

Allegato B2

Quadro degli obiettivi formativi specifici e delle propedeuticità Corso di Laurea in Internet of Things, Big Data, Machine Learning Curriculum unico Rau, art. 12

Insegnamento	Settore Scientifico Disciplin.	Obiettivi formativi specifici (ITA)	Specific educational objectives (ENG)	Propedeuticità
Algoritmi e strutture dati e laboratorio	INF/01	<p>L'insegnamento di Algoritmi e strutture dati e laboratorio ha come principale obiettivo formativo specifico quello di fornire una preparazione adeguata relativamente alle strutture dati di base, alle tecniche per la progettazione di algoritmi, alla valutazione della loro complessità computazionale e verifica della loro correttezza.</p> <p>Indice:</p> <ul style="list-style-type: none"> • Fondamenti <ul style="list-style-type: none"> ○ <i>Ruolo degli algoritmi</i> ○ <i>Progettazione di algoritmi</i> ○ <i>Correttezza degli algoritmi</i> ○ <i>Complessità computazionale</i> • Strutture Dati di Base e relativi Algoritmi <ul style="list-style-type: none"> ○ <i>Vettori, Heap, Liste, Pile, Code, Alberi Binari, Tabelle di Hash, e varianti.</i> ○ <i>Algoritmi per l'ordinamento, la ricerca, l'inserimento e la cancellazione in strutture dati di base</i> ○ <i>Dimostrazione della correttezza di tali algoritmi</i> ○ <i>Valutazione della complessità computazionale di tali algoritmi</i> • Strutture Dati Avanzate, Algoritmi e Tecniche Avanzate di progettazione <ul style="list-style-type: none"> ○ <i>Bilanciamento di alberi binari con algoritmi</i> 	<p>The teaching of Algorithms and Data Structures and Laboratory has as its main specific educational objective that of providing adequate preparation in relation to basic data structures, techniques for designing algorithms, evaluating their computational complexity and verifying their correctness.</p> <p>Index:</p> <ul style="list-style-type: none"> • Basics <ul style="list-style-type: none"> ○ <i>The role of algorithms</i> ○ <i>Algorithm Design</i> ○ <i>Correctness</i> ○ <i>Computational Complexity</i> • Basic Data Structures and Algorithms <ul style="list-style-type: none"> ○ <i>Arrays, Heaps, Lists, Stacks, Queues, Binary Trees, Hash Tables, and similars.</i> ○ <i>Algorithms for sorting, searching, insertion, deletion over basic data structures</i> ○ <i>Proofs of correctness</i> ○ <i>Computational complexiety evaluations</i> • Advanced Data Structures and Design Techniques <ul style="list-style-type: none"> ○ <i>Balanced binary trees and search/insert/delete algorithms</i> ○ <i>B-trees and</i> 	<p>VINCOLANTI: Programmazione e laboratorio</p> <p>CONSIGLIATE: Analisi Matematica, Elementi di matematica e algebra lineare.</p>

		<p><i>ricerca/inserimento/cancellazione</i></p> <ul style="list-style-type: none">○ <i>B-alberi con algoritmi di ricerca/inserimento/cancellazione</i>	<p><i>search/insert/delete algorithms</i></p> <ul style="list-style-type: none">○ <i>Data structures for disjoint sets</i>	
--	--	--	--	--

		<ul style="list-style-type: none"> ○ <i>Strutture Dati per insiemi disgiunti con operazioni Make, Union, Find e relative euristiche</i> ○ <i>Grafi e algoritmi di base sui grafi</i> ○ <i>Alberi minimi di copertura per grafi pesati</i> ○ <i>Cammini minimi in grafi pesati</i> ○ <i>Dimostrazione della correttezza e valutazione della complessità di tali algoritmi</i> <p>Lo/la studente/essa dovrà:</p> <p>Capacità relative alle discipline</p> <p>1.1. Conoscenza e capacità di comprensione Conoscere i concetti base relativi alle strutture dati statiche e dinamiche e alla progettazione di algoritmi Conoscere un linguaggio formale per la descrizione degli algoritmi Conoscere gli strumenti matematici per la valutazione della complessità computazionale Conoscere le strutture dati e gli algoritmi presentati a lezione e le relative complessità spaziali e temporali</p> <p>1.2 Capacità di applicare conoscenza e comprensione Saper implementare e testare gli algoritmi visti a lezione nei linguaggi di programmazione più comuni Saper utilizzare le strutture dati e gli algoritmi presentati nel corso per progettare nuovi algoritmi in grado di risolvere specifici problemi Saper dimostrare la correttezza e valutare la complessità computazionale di tali algoritmi</p> <p>Capacità trasversali / soft skills</p>	<ul style="list-style-type: none"> ○ <i>Graphs and basic graph algorithms</i> ○ <i>Minimum Spanning Trees</i> ○ <i>Shortest paths problems/algorithms</i> ○ <i>Correctness and Complexity of the above listed algorithms</i> <p>The student should be able to:</p> <p>Sector-specific skills</p> <p>1.1. Knowledge and understanding Manage basic concepts of static and dynamic data structures and design of algorithms. Know a formal language for describing algorithms. Evaluate computational complexities. Know the data structures and algorithms presented during the course together with their spatial and temporal complexities.</p> <p>1.2 Applying knowledge and understanding Implement and test algorithms seen in the course exploiting the most common programming languages. Use the data structures and the algorithms presented in the course to design new algorithms that can solve specific problems. Prove the correctness and evaluate the computational complexities of such algorithms</p> <p>Cross-sectoral skills/soft skills</p> <p>2.1 Making judgements Determine which data structures, algorithms and design techniques use to provide efficient solutions to a given algorithmic problem. Choose the most computationally efficient solution. Combine/extend existing algorithms to solve a</p>	
--	--	---	--	--

		<p>2.1 Autonomia di giudizio Saper valutare quali strutture dati, quali algoritmi e quali tecniche di progettazione utilizzare per fornire una soluzione efficiente ad un dato problema algoritmico. Saper scegliere tra più possibili soluzioni quella computazionalmente più efficiente. Saper combinare/estendere algoritmi esistenti per risolvere uno specifico problema Saper dimostrare o confutare la correttezza e valutare la complessità computazionale di algoritmi da lui/lei progettati o reperiti in letteratura. Saper selezionare il linguaggio di programmazione e le librerie più adatte per l'implementazione degli algoritmi. Saper testare le implementazioni.</p> <p>2.2 Abilità comunicative. Essere in grado di motivare, le scelte effettuate in termini di strutture dati e algoritmi per la risoluzione di un dato problema. Essere in grado di reperire le eventuali informazioni aggiuntive necessarie per la progettazione di un algoritmo efficiente.</p> <p>2.3 Capacità di apprendimento Saper reperire ed utilizzare risorse informatiche e scientifiche per l'approfondimento autonomo delle tematiche studiate a lezione. Saper affrontare lo studio di nuove problematiche di natura algoritmica. Saper individuare nuovi strumenti per lo sviluppo e l'implementazione di strutture dati e algoritmi</p>	<p>specific problem. Demonstrate or refute the correctness and evaluate the computational complexity of algorithms. Select the programming language and the most suitable libraries for implementing an algorithm. Test implementations.</p> <p>2.2 Communication skills. Motivate, make choices in terms of data structures and algorithms for solving a given problem. Find any additional information needed to design an efficient algorithm.</p> <p>2.3 Learning skills Find and exploit scientific resources for the in-depth study of topics studied in the course. Deal with new problems of algorithmic nature. Identify new tools for developing data structures and algorithms.</p>	
Analisi Matematica	MAT/05	Il corso di Analisi Matematica vuole fornire una solida preparazione di base nel calcolo differenziale e integrale in una variabile. La	The Calculus course gives a solid grounding in the differential and integral calculus in one variable. There are two purposes: familiarizing	

