype will be developed on top of modern frameworks (Spring, Angular, Ionic, Amazon AWS).

Metodi didattici	Online or classroom lessons, classroom practice, project work in pair programming.
Modalità d'esame	The exam consists of two tests: a written test, intended to verify the theory of software engineering concepts (10 points out of 30); a software prototype implementation, intended to verify the practice of design patterns, MVC architectures and tests, which will be discussed during an oral examination (20 points of 30). Both written test and software prototype implementation are mandatory. The software prototype should be developed in pairs. The software system must be designed using UML, adopting standard design patterns. The software system must be developed starting from MVC frameworks (Spring, Angular, Ionic, AWS), using a structured programming language, and must be systematically tested collecting metrics. A mobile extension of the software system is required. The software prototype must be developed following an agile process and must be documented. A month before the end of the course, the general requirements of the software prototype will be published by the teacher, a new requirements set for each year. The requirements will be effective till a new set of specifications will appear. The mark of the written exam has the same temporal extension of the project's requirements.
Programma	Software engineering principles: - Software qualities and software engineering principles; - Software production process; - Management of software engineering. Software architectures: - Design and software architectures; - Software architectures specification. Software design pattern: - Introduction to standard architectures and design patterns; - How to select and adopt a design pattern; - Creational patterns, structural patterns, behavioral patterns. Software verification: - Introduction to man software verification methods; - Black-box and white-box methods; - Test in the large, test in the small, correctness proofs; - Software metrics. Introduction to Spring framework: - Introduction to Java EE; - Creating a dynamic web project with Java EE; - Introduction to Spring framework and development environment setup; - Developing a Spring MVC application; - Accessing Data Layer with Spring Data JPA; - Building a RESTful Web Service. Software development and verification tools: - Unit testing with JUnit and Refactoring; - Versioning control and code sharing with GIT. Cloud computing with Amazon Web Services and EC2: - Introduction to Amazon EC2 platform; - Introduction to Amazon API Gateway; - Configuring an EC2 instance and publishing API.

	Mobile apps development with AngularJS: - Angular: Project Setup; - Angular: Component, Template & Data Binding; - Angular: Forms (Input, Validation, Template-Driven); - Angular: Services, Routing, HTTP; - Developing Cross-platform Mobile App with Ionic.
Testi di riferimento	1. Ghezzi, Jazayeri, Mandrioli - Fundamentals of Software Engineering (2nd edition) - Pearson College Div 2002. 2. Fowler - UML Distilled (3rd edition) - Addison Wesley Object Technology 2003. 3. Gamma, Helm, Johnson, Vlissides - Design patterns - Addison Wesley 2002. 4. Larman - Agile and Iterative Development: A Manager's Guide - Addison-Wesley Professional 2003. 5. Beck - Test Driven Development: By Example - Addison-Wesley Professional 2002.
Altre informazioni utili	www.unisalento.it/people/luca.mainetti

UNIVERSITÀ DEL SALENTO

SCHEDA INSEGNAMENTO

Big Data Managment

Corso di studio di riferimento	LM55 - CdL Magistrale in Computer Engineering
Dipartimento di riferimento	Dipartimento di Ingegneria dell'Innovazione
Settore Scientifico Disciplinare	ING-INF/05
Docente	Mario Bochicchio
Crediti Formativi Universitari	9
Ore di attività frontale	81
Ore di studio individuale	144
Anno di corso	II anno
Semestre	I
Lingua di erogazione	Inglese
Percorso	PERCORSO COMUNE

Prerequisiti	Good knowledge of Object Oriented Languages (at least 1), techniques and tools. Elements of computer networks and Web technologies.
Contenuti	The aim is to provide the basics about the main database theories, techniques and tools to design / implement databases and database applications. Topics: •Database, relational databases, NoSQL and NewSQL; •DataBase Management Systems; •Relational Model and Relational Algebra; •SQL: data definition and manipulation; •Basics of Human-Computer Interaction and interface design for DB; •Architectural aspects: Clients, Servers, Peers, Devices, IoT, …
Obiettivi formativi	Acquired skills - Students will be able to design and understand data models, to create and manage databases and to design and implement data-centric applications.
Metodi didattici	Teaching method Frontal lessons and lectures, for theoretical aspects, will be followed by participatory learning sessions and hands-on sessions to reinforce the comprehension and to acquire the abilities relevant to the field of database design.
Modalità d'esame	- Students evaluation - •Written test: on all aspects covered by the program - •Oral Test: - All theoretical aspects covered by the program
Programma	Fundamental of Database Systems, Elmasri-Navathe: 7th edition