	<p>finalità è duplice: familiarizzare il discente con il metodo dimostrativo in generale, e nel contempo fornire nozioni senza le quali si è tagliati fuori dalla comprensione della statistica, dell'analisi dei dati numerici, nonché di buona parte dell'informatica teorica. La teoria viene presentata senza rinunciare a un certo livello di rigore formale, modulando il passo con la progressiva familiarità che gli studenti acquisiscono con il metodo logico-deduttivo.</p> <p>Enfasi viene data al significato geometrico intuitivo dei concetti di limite, derivata e integrale.</p> <p>Argomenti: nozioni di logica, insiemi numerici con operazioni e ordinamento, induzione matematica, limiti, derivate, integrali in una variabile.</p> <p>Capacità relative alle discipline</p> <p><i>Conoscenza e capacità di comprensione:</i> conoscere il significato geometrico e algebrico di limiti, derivate, integrali.</p> <p><i>Conoscenza e capacità di comprensione applicate:</i> saper manipolare in modo logicamente corretto espressioni contenenti limiti, derivate, integrali e saperle usare per risolvere problemi.</p> <p>Capacità trasversali/soft skills</p> <p><i>Autonomia di giudizio:</i> saper collegare fra loro idee algebriche, geometriche e computazionali.</p> <p><i>Abilità comunicative:</i> saper scrivere in modo comprensibile la risoluzione di un problema matematico, e saper dimostrare oralmente un teorema.</p> <p><i>Capacità di apprendimento:</i></p>	<p>the student with the demonstrative method in general, and, at the same time, teach the notions that are a prerequisite for understanding statistics, numerical data analysis and much of theoretical computer science. The theory is explained without forfeiting some level of formal rigour, gradually accompanying the student in the discovery of the logical-deductive method.</p> <p>Emphasis is given to the intuitive geometric meaning of the notions of limit, derivative and integral.</p> <p>Topics: Notions of logic, numerical sets with algebraic operations and ordering, mathematical induction, limit, derivative, integral in one variable.</p> <p>Sector-specific skills</p> <p><i>Knowledge and understanding:</i> The algebraic and geometrical meaning of limit, derivative, integral.</p> <p><i>Applying knowledge and understanding:</i> Being able to manipulate in a logically correct way expressions involving limits, derivative, integral, and using them to solve problems.</p> <p>Cross-sectoral skills/soft skills</p> <p><i>Making judgements:</i> Being able to make useful connections between algebraic, geometric and computational ideas.</p> <p><i>Communication skills:</i> Being able to write a report on how to solve a mathematical problem, and to prove a theorem orally.</p> <p><i>Learning skills:</i></p>	
--	---	---	--

		saper leggere un testo scientifico che usa il metodo dimostrativo e gli strumenti di base dell'analisi matematica.	Being able to read a scientific text that uses the demonstrative method and the basic notions of calculus.	
Architettura degli elaboratori	INF/01	<p>Scopo del corso è descrivere la struttura e il funzionamento dei calcolatori nelle loro diverse componenti, illustrando le principali tecnologie utilizzate.</p> <p>Il percorso didattico inizia con lo studio delle parti elementari che costituiscono un computer e considera componenti sempre più complesse fino ad arrivare allo studio di architetture complete di calcolatori.</p> <p>Argomenti trattati sono: la progettazione di semplici circuiti logici, le codifiche hardware dell'informazione, la struttura del processore, il collegamento e la comunicazione tra periferiche e processore, il tema della memoria (cache, centrale, di massa, virtuale), la programmazione in linguaggio assembly.</p> <p>Capacità relative alle discipline</p> <p>1.1 Conoscenza e capacità di comprensione: dopo aver superato l'esame si ritiene che lo studente sia in grado di comprendere il funzionamento dell'hardware e conoscere i fattori che influenzano le prestazioni dei calcolatori.</p> <p>1.2 Conoscenza e capacità di comprensione applicate: lo studente acquisisce la capacità di analizzare e progettare semplici circuiti digitali, e di comprendere l'informazione scritta in forma binaria.</p> <p>Capacità trasversali/soft skills</p> <p>2.1 Autonomia di giudizio: lo studente acquisisce gli strumenti per poter valutare e scegliere in</p>	<p>The course aims at describing the structure and operations of the diverse components forming a computer, by illustrating the most popular technologies.</p> <p>The course starts with an analysis of the basic elements of a computer, and then it considers components that are progressively more complex until arriving at the study of complete computer architectures.</p> <p>The following arguments will be dealt with: design of simple logic circuits, hardware coding of the information, structure of a processor, communication between a processor and its peripherals, memory (central, cache, mass, virtual, programming in assembly language.</p> <p>The lectures include laboratory sessions where students will exercise on logic circuit design and machine language programming.</p> <p>Sector-specific skills</p> <p>1.1 Knowledge and understanding: past the exam, students will understand how computer hardware works; furthermore, they will know the main factors that influence the performance of a computer.</p> <p>1.2 Applying Knowledge and understanding: the laboratory activity will allow students to get used with the low-level design of both simple digital systems and programs in machine language, i.e. directly executable by the computer.</p>	

		<p>maniera ponderata l'architettura hardware da utilizzare.</p> <p>2.2 Abilità comunicative: lo studente impara l'esatto significato dei vari termini usati per descrivere le caratteristiche e le specifiche di un calcolatore ed usarle in maniera appropriata.</p> <p>2.3 Capacità di apprendimento: il corso mira a fornire allo studente un quadro concettuale sufficientemente rigoroso completo da permettergli di affrontare con sicurezza lo studio dettagliato delle varie componenti di un calcolatore.</p>	<p>Cross-sectoral skills/soft skills</p> <p>2.1 Making Judgements: students will learn tools for evaluating with ponder and choosing a hardware architecture.</p> <p>2.2 Communication skills: students will learn the exact meaning of the different words used for properly describing characteristics and specifications of a computer.</p> <p>2.3 Learning skills: the course aims at providing a conceptual framework, precise enough to secure a detail study of the diverse components finding place in a computer.</p>	
Basi di dati e laboratorio	INF/01	<p>Obiettivo fondamentale del corso è l'acquisizione dei concetti, degli strumenti e delle metodologie fondamentali nel campo delle basi di dati, con particolare attenzione ai modelli (concettuale, logico e fisico), ai linguaggi (di definizione, di aggiornamento e di interrogazione) e all'architettura dei sistemi per basi di dati.</p> <p>Vengono fornite le nozioni fondamentali inerenti ai linguaggi per la definizione, l'interrogazione e l'aggiornamento dei dati (algebra relazionale, calcolo relazionale, SQL). Vengono inoltre forniti elementi di progettazione concettuale (raccolta e analisi dei requisiti, costruzione di modelli Entità/Relazioni), logica (ristrutturazione di schemi concettuali, trasformazione di schemi concettuali in schemi logici, normalizzazione dei dati) e fisica (strutture di indicizzazione). Infine, viene introdotta la nozione di transazione e vengono analizzate le componenti principali di un sistema di gestione dei dati. Dopo aver superato l'esame si ritiene che lo studente sia in grado di</p>	<p>The overall aim of the course is to learn the concepts, tools and fundamental methodologies in the realm of database management, with a special emphasis on conceptual, logical, and physical modeling, on data definition and manipulation languages, and on database system architecture. The fundamental notions regarding definition and query languages are explained, as well as elements of conceptual modeling (from requirement elicitation and analysis to the construction of Entity-Relationship models), logical modeling (including schema restructuring, translation into the Relational Model, and data normalization), and physical modeling (indexing). Finally, the student will learn about transaction management and will acquire an understanding of the main components of a database management system. After this course, the student will be able to formalize into a relational language</p>	<p>CONSIGLIATE: Algoritmi e strutture dati e laboratorio, Sistemi operativi e laboratorio</p>

	<p>formalizzare in un linguaggio relazionale operazioni di definizione e manipolazione dei dati espresse in linguaggio naturale e di progettare semplici basi di dati a livello concettuale (costruzione di schemi Entità/Relazioni a partire da insiemi di requisiti espressi in linguaggio naturale), logico (trasformazione di schemi Entità/Relazioni in schemi relazionali, formalizzazione della semantica intesa delle relazioni tramite dipendenze funzionali, normalizzazione di schemi relazionali) e fisico (definizione degli opportuni indici).</p> <p>Capacità relative alle discipline</p> <p>1.1. Conoscenza e capacità di comprensione</p> <ul style="list-style-type: none"> ○ Concetto di “modello dei dati”. ○ Modelli per la progettazione concettuale di una base di dati. ○ Il modello relazionale. ○ Linguaggi formali per la definizione e la manipolazione dei dati: algebra relazionale e calcolo relazionale. ○ Elementi architetturali di un sistema di gestione dei dati. ○ Politiche e meccanismi per la gestione concorrente delle transazioni. ○ Algoritmi e strutture dati per l’indicizzazione efficiente dei dati. <p>1.2. Conoscenza e capacità di comprensione applicate</p> <ul style="list-style-type: none"> ○ Elementi di progettazione concettuale: analisi e raccolta dei requisiti, diagrammi Entità/Relazione. ○ Progettazione logica e fisica di una base di dati. ○ Il linguaggio SQL. 	<p>operations on data informally expressed in a natural language. The student will also be able to understand and develop conceptual database designs (from requirements to E-R diagrams), logical database designs (from the translation of E-R models into relational schemas to the formalization of relational semantics through functional dependencies and schema normalization), and physical database designs (definition of index data structures).</p> <p>Sector-specific skills</p> <p>1.1. Knowledge and understanding</p> <ul style="list-style-type: none"> ○ Notion of “data model”. ○ Models for conceptual database design. ○ The Relational Model. ○ Formal languages for data definition and manipulation: Relational Algebra and Relational Calculus. ○ Architecture of a database management system. ○ Policies and mechanisms for concurrent transaction management. ○ Algorithms and data structures for efficient data indexing. <p>1.2. Applying knowledge and understanding</p> <ul style="list-style-type: none"> ○ Elements of conceptual design: requirement analysis, Entity/Relationship diagrams. ○ Logical and physical database design. ○ The SQL language. ○ Stored procedures, user-defined functions, and triggers. ○ Case studies: PostgreSQL and MySQL. 	
--	---	---	--

		<ul style="list-style-type: none"> ○ Procedure memorizzate, funzioni definite dall'utente e trigger. ○ Casi di studio: i sistemi di gestione dei dati PostgreSQL e MySQL. <p>Capacità trasversali / soft skills</p> <p>2.1. Autonomia di giudizio</p> <ul style="list-style-type: none"> ○ Analizzare e formalizzare i requisiti fondamentali di un sistema informativo. ○ Confrontare soluzioni diverse di un problema di gestione dei dati. ○ Formulare in modo corretto interrogazioni complesse. ○ Valutare l'opportunità o meno di usare strutture d'indicizzazione per l'ottimizzazione delle interrogazioni. <p>2.2. Abilità comunicative</p> <ul style="list-style-type: none"> ○ Interagire con i soggetti direttamente o indirettamente coinvolti in un progetto di gestione dei dati. ○ Motivare le scelte poste in atto in tutte le fasi della progettazione di un sistema informativo. <p>2.3. Capacità di apprendimento</p> <ul style="list-style-type: none"> ○ Apprendere in modo autonomo il funzionamento di ulteriori software di gestione di gestione dei dati. ○ Approfondire ulteriori tematiche inerenti alle basi dei dati, quali, ad esempio, le basi di dati XML, i data warehouse, la sicurezza e protezione dei dati. ○ Affrontare le problematiche di gestione dei dati in applicazioni specifiche, come ad esempio, le basi di dati biologici o spazio-temporali. 	<p>Cross-sectoral skills/soft skills</p> <p>2.1. Making judgments</p> <ul style="list-style-type: none"> ○ Analyze and formalize the fundamental requirements of an information system. ○ Compare different solutions to a data management problem. ○ Correctly formulate complex queries. ○ Evaluate the opportunity of using indexes for query optimization. <p>2.2. Communication skills</p> <ul style="list-style-type: none"> ○ Interact with (technical and non-technical) stakeholders involved in a data management project. ○ Convincingly argue about the decisions taken during all the phases of the design of an information system. <p>2.3. Learning skills</p> <ul style="list-style-type: none"> ○ Independently learn how to use other database management software. ○ Delve into further topics in data management, such as XML databases, data warehouses, data security and privacy. ○ Tackle data management issues in specific applications, such as biological or spatio-temporal databases. 	
--	--	--	--	--

<p>Fisica per i dispositivi IOT Physics for IOT devices</p>	<p>FIS/01</p>	<p>Contenuti disciplinari: Cenni di elettrostatica: carica elettrica, campo elettrico, potenziale elettrico, capacità elettrica. Conduzione elettrica nei materiali: corrente elettrica, resistenza elettrica, potenza. Conduttori ohmici. Circuiti elettrici: elementi circuitali base, circuiti eleggi di Kirchhoff, circuiti RC. Il campo magnetico. La forza di Lorenz. L'energia associata al campo magnetico. Induzione ed induttanza. Generatori e motori elettrici. Circuiti in corrente alternata. Dispositivi a semiconduttore: la conduzione nei semiconduttori, il diodo, il transistor, il LED. La trasduzione e alcuni sensori semplici: Resistori, Fotodiodi e Fotoresistori, Termistori. Come esiti del corso si attende che gli studenti abbiano Capacità relative alle discipline 1.1 Conoscenze e capacità di comprensione: abbiano acquisito gli elementi base nel campo della teoria dei circuiti elettrici, dei dispositivi a semiconduttore e di alcuni sensori di tipo più comune. conoscano i principi di funzionamento ed utilizzo della strumentazione usata in un laboratorio per il montaggio e l'analisi di semplici dispositivi. 1.2 Capacità di applicare conoscenza e comprensione: siano capaci di applicare le conoscenze acquisite per analizzare e/o progettare semplici dispositivi basati su sensori. Capacità trasversali 2.1 Abilità comunicative: siano in grado di esporre in modo efficace</p>	<p>Disciplinary contents: Elements of electrostatics: electric charge, electric field, electric potential, capacitance. Electrical conduction in materials: electrical current, electrical resistance, power. Ohmic conductors. Electrical circuits: basic circuit elements, circuits and Kirchhoff's laws, RC circuits. The magnetic field. Lorenz's force. The energy associated to the magnetic field. Induction and inductance. Electric generators and motors. AC circuits. Semiconductor devices: conduction in semiconductors, the diode, the transistor, the LED. Transduction and some simple sensors: Resistors, Photodiodes and Photoresistors, Thermistors. As course outcomes, students are expected to have Framework skills 1.1 Knowledge and understanding: They have acquired the basic elements in the field of electrical circuit theory, semiconductor devices and some of the more common type sensors. They know the principles of operation and use of the instrumentation used in a laboratory for the assembly and analysis of simple devices. 1.2 Ability to apply knowledge and understanding: They are capable of applying the knowledge acquired to analyse and / or design simple sensor-based devices Transversal Skills 2.1 Communication skills:</p>	
---	---------------	---	--	--

		<p>i principi di funzionamento di un dispositivo e le motivazioni per le scelte effettuate nella fase della sua progettazione</p>	<p>They are able to effectively expose the operating principles of a device and the reasons for the choices made in its design phase.</p>	
--	--	---	---	--

<p>Elementi di matematica e algebra lineare Elements of mathematics and linear algebra</p>	<p>MAT/01</p>	<p>Obiettivi Formativi Specifici.</p> <p>Questo corso, primo fra i corsi di matematica che gli studenti del corso di IBW devono affrontare, ha come primo obiettivo riprendere alcuni argomenti fondamentali del linguaggio matematico di base, in parte già affrontati durante il percorso scolastico dello studente, quali il linguaggio insiemistico, la teoria delle funzioni e relazioni, l'aritmetica e il calcolo combinatorio e di utilizzarli come palestra graduale per sviluppare capacità analitiche, dimostrative e modellistiche. Nella seconda parte del corso si introducono i primi concetti di algebra lineare e si presentano esempi di applicazione dei suoi strumenti alla risoluzione di sistemi lineari e alla rappresentazione dei movimenti nel piano e nello spazio. Queste nozioni vengono poi generalizzate a dimensioni superiori.</p> <p>La prima parte del corso è suddivisa in quattro parti:</p> <p>1. Introduzione al linguaggio matematico elementare Obiettivo di questa parte è porre le basi per riuscire ad esprimersi correttamente nel linguaggio</p>	<p>Specific educational objectives</p> <p>This is the first course of mathematics for IBW students and its first objective is to reconsider some fundamental arguments of the elementary mathematical language, which in part are already considered in high school, such as set theory, functions and relations, arithmetics and combinatorics, and use them as a training ground to develop analytical and proof oriented skills. In the second part of the course the first notions of linear algebra are introduced and applied to the solution of linear systems and to the representation of motion in plane and space. These concepts are then generalized to higher dimensions.</p> <p>The first notions of linear algebra are introduced and applied to the solution of linear systems and to the representation of motion in plane and space. These concepts are then generalized to higher dimensions.</p> <p>The course is divided into four parts:</p> <p>1. Introduction to the elementary language of mathematics In this part we do the groundwork needed for the correct use of the mathematical language.</p>	
--	---------------	---	--	--

	<p>matematico. Verranno analizzate e messe a confronto diverse notazioni e definizioni insiemistiche e si svilupperanno dei primi ragionamenti astratti in questo ambito.</p> <p>2. Funzioni e relazioni Si introducono le prime definizioni facendo largo uso di esempi atti a superare le difficoltà dovute all'aumentare del livello di astrazione, come ad esempio quando si considera l'insieme delle classi di una relazione d'equivalenza.</p> <p>3. Aritmetica e congruenze Prima vera palestra per lo sviluppo di capacità dimostrative. Queste tematiche vengono utilizzate per esemplificare l'uso della matematica elementare nelle applicazioni, ad esempio in crittografia.</p> <p>4. Calcolo combinatorio Vengono analizzate diverse tecniche per il calcolo delle cardinalità di insiemi finiti. Queste tematiche sono adatte per imparare a modellare problemi che nascono in ambiti diversi e mettere a frutto le capacità dimostrative acquisite nei punti precedenti.</p> <p>La seconda parte del corso è a sua volta divisa in quattro parti.</p> <p>1. Numeri reali e complessi. Si considera il passaggio dai numeri reali ai numeri complessi (anche da un punto di vista storico) ed i paradossi</p>	<p>We analyze and compare different set notations and definitions, and we develop the first abstract reasonings using them.</p> <p>2. Functions and relations We introduce the first definitions using examples that allows to overcome the first difficulties we met when the level of abstraction increases, as when we consider the set of equivalence classes.</p> <p>3. Arithmetics and congruences The first training ground for developing reasoning skills. We use this material as an example of the use of elementary mathematics in applications, for example in cryptography.</p> <p>4. Combinatorics We analyze different techniques for the calculus of the cardinality of finite sets. This setting is appropriate for learning how to model different problems and use the reasoning skills acquired in the previous parts of the program.</p> <p>The second part of the course is also divided into four parts.</p> <p>1. From real to complex number. The transition from real to complex numbers (also from an historical point of view) and the paradoxes related to the cardinality of infinite sets are considered.</p>	
--	---	---	--