	Chapters: - 1: Databases and Database Users - 2: Database System Concepts and Architecture - 3: Data Modeling Using the Entity–Relationship (ER) Model - 4: The Enhanced Entity–Relationship (EER) Model - 5: The Relational Data Model and Relational Database Constraints - 6: Basic SQL - 7: More SQL: Complex Queries, Triggers, Views, and Schema Modification - 8: The Relational Algebra and Relational Calculus 8.1: Unary Relational Operations: SELECT and PROJECT 8.2: Relational Algebra Operations from Set Theory 8.3: Binary Relational Operations: JOIN and DIVISION 8.4: Additional Relational Operations 8.5: Examples of Queries in Relational Algebra - 9: Relational Database Design by ER- and EER-to-Relational Mapping - 10: Introduction to SQL Programming Techniques - 11: Web Database Programming Using PHP - 12: Object and Object-Relational Databases - 14: Basics of Functional Dependencies and Normalization for Relational Databases 14.1: Informal Design Guidelines for Relation Schemas 14.2: Functional Dependencies 14.3: Normal Forms Based on Primary Keys 14.4: General Definitions of Second and Third Normal Forms 14.5: Boyce-Codd Normal Form - 16: Disk Storage, Basic File Structures, Hashing, and Modern Storage Architectures - 17: Indexing Structures for Files and Physical Database Design - 20: Introduction to Transaction Processing Concepts and Theory - 21: Concurrency Control Techniques - Teaching material: more concepts on requirement elicitation and database application design and implementation, multidimensional analisys, datawharehouse, big data, big data management, database security, database administration, NoSQL, NewSQL, distributed databases.
Testi di riferimento	R. Elmasri, S. Navathe, Fundamental of Database Systems, 7a edizione, Pearson ed.
Altre informazioni utili	Office Hours By appointment; contact the instructor by email or at the end of class meetings.

UNIVERSITÀ DEL SALENTO

SCHEDA INSEGNAMENTO

PARALLEL ALGORITHMS

Corso di studio di riferimento	LM55 - CdL Magistrale in Computer Engineering
Dipartimento di riferimento	Dipartimento di Ingegneria dell'Innovazione
Settore Scientifico Disciplinare	ING-INF/05
Docente	Massimo CAFARO
Crediti Formativi Universitari	9
Ore di attività frontale	81
Ore di studio individuale	144
Anno di corso	II anno
Semestre	I
Lingua di erogazione	Inglese
Percorso	PERCORSO COMUNE

Prerequisiti	Calculus I and II, Probability Theory. Programming skills and working knowledge of the C programming language.
Contenuti	The course provides a modern introduction to design, analysis and implementation of sequential and parallel algorithms. In particular, the course is based on a pragmatic approach to parallel programming of message-passing algorithms through the C language and the MPI library.
Obiettivi formativi	Knowledge and understanding. Students must have a solid background with a broad spectrum of basic knowledge of sequential and parallel algorithms: the students must have the basic cognitive tools to think analytically, creatively, critically and in an inquiring way, and have the abstraction and problem-solving skills needed to cope with complex systems; they must have a solid knowledge of the design and implementation of sequential and parallel efficient algorithms; they must have the tools for analysing the resources used by algorithms; they must have a catalogue of the most well-known and efficient sequential and parallel algorithms for basic computational problems. Applying knowledge and understanding. After the course the student should be able to: Describe and use the main design techniques for sequential algorithms; Design, prove the correctness and analyze the computational complexity of sequential algorithms; Understand the differences among several algorithms solving the same problem and recognize which one is better under different