	<p>relativi alla nozione di cardinalità per insiemi infiniti.</p> <p>2. Sistemi lineari e la loro rappresentazione matriciale. In questa parte lo studente apprende come una rappresentazione matematica possa essere utile per rappresentare e risolvere concisamente un problema.</p> <p>3. Definizioni e tecniche dell'algebra lineare. Le definizioni (dipendenza, basi,...) sono prima introdotte nel piano e nello spazio e poi generalizzate a dimensioni superiori. Obiettivo di questa parte è di esporre lo studente in maniera graduale ad un maggior grado di astrazione.</p> <p>4. Trasformazioni lineari e la loro rappresentazione tramite matrici. Si considerano in primo luogo rotazioni e simmetrie nel piano e nello spazio e poi dimensioni superiori. Obiettivo di questa parte è di esemplificare come certi movimenti nel piano e nello spazio possano essere riprodotti nel computer attraverso la loro rappresentazione matematica.</p> <p>Capacità relative alle discipline</p> <p>1.1 Conoscenza e capacità di comprensione Conoscere i concetti di base della matematica discreta e dell'Algebra Lineare. Conoscere la simbologia Matematica nel continuo e nel discreto. Saper enunciare correttamente ed esemplificare i teoremi studiati.</p> <p>1.2 Conoscenza e capacità di comprensione applicate Saper sviluppare semplici dimostrazioni e modellare matematicamente semplici problemi combinatori. Saper formalizzare un problema usando le tecniche dell'algebra lineare.</p>	<p>2. Linear systems and their matrix representation. Here the student learn how a mathematical representation can be useful to represent and solve a problem concisely.</p> <p>3. Definitions and tools of linear algebra The definitions (dependence, basis,..) are first introduced in plane and space and then generalized to higher dimensions. The objective is here to expose the student gradually to a higher level of abstraction.</p> <p>4. Linear transformations and their matrix representation Starting from rotations and symmetries in plane and space the concepts are then generalized to higher dimension. Here we exemplify how certain motion can be reproduced in the computer using their mathematical representation.</p> <p>Framework skills</p> <p>1.1 Knowledge and understanding To know the basic concept of discrete mathematics and linear algebra. Know the mathematical symbology in the continuous and in the discrete. Knowing how to correctly state and exemplify the theorems studied.</p> <p>1.2 Applying knowledge and understanding To know how to develop easy proofs and how to build mathematical models of easy combinatorial problems. To know how to formalize real problems using the techniques of linear algebra.</p>	
--	---	--	--

		<p>Capacità trasversali</p> <p>2.1 Autonomia di giudizio Saper riconoscere un ragionamento corretto. Saper valutare quale strumento utilizzare per risolvere uno specifico problema.</p> <p>2.2 Abilità comunicative. Saper formalizzare le proprie intuizioni e spiegare in modo comprensibile una dimostrazione matematica. Saper scrivere in modo conciso la soluzione di un problema.</p> <p>2.3 Capacità di apprendimento Essere in grado di approfondire autonomamente gli argomenti trattati nel corso.</p>	<p>Cross-sectoral skills/soft skills</p> <p>2.1 Making judgements To be able to recognize a correct reasoning. To know what tool is needed to solve a specific problem.</p> <p>2.2 Communication skills To know how to communicate intuitions and explain in a clear way a mathematical proofs.</p> <p>2.3 Learning skills Be able to go deeper independently the topics covered in the course.</p>	
--	--	--	---	--

<p>Fondamenti di Scienza dei Dati e Laboratorio</p>	<p>INF/01</p>	<p>Obiettivi formativi specifici (ITA)</p> <p>Uno dei fattori principale contraddistingue l'attuale era tecnologica da quelle precedenti è la maggiore quantità giornaliera di dati prodotti. Infatti, ogni giorno, la maggior parte dei nostri dispositivi, come smartphone o computer, dispositivi IoT, automobili, sensori ambientali e così via, generano o raccolgono dati che vengono generalmente condivisi tramite internet e salvati in database remoti. Come ben noto alle grandi aziende tech tali dati, inizialmente grezzi, non sono molto interessanti di per sé. Tuttavia, possono diventare preziosissimi se adeguatamente studiati e analizzati attraverso studi di Scienza dei Dati. Questo corso ti insegnerà le principali tecniche, metodologie e strumenti di Scienza dei Dati accompagnandoti nel sentiero della Scienza dei Dati che inizia dall'importazione dei dati nel tuo progetto, attraversa la loro analisi, fino alle fasi di visualizzazione e comunicazione dei tuoi risultati. Tale percorso è anche noto con il nome di Data Science workflow e, attraverso l'uso di Python, questo corso ti guiderà passo dopo passo in tutte le sue sei fasi, vale a dire:</p> <ul style="list-style-type: none"> - Importazione: imparare a leggere i dati in diversi formati e importarli nell'ambiente di lavoro; - Normalizzazione: portare i dati in forma normale allo scopo di favorire l'analisi; - Trasformazione: interrogare e manipolare il dataset, ad esempio filtrando solo i dati significativi o aggiungendo nuove variabili funzione di variabili già esistenti; - Visualizzazione: visualizzare alcune variabili del dataset allo scopo di verificare alcune ipotesi o porti ulteriori domande; 	<p>Specific educational objectives (ENG)</p> <p>One main factor differentiating the current technology era from previous ones is the daily amount of data produced. Indeed, every day, most of our devices, like smartphones or computers, IoT devices, cars, environmental sensors, and so on, generate or collect data that is typically shared through the internet and saved in some remote database. As big tech companies know very well, thus raw data is not interesting as they are, but they can become much more valuable if adequately studied and analyzed through data science studies. This course will teach you the main data science techniques, methodologies and tools accompanying you through the data science path that goes from the import of the data in your project, through its analysis, to the final steps of visualization and communicating your results. Such a path is known as data science workflow, and through the use of Python, this course will guide you step by step on all his six phases, namely:</p> <ul style="list-style-type: none"> - Import: learning to read data in different formats and importing them into the working environment; - Normalization: bring the data in normal form to facilitate the analysis; - Transformation: query and manipulate the dataset, e.g. filtering only meaningful data or adding new variables function of already existing variables; - Visualization: display some variables of the dataset to verify some hypotheses or ask further questions; 	<p>Propedeuticità</p> <p>CONSIGLIATI: Programmazione e laboratorio, Statistica e laboratorio</p>
---	---------------	--	---	---

	<p>- Modellazione: creare modelli di regressione lineare per investigare ulteriormente il dataset; - Comunicazione: imparare a comunicare in modo efficace e accattivante i risultati dell'analisi.</p> <p>Successivamente, il corso tratta applicazioni tipiche di scienza dei dati, come la gestione dei big data, le serie temporali, l'elaborazione automatica del linguaggio naturale e la trattazione dati geografici.</p> <p>Capacità relative alle discipline</p> <p>1.1 Conoscenza e capacità di comprensione Lo studente acquisirà la seguente conoscenza: Procedure formali per risolvere problemi di apprendimento automatico e analizzare i risultati ottenuti; Metodi per importare, riordinare, trasformare, visualizzare e modellare i dati; Tecniche di elaborazione e diffusione dei risultati delle analisi di dati; Acquisizione di conoscenze necessarie per gestire l'intero flusso della scienza dei dati, dall'importazione dei dati alla comunicazione dei risultati.</p> <p>1.2 Conoscenza e capacità di comprensione applicate: Lo studente: Analizzerà i dati con gli strumenti della scienza dei dati; Comunicherà i risultati delle analisi effettuate.</p> <p>Capacità trasversali/soft skills</p> <p>2.1 Autonomia di giudizio Lo studente deve essere in grado di interpretare i risultati sperimentali delle analisi e trarre conclusioni relative al dominio del discorso.</p>	<p>- Modelling: create linear regression models or others to investigate the dataset further; - Communication: learning to communicate the results of the analysis in an effective and engaging way.</p> <p>Then, the course will discuss typical use cases and applications of Data Science, such as big data management, temporal series, natural language processing and geographic data.</p> <p>Discipline-related capabilities</p> <p>1.1 Knowledge and understanding The student will acquire the following knowledge: Formal procedures to solve machine learning problems and analyze the obtained results; Methods for importing, sorting, transforming, visualizing and modeling data; Techniques for processing and disseminating the results of data analyses; Gaining the knowledge needed to manage the whole data science flow, from importing data to reporting results.</p> <p>1.2 Applied knowledge and understanding The student will: Analyze data with data science tools; Communicate the results of the analyzes carried out.</p> <p>Cross skills / soft skills</p> <p>2.1 Autonomy of judgment The student must be able to interpret the experimental results of the analysis and draw conclusions regarding the topic domain.</p> <p>2.2 Communicative skills</p>	
--	--	--	--

		<p>2.2 Abilità comunicative Lo studente deve essere in grado di comunicare in modo efficace e accattivante i risultati delle analisi. Verranno valutate chiarezza espositiva, organizzazione dei contenuti, cura dei dettagli, narrativa della presentazione e appeal grafico.</p> <p>2.3 Capacità di apprendere Lo studente deve dimostrare di aver acquisito la capacità di scegliere un significativo caso di studio, analizzare i dati per estrarre informazioni importanti, trarre conclusioni e saperle comunicare efficacemente.</p>	<p>The student must be able to communicate the analysis results effectively and engagingly. The evaluation will regard expository clarity, content organization, attention to detail, presentation narrative, and graphic appeal.</p> <p>2.3 Learning ability The students will be evaluated on: expository clarity, organization of contents, attention to detail, presentation narrative, and graphic appeal.</p>	
--	--	---	--	--

<p>Ingegneria del Software</p>	<p>ING-INF/05</p>	<p>L'obiettivo fondamentale del corso è l'acquisizione dei concetti, degli strumenti e delle metodologie fondamentali per l'ingegnerizzazione di sistemi software di qualità. Al termine del corso, gli studenti avranno acquisito familiarità con le nozioni di base riguardanti (I) il ciclo di vita del Software, (II) l'analisi e la specifica dei requisiti, (III) la progettazione e l'analisi di sistemi software e (IV) modelli e tecniche per garantire la qualità del software, con particolare attenzione alle tecniche di verifica e la validazione. Inoltre essi avranno appreso l'uso del linguaggio di modellazione UML. Tale linguaggio sarà impiegato per esercitazioni al computer, con l'obiettivo di mettere gli studenti in condizione di risolvere problemi pratici di analisi, progettazione e sviluppo software.</p> <p>Capacità relative alle discipline</p> <p>1.1. Conoscenza e capacità di comprensione Gli studenti dovranno acquisire specifiche conoscenze dei principali concetti e principi dell'Ingegneria del Software. In particolare, essi dovranno dimostrare: di conoscere i principali modelli di ciclo di vita del software; di conoscere le tecniche di analisi e specifica dei requisiti; di conoscere tecniche di analisi e di progettazione di sistemi software attraverso il linguaggio di modellazione UML; di conoscere le principali tecniche di verifica e validazione del software, con particolare attenzione ai principi alla base del testing; conoscere i principali parametri di qualità ed i principali standard del software; conoscere le tecniche agili di sviluppo del software.</p> <p>1.2 Capacità di applicare conoscenza e comprensione Gli studenti dovranno dimostrare di essere in grado di: specificare i requisiti e di saper progettare un sistema software su piccola scala attraverso il linguaggio UML; saper selezionare le metodologie e</p>	<p>The goal of the course is to acquire concepts, tools, and fundamental methodologies for the engineering of high-quality software systems. At the end of the course, students will be familiar with basic notions concerning (I) Software life-cycle, (II) Requirements analysis and specification, (III) Design and analysis of software systems, and (IV) Models and techniques to ensure software quality, with particular attention to verification and validation techniques. Additionally, they will learn the use of the UML modeling language. This language will be employed for computer-based exercises, with the aim of enabling students to solve practical problems of software analysis, design, and development.</p> <p>Skills related to the disciplines:</p> <p>1.1. Knowledge and understanding: Students will acquire specific knowledge of the main concepts and principles of Software Engineering. In particular, they should demonstrate knowledge of: the main software life-cycle models; techniques for requirements analysis and specification; analysis and design techniques for software systems using the UML modeling language; the main techniques for software verification and validation, with a particular focus on testing principles; the main quality parameters and software standards; and agile software development techniques.</p> <p>1.2. Ability to apply knowledge and understanding: Students should be able to specify requirements and design a small-scale software system using the UML language;</p>	<p>VINCOLANTE: Programmazione e laboratorio</p> <p>CONSIGLIATI: Programmazione Orientata agli Oggetti e Laboratorio</p>
--------------------------------	-------------------	---	--	---

	<p>le tecniche di sviluppo più adeguate a diversi contesti e domini applicativi; saper progettare casi di test rilevanti e valutare la qualità del software.</p> <p>Capacità trasversali / soft skills</p> <p>2.1 Autonomia di giudizio: Saper valutare indipendentemente le caratteristiche ed i requisiti di un'applicazione informatica. Effettuare e rappresentare l'analisi e la progettazione di software che risponda a requisiti predefiniti. Saper valutare la qualità del software attraverso opportuni casi di test. Saper adattare le tecniche apprese a diversi domini applicativi.</p> <p>2.2 Abilità comunicative: saper illustrare con rigore logico e terminologico gli elementi rilevanti di un progetto software, mediante le tecniche di rappresentazione ritenute più opportune e motivando le scelte effettuate.</p> <p>2.3 Capacità di apprendimento: Essere in grado di apprendere le nozioni base dell'Ingegneria del Software fornite a lezione. Saper reperire ed utilizzare risorse informatiche e scientifiche per l'approfondimento autonomo delle tematiche studiate a lezione. Saper individuare nuovi strumenti per l'analisi e lo sviluppo di sistemi software di qualità.</p>	<p>select appropriate development methodologies and techniques for different contexts and application domains; design relevant test cases and evaluate software quality.</p> <p>Transversal skills / soft skills:</p> <p>2.1. Autonomy of judgment: Be able to independently evaluate the characteristics and requirements of a software application. Conduct and represent analysis and design of software that meets predefined requirements. Assess software quality through appropriate test cases. Adapt learned techniques to different application domains.</p> <p>2.2. Communication skills: Be able to rigorously and logically illustrate the relevant elements of a software project, using appropriate representation techniques and justifying the choices made.</p> <p>2.4. Learning ability: Be able to learn the basic concepts of Software Engineering provided in lectures. Identify and use computer and scientific resources for independent study of topics covered in lectures. Identify new tools for the analysis and development of high-quality software systems.</p>	
--	--	---	--

<p>Interazione Uomo-Macchina Interazione persona-macchina</p>	<p>INF/01</p>	<p>Scopo del corso è di introdurre i principi, le metodologie di progettazione e le diverse scelte implementative per la costruzione di software che sia usabile in modo semplice, intuitivo, produttivo, affidabile e piacevole dagli utenti a cui è rivolto. Il raggiungimento di questo obiettivo richiede lo studio di tre diverse tematiche: i fattori umani (caratteristiche psicologiche dell'utente); la macchina (possibilità offerte dalle periferiche di input/output esistenti); l'interazione (analisi, progetto, valutazione di interfacce utente). Oltre a presentare le nozioni di base della disciplina, il corso pone anche l'accento su alcuni sviluppi recenti di particolare importanza, quali il groupware, le interfacce 3D, i social network e la comunicazione fra persone in rete. Il corso prevede anche la partecipazione ad una reale valutazione su utenti di un'interfaccia in modo da consentire allo studente di veder applicate concretamente le tecniche di valutazione apprese nelle lezioni.</p>	<p>The aim of the course is to introduce the principles, design methodologies and the various implementation choices to build software that can be used in a simple, intuitive, productive, reliable, and enjoyable way by the users to whom it is addressed. The achievement of this goal requires the study of three different themes: human factors (psychological characteristics of the user); the machine (possibility offered by existing input / output devices); interaction (analysis, design, evaluation of user interfaces). In addition to presenting the core concepts of the discipline, the course also emphasizes some recent major developments such as groupware, 3D interfaces, social networks, and on-line communication between people. The course also includes participation in a real evaluation of a user interface so that the student can effectively apply the evaluation techniques learned in the course.</p>	<p>VINCOLANTI: Programmazione e laboratorio, Architettura degli elaboratori</p>
---	---------------	---	---	---

		<p>Capacità relative alle discipline</p> <p>1.1 Conoscenza e capacità di comprensione: gli studenti acquisiscono conoscenze specifiche multidisciplinari sulle caratteristiche degli utenti e delle interfacce ad essi rivolte. Essi inoltre imparano a valutare ed a scegliere fra varie tecniche di interazione, (modellazione di sistemi interattivi) a seconda degli obiettivi del sistema informatico a cui è destinata l'interfaccia, del suo contesto d'uso e del suo utente target.</p> <p>1.2 Conoscenza e capacità di comprensione applicate: grazie ad una serie di casi di studio reali presentati a lezione ed alla partecipazione ad una reale attività di valutazione interfaccia su utenti, lo studente acquisisce specifiche capacità di applicare a casi reali le conoscenze maturate sui vari aspetti della disciplina (progettare, validare e collaudare un semplice sistema interattivo).</p> <p>Capacità trasversali/soft skills</p> <p>2.1 Autonomia di giudizio: lo studente acquisisce una capacità di valutazione critica delle diverse caratteristiche di un'interfaccia utente e di come esse possono influire positivamente o negativamente sull'efficacia dell'interfaccia in diversi contesti d'uso e per diverse categorie di utenza.</p>	<p>Sector-specific skills</p> <p>1.1 Knowledge and understanding: Students acquire specific multidisciplinary knowledge on the characteristics of users and human-computer interfaces. They also learn to evaluate and choose from various interaction techniques (modeling of interactive systems), depending on the objectives of the computer application for which the interface is intended, its context of use, and its target user.</p> <p>1.2 Applied knowledge and understanding: Through a series of case studies and participation in a real evaluation of a user interface, students acquire specific skills to apply knowledge of the discipline to the various aspects of real-world projects concerning user interfaces (design, validate and test a simple interactive system).</p> <p>Cross-sectoral skills/soft skills</p> <p>2.1. Autonomy of judgment: Students acquire the ability to critically evaluate the different features of a user interface and how they can positively or negatively affect the effectiveness of the interface in different contexts of use and for different categories of users.</p>	
--	--	--	--	--