	conditions; Describe and use basic sequential algorithms; Describe and use basic data structures; know about the existence of advanced data structures; Understand the difference between sequential and parallel algorithms; Design, implement and analyze message-passing based parallel algorithms in C using the MPI library; Describe and use basic parallel algorithms. Making judgements. Students are guided to learn critically everything that is explained to them in class, to compare different approaches to solving algorithmic problems, and to identify and propose, in an autonomous way, the most efficient solution they find. Communication. It is essential that students are able to communicate with a varied and composite audience, not culturally homogeneous, in a clear, logical and effective way, using the methodological tools acquired and their scientific knowledge and, in particular, the specialty vocabulary. The course promotes the development of the following skills of the student: ability to expose in precise and formal terms an abstract model of concrete problems, identifying the salient features of them and discarding the nonessential ones; ability to describe and analyze an efficient solution to the problem in question. Learning skills. Students must acquire the critical ability to relate, with originality and autonomy, to the typical problems of data mining and, in general, cultural issues related to other similar areas. They should be able to develop and apply independently the knowledge and methods learnt with a view to possible continuation of studies at higher (doctoral) level or in the broader perspective of cultural and professional self-improvement of lifelong learning. Therefore, students should be able to switch to exhibition forms other than the source texts in order to memorize, summarize for themselves and for others, and disseminate scientific knowledge.
Metodi didattici	The course aims to enable students to abstract formal algorithmic models and problems from concrete computational problems, and to design efficient algorithmic solutions for them. This will be done using the following teaching method. Every computational problem will be introduced, motivating it with concrete examples. The presentation of each topic will be divided into four parts: 1. Description of the actual computational problem. 2. Modelling the real problem by means of an abstract problem. 3. Resolution of the abstract problem through an algorithm obtained through the application of the general techniques of design of algorithms introduced in the course. 4. Analysis of the resources used by the algorithm. The course consists of frontal lessons using slides made available to students via the Moodle platform, and classroom exercises. There will be theoretical lessons aimed at learning the basic techniques for the project and analysis of algorithms, and a part of

	lessons of an exercise type in which you will illustrate, with plenty of examples, how the theoretical knowledge acquired can be used in order to solve algorithmic problems of practical interest and implement parallel algorithms in C language through the MPI library.
Modalità d'esame	Oral exam. Optionally, a student may be assigned a small project. During the exam the student is asked to illustrate theoretical topics in order to verify his/her knowledge and understanding of the selected topics. The student may also be asked to design a very simple algorithm in order to assess his/her ability to identify and use the relevant design techniques; alternatively, the student may be asked to analyze the complexity of a small code fragment.
Programma	Sequential A lgorithms Introduction. Order of growth. Analysis of algorithms. Decrease and conquer. Divide and conquer. Recurrences. Randomized algorithms. Transform and conquer. Dynamic programming. Greedy algorithms. Complexity and computability. NP-Completeness. Parallel Algorithms Introduction. The transition from sequential to parallel computing. Parallel complexity. Parallel architectures. Parallel algorithm design. Message-Passing programming. Sieve of Erathostenes. Floyd all-pairs shortest path algorithm. Performance analysis. Matrix-vector multiplication. Document classification. Matrix multiplication.
Testi di riferimento	Introduction to Algorithms. Third edition. Cormen, Leiserson, Rivest, Stein. The MIT Press Parallel Programming in C with MPI and OpenMP International Edition (2004) Michael J. Quinn McGraw-Hill
Altre informazioni utili	Office Hours By appointment; contact the instructor by email or at the end of class meetings.

UNIVERSITÀ DEL SALENTO

SCHEDA INSEGNAMENTO

Robotics

Corso di studio di riferimento	LM55 - CdL Magistrale in Computer Engineering
Dipartimento di riferimento	Dipartimento di Ingegneria dell'Innovazione
Settore Scientifico Disciplinare	ING-INF/04
Docente	Gianfranco Parlangeli
Crediti Formativi Universitari	9
Ore di attività frontale	81
Ore di studio individuale	144
Anno di corso	II anno
Semestre	I
Lingua di erogazione	Inglese
Percorso	PERCORSO COMUNE

Prerequisiti	Sufficiency in calculus, mechanics, control theory and linear algebra
Contenuti	This course offers a broad overview of fundamental topics in the area of robotics and mobile robotics and multi-robotic systems. It is aimed at providing principles and tools to state and solve the design problems for industrial robots and mobile devices, and the solution is numerically sought with the aid of a suitable software (Mathworks Matlab is used in the course).
Obiettivi formativi	Ability to apply knowledge and understanding) Describe and explain the main peculiarities (both advantages and disadvantages) of each facet of the design of a robotic, mobile robotic and multi-robotic systems. (Ability to apply knowledge and understanding) + (Communication skills) + (Autonomy of judgment) Be aware, describe and explain the practical problems of controlling complex systems and how to overcome these drawbacks using modern approaches. (Ability to apply knowledge and understanding) + (Learning ability) + (Autonomy of judgment) Starting from a practical problem, the student must be able to formalize an adequate theoretical formulation, and also should be able to build a framework of simulation to find a computer solution of the mathematical problem with the use of a suitable software. (Communication skills) + (Learning skills) Students can develop a project on an application of interest in which to apply the methodologies developed along the course.
Metodi didattici	Lezioni frontali svolte in aula dal docente tramite l'ausilio di gesso e lavagna. Nel corso delle lezioni saranno occasionalmente illustrati e discussi software commerciali.
Modalità d'esame	The exam is an oral discussion (including possibly one written exercise) and it is aimed to determine to what extent the student has: 1) the ability to

	identify and use data to formulate responses to well-defined problems, 2) problem solving abilities to seek a solution through an algorithm.
Programma	Introduction to Robotics. Robot Mechanical Structures. Robot Manipulators, Mobile Robots, Industrial robotics. Advanced Robotics, Field Robots, Service Robots. Robot Modelling, Planning and Control. Mathematical background and connections with other courses. Kinematics. Euler Angles. Denavit–Hartenberg Convention.Kinematics of Typical Manipulator Structures. The Inverse Kinematics Problem. Differential Kinematics and Statics. Geometric Jacobian. Kinematic Singularities. Analysis of Redundancy.Statics. Kineto-Statics Duality. Trajectory Planning. Joint Space Trajectories. Dynamics. Lagrange Formulation. Newton–Euler Formulation. Dynamic Manipulability Ellipsoid. Motion Control. Force Control. Mobile Robots. Nonholonomic Constraints. Kinematic Model, Dynamic Model. Planning, Motion Control.
Testi di riferimento	Title: Robotics: Modelling, Planning and Control Authors: Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G. Publisher: Springer-Verlag London Copyright Year: 2009
Altre informazioni utili	

UNIVERSITÀ DEL SALENTO

SCHEDA INSEGNAMENTO

Estimation and Data Analysis with Applications

Corso di studio di riferimento	LM55 - CdL Magistrale in Computer Engineering
Dipartimento di riferimento	Dipartimento di Ingegneria dell'Innovazione
Settore Scientifico Disciplinare	ING-INF/04
Docente	Daniela De Palma
Crediti Formativi Universitari	9
Ore di attività frontale	81
Ore di studio individuale	144
Anno di corso	II anno
Semestre	II
Lingua di erogazione	Inglese
Percorso	PERCORSO COMUNE

Prerequisiti	Sufficiency in calculus, probability theory, linear algebra.
Contenuti	This course offers a broad overview of fundamental and emerging topics in the area of estimation theory and data analysis; furthermore, a set of applications are illustrated in the fields of robotics, multi-agent and cyber-physical systems, and social systems. It is aimed at providing principles and tools to state and solve estimation problems in technological systems, and the solution is numerically sought with the aid of a suitable software (Mathworks Matlab).
Obiettivi formativi	After the course the student should be able to: (Knowledge and understanding) Describe and explain the main peculiarities (both advantages and disadvantages) of each mathematical framework for the estimation problems considered in the course. (Applying knowledge and understanding) + (Communication) + (Making judgements) Be aware of, describe and explain practical problems of bad data gathering and robustness issues in the framework of estimation theory. (Applying knowledge and understanding) + (Learning skills) For a given practical problem at hand, be able to state an estimation problem in a natural mathematical setting, either stochastic or deterministic, based on the problem assumptions. (Applying knowledge and understanding) + (Communication) + (Making judgements) Build a simulation framework to find a computer-aided solution of the stated mathematical problem with the use of a suitable software.
Metodi didattici	Frontal lessons and lectures.
Modalità	Oral exam and development of a project.