		<p>2.2 Abilità comunicative: lo studente impara a descrivere un'interfaccia utente in modo tecnicamente corretto ed usando la terminologia appropriata. Il corso inoltre dedica alcune lezioni al tema della comunicazione interpersonale di tipo mediato attraverso strumenti informatici quali e-mail, videoconferenza e social network.</p> <p>2.3 Capacità di apprendimento: il corso fornisce le basi e gli strumenti che permettono allo studente di approfondire ed affrontare autonomamente problemi inerenti alla progettazione e valutazione di interfacce utente.</p>	<p>2.2 Communication Skills: Students learn to describe a user interface in a technically correct way and using the appropriate terminology. The course also devotes some lessons to the topic of interpersonal communication mediated through computer tools such as e-mail, videoconferencing and social networks.</p> <p>2.3 Learning Abilities: The course provides the knowledge and tools that enable the student to deepen and address autonomously issues related to the design and evaluation of user interfaces.</p>	
Internet of Things	INF/01	<p>L'obiettivo del corso è fornire allo studente le conoscenze fondamentali e le abilità pratiche relative alla progettazione ed all'implementazione di sistemi hardware/software nell'ambito dell'Internet degli oggetti (in inglese: Internet of Things, abbreviato in IoT). In particolare, partendo dalla contrapposizione fra il classico "paradigma del desktop" (che ha caratterizzato l'informatica per molti anni) ed il nuovo modello di "ubiquitous computing" che lo sta affiancando e rapidamente rimpiazzando, vengono introdotti ed analizzati gli avanzamenti tecnologici che consentono di avere a disposizione oggi reti di dispositivi, sensori e attuatori interconnessi con cui è possibile scambiare dati ed a cui è altresì possibile inviare comandi di vario genere. L'attenzione si sposta quindi dall'interazione uomo-macchina all'interazione diretta fra macchine (in inglese Machine to Machine, abbreviato in M2M), con</p>	<p>The aim of the course is to provide students with basic knowledge and practical skills about the design and implementation of hardware/software systems in the Internet of Things (IoT) scenario. Indeed, the new "ubiquitous computing" model is quickly overlapping and replacing the classic desktop paradigm (which characterized the computing world for many years), thanks to technological advances that allow the user to interact with networking devices, sensors and interconnected actuators, in order to exchange data and to remotely program systems by sending commands of various kinds. Thus, the focus shifts from human-machine interaction to machine-to-machine interaction (M2M) with the use of protocols and algorithms suitable for the transmission and the efficient manipulation of data streams.</p>	<p>CONSIGLIATI: Architettura degli Elaboratori, Sistemi Operativi e Laboratorio, Reti di Calcolatori, Tecnologie Web per il Cloud e laboratorio, Programmazione Orientata agli Oggetti e Laboratorio</p>

	<p>l'utilizzo di protocolli ed algoritmi appositi, orientati alla trasmissione ed alla manipolazione efficiente di flussi di dati. Il corso prevede delle lezioni pratiche da tenersi in laboratorio in modo da consentire allo studente la sperimentazione diretta delle nozioni apprese. Argomenti trattati:</p> <ul style="list-style-type: none"> ○ Hardware: piattaforme di prototipizzazione rapida (Arduino), single-board computer (Raspberry), sensori, tecnologie di comunicazione wireless (RFID, Bluetooth LE, Wi-Fi) ○ Protocolli di comunicazione M2M (DDS, AMQP, MQTT, JMS, REST, CoAP ecc.) ○ Algoritmi di Stream Processing ○ Problematiche di programmazione relative all'IoT: concorrenza, real-time computing, sincronizzazione, gestione degli errori ○ Framework per applicazioni IoT <p>Capacità relative alle discipline</p> <p>1.1 Conoscenza e capacità di comprensione: lo studente acquisisce conoscenze specifiche della programmazione dei sistemi IoT e delle problematiche connesse al loro funzionamento. Egli inoltre impara a valutare ed a scegliere varie tecniche ed algoritmi, a seconda dell'ambito di implementazione e delle specifiche progettuali.</p> <p>1.2 Conoscenza e capacità di comprensione applicate: grazie all'attività di laboratorio, lo studente acquisisce specifiche capacità di applicare a casi reali le conoscenze maturate sui vari aspetti dei sistemi IoT.</p> <p>Capacità trasversali/soft skills</p> <p>2.1 Autonomia di giudizio: lo studente acquisisce una capacità di valutazione critica sulle</p>	<p>Practical lessons will be held in the laboratory, in order to allow students to experiment with several IoT-technologies and tools.</p> <p>Main topics:</p> <ul style="list-style-type: none"> ○ Hardware: rapid prototyping platforms (Arduino), single-board computers (Raspberry), sensors, wireless communication technologies (RFID, Bluetooth LE, Wi-Fi) ○ M2M communication protocols (DDS, AMQP, MQTT, JMS, REST, CoAP etc.) ○ Stream Processing Algorithms ○ Programming issues related to IoT: concurrency, real-time computing, synchronization, error handling ○ Frameworks for building IoT applications <p>Sector-specific skills</p> <p>1.1 Knowledge and understanding: the student will acquire specific knowledge about programming IoT systems and about the issues related to their operation. She/He will also learn how to evaluate and how to choose among several techniques and algorithms, depending on the purpose of the implementation and on the design specifications.</p> <p>1.2 Applied knowledge and understanding: thanks to the lab activities, the student will acquire specific skills in the application of the knowledge on the various aspects of IoT systems to real case studies.</p> <p>Cross-sectoral skills/soft skills</p> <p>2.1 Making Judgements: the student will acquire a critical appraisal skill on technologies, algorithms and programming techniques that can positively or negatively</p>	
--	---	---	--

		<p>tecnologie, gli algoritmi e le tecniche di programmazione che possono influire positivamente o negativamente sulla corretta ed efficace implementazione di un sistema IoT in diversi contesti applicativi.</p> <p>2.2 Abilità comunicative: lo studente impara a descrivere in modo tecnicamente corretto ed usando la terminologia appropriata un sistema IoT.</p> <p>2.3 Capacità di apprendimento: lo studente impara ad essere autonomo nell'espandere le proprie conoscenze sui sistemi IoT oltre le nozioni e gli esempi appresi ed analizzati durante il corso, scoprendo e sfruttando anche le connessioni con altri corsi riguardanti i sistemi operativi, le reti e la programmazione.</p>	<p>affect the correct and effective implementation of an IoT system in different application contexts.</p> <p>2.2 Communication Skills: the student will learn how to describe an IoT system using a technically correct and appropriate terminology.</p> <p>2.3 Learning Skills: the student will learn how to be autonomous in expanding his knowledge of IoT systems beyond the notions and examples learned and analyzed during the course, discovering and exploiting connections with other courses such as operating systems, computer networks, and programming.</p>	
Machine Learning for Big Data	ING-INF/05	<p>Il corso si propone di presentare nel dettaglio metodi e algoritmi di apprendimento automatico (in inglese: Machine Learning) discutendone le proprietà e i criteri di applicabilità. Il Machine Learning è il "campo di studio che dà ai computer l'abilità di apprendere (a realizzare un compito) senza essere esplicitamente programmati a farlo" (Arthur Samuel, 1950). Recentemente grazie alla crescente quantità di informazioni digitalmente disponibili il Machine Learning è diventato un importante settore dell'informatica con moltissime applicazioni in svariati settori tra cui la medicina, l'analisi automatica del testo, i sistemi di guida autonoma ecc. L'obiettivo principale del corso è quindi fornire allo studente le conoscenze fondamentali per risolvere problemi di apprendimento automatico attraverso una corretta formulazione del problema, una scelta critica dell'algoritmo di apprendimento e</p>	<p>The course aims to present machine learning algorithms and techniques discussing in deep the main features and applicability criteria. Machine Learning is "field of study that gives computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1950). Recently, thanks to the increasing amount of data digitally available, Machine Learning has become an important field of computer science with several applications in different scientific areas such as medicine, automatic text analysis, autonomous driving systems, and so on. The course aims to enable students to acquire basic knowledge to solve machine-learning tasks through a proper formulation of the problem, a critical choice of the learning algorithm and an experimental analysis of the obtained results. The course includes laboratory activities so that students can</p>	<p>CONSIGLIATE: Analisi matematica, Programmazione e laboratorio</p>