d'esame	The objective of the exam is to determine to what extent the student has: 1) the ability to identify and use data to formulate responses to well-defined problems, 2) problem solving abilities to seek a solution through an algorithm.
Programma	Introduction. Mathematical background and connections with other courses. Set membership estimation: introduction, fundamental results and theorems. Set membership estimation: some applications. Stochastic Estimators: definitions, properties, performances and fundamental limitations. Foundations of maximum likelihood estimation. The Bayesian approach to the estimation problem. Kalman filter: discrete-time stochastic state models, (two-steps) structure, computation of the optimal gain, the alternative geometric approach. Steady–state behavior. Extended Kalman Filter. Applications of Kalman Filter. Smoothing Algorithms. Robust estimation: introduction, fundamental definitions, estimator classes and performances. Applications of the previous issues and results to various fields.
Testi di riferimento	Yaakov Bar-Shalom, X. Rong Li, Thiagalingam Kirubarajan "Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software", 2001 John Wiley & Sons, Inc. D. Simon, "Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches", John Wiley & Sons, 2006 Anderson, Brian D.O., and John B. Moore. "Optimal Filtering", 1979. L. Ljung, "System Identification: Theory for the User", Prentice Hall PTR, Upper Saddle River, NJ, 1999. Rousseeuw PJ, Leroy AM. "Robust Regression and Outlier Detection". John Wiley & Sons: Hoboken, NJ, USA, 2003. Huber PJ, Ronchetti EM. "Robust Statistics" - Second Edition. Wiley: New York, 2009. Milanese, M., Norton, J., Piet-Lahanier, H., Walter, É. (Eds.). (2013). "Bounding approaches to system identification" Springer Science & Business Media. Zaki, Mohammed J., and Wagner Meira Jr. "Data mining and analysis: fundamental concepts and algorithms", Cambridge University Press, 2014.
Altre informazioni utili	

UNIVERSITÀ DEL SALENTO

SCHEDA INSEGNAMENTO

HIGH PERFORMANCE COMPUTING

Corso di studio di riferimento	LM55 - CdL Magistrale in Computer Engineering
Dipartimento di riferimento	Dipartimento di Ingegneria dell'Innovazione
Settore Scientifico Disciplinare	ING-INF/05
Docente	Giovanni ALOISIO
Crediti Formativi Universitari	9
Ore di attività frontale	81
Ore di studio individuale	144
Anno di corso	II anno
Semestre	II
Lingua di erogazione	Inglese
Percorso	PERCORSO COMUNE

Prerequisiti	Students are expected to have the following background: Knowledge of basic computer science principles and skills, at a level sufficient to write a reasonably non-trivial computer program. Familiarity with the basic probability theory. Familiarity with the basic linear algebra. Good knowledge of the contents of first level courses on Informatics (Fondamenti di Informatica and Calcolatori Elettronici) and of the courses on Parallel Algorithms - 2nd Year (1st semester) of the 2nd Level Degree in Computer Engineering.
Contenuti	The course provides a broad introduction to Machine Learning and Neural Networks Topics include: Linear regression, Linear regression with multiple variables, Regularization, Logistic regression, Neural Networks, Advice for Machine Learning, Machine Learning system design, Support Vector Machines. Hands-off are also organized to provide students the capacity to develop specific use cases, using the Python language and Jupiter Notebooks.
Obiettivi formativi	The course focuses on the basic of Machine Learning and Neural Networks. The theory behind Machine Learning and Neural Networks is presented and applications are developed using the Python language and Jupiter Notebooks.
Metodi didattici	Compulsory attendance is requested, since attendance at lectures and laboratory is mandatory because the course is based on the learning by doing approach on the advanced computing resources provided. The course will be held in the HPC Lab of the Department of Innovation Engineering. Lectures will be also given exploiting communication and collaboration platforms, like Microsoft Teams.
Modalità d'esame	The oral exam is aimed at verifying to what extent the student has gained knowledge and understanding of the selected topics of the course and is able to communicate about his understanding. Students, divided into small groups (max two students), will also get hands-on experience, developing small projects on specific topics of the course. The max final vote is expressed as 30/30 with the possibility to get the laude.