		<p>un'analisi sperimentale per valutare i risultati ottenuti. Per questo motivo il corso prevede lezioni in laboratorio in modo da consentire allo studente di sperimentare direttamente le nozioni apprese su applicazioni reali.</p> <p>Capacità relative alle discipline</p> <p>1.1 Conoscenza e capacità di comprensione: durante il corso lo studente acquisisce conoscenze teoriche dei principali metodi di apprendimento automatico, impara inoltre le procedure formali per la valutazione e l'analisi dei risultati ottenuti.</p> <p><i>Conoscenza e capacità di comprensione applicate:</i> l'attività di laboratorio permette allo studente di consolidare le conoscenze teoriche presentate durante le lezioni frontali del docente attraverso il loro utilizzo in casi reali e risolvere problemi di apprendimento automatico attraverso modellazione, implementazione di tecniche di machine learning, analisi sperimentale dei risultati.</p> <p>Capacità trasversali/soft skills</p> <p>2.1 Autonomia di giudizio: lo studente acquisisce capacità teoriche e pratiche che gli consentono di sviluppare algoritmi di Machine Learning e analizzarne criticamente i risultati prodotti.</p> <p>2.2 Abilità comunicative: lo studente impara la terminologia appropriata ed è in grado di descrivere le principali caratteristiche degli algoritmi presentati durante il corso.</p> <p>2.3 Capacità di apprendimento: il corso mira a fornire allo studente gli strumenti necessari per affrontare e risolvere autonomamente problemi inerenti all'apprendimento automatico.</p>	<p>directly test on real applications concepts learned in class.</p> <p>Sector-specific skills</p> <p>1.1 Knowledge and understanding: During the course, the student learns basic knowledge of the main methods of machine learning. He also learns procedures for evaluating and analysing the obtained results.</p> <p>1.2 Applied knowledge and understanding: the laboratory activity allows the student to consolidate the theoretical knowledge, presented during the lectures, through their use in real application cases.</p> <p>Cross-sectoral skills/soft skills</p> <p>2.1 Making judgements: the student acquires theoretical and practical skills that allows him to develop machine-learning algorithms and to analyse critically the obtained results and solve machine learning problems through modeling, implementation of machine learning techniques, experimental analysis of results.</p> <p>2.2 Communication skills: the student learns appropriate terminology and he is able to present the main features of machine learning algorithms tested in the course.</p> <p>2.3 Learning skills: the course aims to provide students with the basic knowledge needed to understand and solve automatically machine-learning problems.</p>	
Programmazione e laboratorio	INF/01	Il corso introduce i fondamenti scientifici e metodologici della programmazione e si propone di mettere lo studente in condizione di cogliere pienamente la natura di questa attività,	The course introduces the scientific and methodological foundations of computer programming, in order for students to understand the nature of programming and to	

		<p>stimolando inoltre un atteggiamento critico nei confronti degli strumenti utilizzati.</p> <p>L'approccio seguito è di tipo functional-first (IEEE-CS/ACM Computing Curricula 2001); il percorso didattico si articola attorno ai principali processi di astrazione che consentono di gestire la complessità dei problemi affrontati in ambito informatico: l'astrazione procedurale, l'astrazione sui dati e l'astrazione sullo stato.</p> <p>Al termine del corso lo studente dovrebbe aver acquisito le competenze di base e le capacità operative necessarie al fine di progettare, organizzare e formalizzare programmi di piccole dimensioni, sviluppati secondo i paradigmi funzionale, imperativo e object-oriented; dovrebbe inoltre essere in grado di analizzare la struttura logica di un programma al fine di verificarne la correttezza in relazione alle specifiche.</p> <p>Le attività di Laboratorio consistono nel progetto, nello sviluppo e nella sperimentazione di programmi di piccole dimensioni, e hanno lo scopo di stimolare le capacità organizzazione e lavoro autonomo da parte degli studenti.</p> <p>Programma Sintetico</p> <p>I. Astrazione procedurale:</p> <ul style="list-style-type: none"> ○ Procedure come astrazione di espressioni; ○ Ricorsione, ricorsione ad albero e ricorsione dicoda; ○ Argomenti e valori procedurali; ○ Correttezza di programmi ricorsivi. <p>II. Astrazione sui dati:</p> <ul style="list-style-type: none"> ○ Tipi semplici e tipi strutturati; ○ Protocollo relativo a un dato astratto; ○ Utilizzo e realizzazione di un dato astratto. 	<p>develop a critical attitude toward the related tools.</p> <p>The approach is functional-first (IEEE-CS/ACM Computing Curricula 2001); the course is organized around the main forms of abstraction, which are exploited to deal with problem complexity: procedural abstraction, data abstraction, and abstraction of state.</p> <p>At the end of the course, the student should have acquired the basic knowledge and skills to design organize and write small-scale programs, developed according to the functional, imperative and object-oriented paradigms; moreover, he/she should be able to analyze the logic of a program to verify its correctness with respect to its formal specifications.</p> <p>The laboratory sessions are about design, development and experimentation of small-scale programs; they are meant to stimulate students' organization skills as well as their ability to work autonomously.</p> <p>Short Syllabus</p> <p>I. Procedural abstraction:</p> <ul style="list-style-type: none"> ○ Expressions and procedures; ○ Recursion, tree recursion and tail recursion; ○ Higher order functions; ○ Correctness of recursive programs. <p>II. Data abstraction:</p> <ul style="list-style-type: none"> ○ Simple and structured types; ○ Operations on abstract data; ○ Use vs. implementation of abstract data. <p>III. Abstraction of state:</p> <ul style="list-style-type: none"> ○ Concept of state and imperative paradigm; 	
--	--	--	---	--

	<p>III. Astrazione sullo stato:</p> <ul style="list-style-type: none"> ○ Concetto di stato e paradigma imperativo; ○ Tecniche di memoization (top-down) e programmazione dinamica (bottom-up); ○ Classi, oggetti e incapsulamento dello stato; ○ Programmazione orientata agli oggetti; ○ Correttezza di programmi iterativi. <p>Capacità relative alle discipline Lo/la studente/essa dovrà:</p> <p>1.1. Conoscenza e capacità di comprensione</p> <ul style="list-style-type: none"> ○ Acquisire le basi della programmazione funzionale e imperativa; ○ Acquisire i primi rudimenti della programmazione object-oriented; ○ Conoscere sintassi e semantica dei principali costrutti e delle principali strutture dati nei linguaggi Scheme e Java; <p>Conoscere le relazioni fra ricorsione e iterazione; Conoscere le relazioni fra i concetti di classe e oggetto.</p> <p>1.2 Capacità di applicare conoscenza e comprensione</p> <ul style="list-style-type: none"> ○ Saper codificare semplici algoritmi ricorsivi e imperativi; ○ Saper analizzare la logica di un semplice programma ricorsivo o imperativo per verificarne la correttezza rispetto alle specifiche e per stimarne le prestazioni; ○ Saper riconoscere quando è appropriato e saper applicare tecniche di memoization e programmazione dinamica per migliorare le prestazioni di programmi ricorsivi; ○ Saper realizzare una classe rispettando il protocollo specificato. 	<ul style="list-style-type: none"> ○ Memoization (top-down) and dynamic programming (bottom-up); ○ Classes, objects and information-hiding; ○ Object-oriented programming; ○ Correctness of iterative programs. <p>Sector-specific skills</p> <p>1.1. Knowledge and understanding</p> <ul style="list-style-type: none"> ○ Basic knowledge of functional and imperative programming; ○ First rudiments of object-oriented programming; ○ Basic knowledge of the syntax and semantics of the main program and data structures in Scheme and Java; ○ Knowledge of the relationships between recursion and iteration; ○ Knowledge of the relationships between the concepts of class and object. <p>1.2 Applying knowledge and understanding</p> <ul style="list-style-type: none"> ○ Being able to code simple recursive and imperative algorithms; ○ Being able to analyze the logic of a simple recursive or imperative program in order to verify its correctness with respect to the given specifications and to estimate its performances; ○ Being able to apply the memoization and dynamic programming techniques to improve the performances of recursive programs; ○ Being able to implement a class with a given interface. 	
--	--	---	--

		<p>Capacità trasversali / soft skills Lo/la studente/essa dovrà:</p> <p>2.1 Autonomia di giudizio</p> <ul style="list-style-type: none"> ○ Saper analizzare i problemi al fine di identificare gli aspetti che si prestano ad essere affrontati attraverso programmi, esprimere le specifiche in maniera precisa e scegliere gli strumenti di programmazione adatti allo scopo; ○ Saper applicare le proprie conoscenze per capire i processi di elaborazione dell'informazione. <p>2.2 Abilità comunicative.</p> <ul style="list-style-type: none"> ○ Essere in grado di confrontarsi con i pari al fine di progettare o migliorare la soluzione algoritmica di un problema. <p>2.3 Capacità di apprendimento</p> <ul style="list-style-type: none"> ○ Essere in grado di attuare sperimentazioni sistematiche di un programma per verificarne la correttezza e per stimarne le prestazioni; ○ Essere in grado di orientarsi nell'ambito dei linguaggi di programmazione e di apprendere autonomamente nuovi linguaggi. 	<p>Cross-sectoral skills/soft skills</p> <p>2.1 Making judgements</p> <ul style="list-style-type: none"> ○ Being able to analyze problems in order to identify what can be achieved by a program, to formalize appropriate specifications and to choose appropriate programming tools; ○ Being able to understand information-processing processes. <p>2.2 Communication skills.</p> <ul style="list-style-type: none"> ○ Being able to work together with peers in order to design or improve the algorithmic solution of a problem. <p>2.3 Learning skills</p> <ul style="list-style-type: none"> ○ Being able to plan experiments on a program to test its correctness and to estimate its performances; ○ Being able to learn autonomously new programming languages. 	
--	--	---	--	--