UNIVERSITÀ DEL SALENTO

Programma	Introduction to Machine Learning (2 hours). Linear Regression: Model representation, Cost Function, Cost Function Intuition, Gradient Descent, Gradient Descent Intuition, Gradient Descent for Linear Regression (6 hours). Linear Regression with multiple variables: Multiple features, Gradient Descent for Multiple variables, Gradient Descent in practice I: feature scaling, Gradient Descent in practice II: Learning rate, Features and polynomial regression, Normal Equation, Normal Equation and non-invertibility (6 hours). Regularization: The problem of Overfitting, Cost Function, Regularized Linear Regression (6 hours). Logistic Regression: Classification, Hypothesis Representation, Decision Boundary, Cost Function, Simplified cost function and gradient descent, Advanced Optimization, Multi-class classification: One-vs-all, Overfitting, Cost Function, Regularized Logistic Regression (6 hours). Neural Networks-Representation: Non-linear hypothesis, Neurons and brains, Model Representation, Examples and intuitions, Multi-class classification (6 hours). Neural Networks-Learning: Cost Function and Backpropagation algorithm, Backpropagation intuition. Implementation note: Unrolling parameters, Gradient checking, Random initialization, Putting it together (6 hours). Advice for applying Machine Learning: Deciding what try next, Evaluating a hypothesis, Model selection and training/validation/test sets, Diagnosing bias vs variance, Regularization bias/variance, Learning curves (6 hours). Machine Learning system design: Prioritizing what to work on: Spam classification example, Error analysis, Error metrics for skewed classes, Trading off precision and recall, Data for machine learning (6 hours). Support Vector Machines: Optimization objective, Large Margin Intuition, The mathematics behind large margin classification, Kernels, Using an SVM (6 hours). Hands-on Machine Learning: Use cases developed using Python and Jupiter Notebook (25 hours).
Testi di riferimento	- Andrea Ng , Machine Learning Online book (<a href="http://cnx.org/content/col11500/1.4/">http://cnx.org/content/col11500/1.4/) and in particular the Lectures notes (reported at pag. 223-320 of the online book). - Teachers Slides presented during the course. - Jupiter Notebooks provided during the course.
Altre informazioni utili	Compulsory attendance is requested, since attendance at lectures and laboratory is mandatory because the course is based on the learning by doing approach on the advanced computing resources provided. The course will be held in the HPC Lab of the Enginee

UNIVERSITÀ DEL SALENTO

SCHEDA INSEGNAMENTO

Network Technologies

Corso di studio di riferimento	LM55 - CdL Magistrale in Computer Engineering
Dipartimento di riferimento	Dipartimento di Ingegneria dell'Innovazione
Settore Scientifico Disciplinare	ING-INF/05
Docente	Giovanni Ciccarese
Crediti Formativi Universitari	9
Ore di attività frontale	81
Ore di studio individuale	144
Anno di corso	II anno
Semestre	II
Lingua di erogazione	Inglese
Percorso	PERCORSO COMUNE

Prerequisiti	Fundamentals of Computer Networking, Probability Theory, Markov Chains
Contenuti	This course proposes the study of some fundamental aspects of the operation of modern computer networks, such as traffic control and quality of service, the support of wireless and mobile communications, security. The study includes the analysis of the network technologies which represent the state of the art on the above issues and a computer networks design methodology supported by a number of case studies which concern the selection of the most appropriate technologies depending on their operating contexts. Particularly, the criteria for designing network systems that meet given requirements in terms of performance, reliability and availability are discussed.
Obiettivi formativi	Learning Outcomes. Knowledge and understanding After the course the student should - understand the main issues regarding the operation of a modern computer network and how they could be addressed in order to ensure appropriate delivery of the application services; - know the technologies to be considered in designing a modern computer network and, particularly, understand how they address the aforementioned issues; - know what techniques can be adopted to model and analytically evaluate performance, reliability and availability of network systems. Applying knowledge and understanding After the course the student should be able to - design a computer network with given requirements, selecting the most appropriate technologies depending on the operating context; - configure network devices in a campus network for high availability; - understand scientific literature on the modeling of performance, reliability and availability of network systems.

	Making judgements Students should acquire the ability to identify the pros and cons of each possible solution for both the logical network design and the physical network design. This also applies to the probabilistic techniques described during the lectures with regard to the modeling of performance, reliability, and availability. It is desiderable that students are interested in looking for other techniques by consulting specialized literature. Communication After the course the student should have a good command of topics covered in the course, so as to be able to communicate his/her knowledge and solutions in a clear and simple way, using the specific terminology. The course promotes the development of that skill. Learning skills With the aim of developing learning skills that allow students to continue to study in a way that can be largely autonomous, the instructor suggests some selected technical readings whose level of difficulty is significantly higher than that associated with the exercises covered during the course. They deal with the definition of performance models and/or availability models of large, real-world systems.
Metodi didattici	Lectures and exercises.
Modalità d'esame	The exam is oral. First, the student is asked to solve an exercise in order to verify his/her knowledge and understanding about the stochastic modeling of the performance, reliability and availability of network systems. The remaining part of the exam aims at assessing his/her knowledge and understanding about the issues related to the operation of modern computer networks, about the network technologies studied during the course and about the criteria for their selection in the network design process. Moreover, the student may also be asked to configure some protocols, such as HSRP and RSTP, on the network devices of a campus network for high availability simulated by means of a visual network simulation tool.
Programma	Course content: theory - Introduction to the course ore (2 hours) - Congestion Control and Traffic Control: principles of congestion control, approaches towards congestion control, flow control and congestion control in TCP, TCP/IP ECN. (4 hours) - Multicast in the Internet: algorithms for multicasting, multicast in the Internet (multicast addresses in IPv4, IGMP, multicast routing protocols) (2 hours). - IPv6: IPv6 addressing, stateless address autoconfiguration, IPv6 packet format, ICMPv6, transition from IPv4 to IPv6 (4 hours). - Quality of Service (QoS) in IP networks: multimedia networking applications, protocols for real-time conversational applications(RTP, RTCP,SIP), Quality of Service, Queuing Disciplines, Shaping, Policing, Tocken Bucket, QoS in IP networks (RED, IntServ, DiffServ), MPLS (8 hours). - Wireless and Mobile Networks: wireless channel characteristics, Wireless LANs and IEEE 802.11, CAPWAP, planning a wireless access network, Mobile

	IPv6, Vehicular Ad Hoc Networks and IEEE 802.11p (10 hours). - Network Design: capacity planning, reliability, availability,switched LANs with redundant links (STP, RSTP), Virtual LANs, IEEE 802.1Q, Multiple Spanning Tree Protocol, Private Virtual LANs, Default Router redundancy (HSRP, VRRP, GLBP), top-down network design (design requirements, logical design, physical design, test plan and documentation) (18 hours). Course content: exercises On the design of modern computer networks (26 hours) - A number of case studies which concern the selection of the most appropriate technologies depending on their operating contexts are discussed. Moreover, configuration of network devices in a campus network for high availability is considered. - By using some probabilistic techniques that are PERCORSO COMUNEly employed for modeling computer networks and protocols (Markov Chains, Reliability Block Diagrams, Queueing Theory), a number of examples dealing with modeling of performance, reliability and availability of network systems are proposed.
Testi di riferimento	[1] J. Kurose e K.W. Ross, “Computer Networking. A Top-Down Approach”, seventh edition, Pearson. [2] P. Oppenheimer, “Top-Down Network Design”, third edition, Cisco Press. [3] S. Convery, “Network Security Architecture”, Cisco Press. [4] G. Bolch, S. Greiner, H. de Meer, K.S. Trivedi, "Queueing Networks and Markov Chains: Modeling and Performance Evaluation With Computer Science Applications", Wiley-Interscience. [5] K. Trivedi, A. Bobbio, "Reliability and Availability Engineering: Modeling, Analysis, and Applications", Cambridge University Press.
Altre informazioni utili	Office Hours On Wednesdays, from 15:30 to 18:00