<p>Programmazione orientata agli oggetti e laboratorio</p>	<p>ING-INF/05</p>	<p>Scopo del corso è la comprensione profonda dei principi della programmazione orientata agli oggetti e di come questi influenzano la qualità del codice e la sua manutenibilità.</p> <p>Argomenti</p> <p>Il corso prevede di formare lo studente relativamente a questi argomenti: La programmazione in the large vs in the small: il ruolo della qualità nello sviluppo del software. Tipi di astrazione: procedurale, parametrica, per specifica. L'approccio "clean code". Programmi come funzioni totali o parziali; le eccezioni; la programmazione difensiva e quella assertiva. I tipi di dati astratti. Gerarchie di tipi; le interfacce; le classi astratte; il principio dell'inversione della dipendenza. Il principio di Liskov. Tipi generici. Classi anonime. Cenni alla progettazione orientata agli oggetti: la tecnica CRC, il principio della singola responsabilità, alcuni pattern di design (factory, builder, observer, decorator).</p>	<p>The aim of the course is to ground the principles of object orientation and how they are related to quality of the code and its maintainability.</p> <p>Topics</p> <p>The following topics are covered: programming in the large. Abstractions in procedures, parameterization, by specification. The "clean code" approach. Defensive vs assertive programming and the role of exceptions. Type hierarchies, Liskov principle, dependency inversion principle. Generics, anonymous classes. OO design, CRC technique, design patterns (factory, builder, observer, decorator).</p>	<p>VINCOLANTE: Programmazione e laboratorio</p>
--	-------------------	---	--	--

	<p>Capacità relative alla disciplina</p> <p>1.1 Conoscenza e capacità di comprensione conoscere i concetti di base della programmazione orientata agli oggetti, conoscere alcuni pattern di design orientato agli oggetti, conoscere i 7 principi SOLID di progettazione (singola responsabilità, Liskov, inversione dipendenza, open-closed, interface segregation) e di programmazione (clean code).</p> <p>1.2 Capacità di applicare conoscenza e comprensione saper utilizzare Java per la programmazione orientata agli oggetti saper applicare alcuni pattern di design. Questi obiettivi saranno supportati da attività guidata di laboratorio.</p> <p>Capacità trasversali</p> <p>2.1 Autonomia di giudizio saper valutare la qualità del codice prodotto</p> <p>2.2 Abilità comunicative essere in grado di descrivere e motivare le scelte fatte</p> <p>2.3 Capacità di apprendimento essere in grado di approfondire autonomamente gli argomenti toccati a lezione</p>	<p>Sector specific topics</p> <p>1.1 Knowledge and understanding Basic concepts of object-oriented programming; know some object-oriented design patterns; SOLID design principles (single responsibility, Liskov, dependency inversion, open-closed, interfacesegregation); the clean code approach.</p> <p>1.2 Applying knowledge and understanding Being able to use Java and common IDEs; being able to apply design patterns. These objectives will be supported by laboratory work.</p> <p>Cross sectoral skills</p> <p>2.1 Making judgements Being able to assess the quality of the code being produced.</p> <p>2.2 Communication skills being able to describe technical choices and to argue the validity of design decisions.</p> <p>2.3 Learning skills Being able to deepen one’s own understanding of the topics covered during lectures.</p>	
--	---	---	--

Reti di calcolatori	INF/01	<p>Obiettivi Scopo del corso è di introdurre i principi fondamentali delle moderne reti di calcolatori in modo da fornire allo studente le conoscenze necessarie per l'analisi e la progettazione di una rete di calcolatori. In particolare, verranno presentate le caratteristiche generali delle reti, dei segnali a livello fisico, i modelli ISO/OSI e TCP/IP, e le problematiche affrontate dai principali protocolli ad ogni livello di questi modelli. Inoltre, vengono presentate le problematiche di sicurezza delle comunicazioni tra calcolatori; a questo scopo, verranno introdotte anche alcune nozioni di crittografia.</p> <p>Capacità relative alle discipline 1.1 Conoscenza e capacità di comprensione: lo studente acquisisce le conoscenze e capacità necessarie per analizzare e progettare una rete di calcolatori, e comprendere eventuali problematiche ad esse connesse. Le conoscenze coprono i molteplici aspetti delle reti: architettura di rete, modello client/server, programmazione con socket, accesso alla rete condivisa cablata e wireless, principi di internetworking, protocolli di trasporto, controllo della congestione, codifica dei dati multimediali, aspetti di sicurezza e crittografia. 1.2 Conoscenza e capacità di comprensione applicate: grazie ad una serie di esempi presentati a lezione e esercizi mirati alla risoluzione di casi pratici, lo</p>	<p>Aims The aim of the course is to introduce the basic principles of modern computer networks in order to provide the student with the knowledge needed for the analysis and design of a computer network. Specifically, the course presents the general characteristics of networks, physical-level signals, ISO/OSI and TCP/IP models, and the issues addressed by the major protocols at each level of these models. In addition, the security issues of computer communications are addressed; to this purpose, some notions of cryptography will also be introduced.</p> <p>Skills related to the disciplines 1.1 Knowledge and understanding: The student acquires the knowledge and skills needed to analyze and design a computer network, and to understand related issues. This knowledge covers the many aspects of networks: network architecture, client/server model, socket programming, wired and wireless shared network access, internetworking principles, transport protocols, congestion control, encoding of multimedia data, security aspects and encryption. 1.2 Applied knowledge and understanding: By means of a series of examples presented in class and exercises aimed at solving practical cases, the student will learn how to apply to</p>	<p>VINCOLANTE: Sistemi operativi e laboratorio</p> <p>CONSIGLIATA: Programmazione e laboratorio, statistica e laboratorio</p>
---------------------	--------	--	--	---

	<p>studente acquisisce specifiche capacità di applicare a casi reali le conoscenze maturate sui vari aspetti della disciplina. In particolare lo studente sarà in grado di:</p> <ol style="list-style-type: none"> 1) Svolgere compiti di analisi di una rete di calcolatori, in sede di diagnostica e di progettazione; 2) Svolgere compiti di gestione amministrativa e della sicurezza di una semplice rete locale. <p>Capacità trasversali/soft skills</p> <p>2.1 Autonomia di giudizio: lo studente acquisisce una capacità di valutazione critica delle diverse caratteristiche di una rete di calcolatori e di come esse possono influire sul funzionamento di un sistema informativo aziendale o di una applicazione distribuita, in diversi contesti d'uso e per diverse categorie di utenza.</p> <p>2.2 Abilità comunicative: lo studente impara a descrivere e progettare una rete di calcolatori in modo tecnicamente corretto ed usando la terminologia appropriata. Questo permette allo studente di inserirsi rapidamente in contesti professionali in cui questa terminologia tecnica è essenziale.</p> <p>2.3 Capacità di apprendimento: il corso fornisce le basi e gli strumenti che permettono allo studente di approfondire ed affrontare autonomamente problemi inerenti all'analisi, progettazione e realizzazione di una rete di calcolatori.</p>	<p>real-life cases the knowledge about the various aspects of the discipline.</p> <p>In particular, the student will be able to:</p> <ol style="list-style-type: none"> 1) Carry out tasks of analysis of a computer network, in the diagnostic and design phase; 2) Carry out tasks of administrative and security management of a simple local network <p>Cross skills / soft skills</p> <p>2.1. <i>Autonomy of judgment:</i> the student acquires the capability of assessing the different characteristics of a computer network and how they can affect the functioning of a business information system or distributed application in different contexts and for different categories of users.</p> <p>2.2 <i>Communicative Skills:</i> the student learns how to describe and design a computer network in a technically correct manner and using the appropriate terminology. This allows the student to be acquainted in professional contexts where this technical terminology is essential.</p> <p>2.3 <i>Learning Skills:</i> The course provides the basics and tools that enable the student to deepen and face independently issues related to the analysis, design and realization of a computer network.</p>	
--	---	---	--

<p>Sistemi operativi e laboratorio</p>	<p>ING-INF/05</p>	<p>Capacità relative alla disciplina</p> <p>1.1 <i>Conoscenza e capacità di comprensione:</i> lo studente acquisisce conoscenze specifiche sull'architettura, sul funzionamento e sui principi di progettazione di un moderno sistema operativo. Egli inoltre impara a valutare ed a scegliere l'adozione di un sistema operativo a seconda dell'ambito di utilizzo e delle caratteristiche dell'hardware a disposizione.</p> <p>1.2 <i>Capacità di applicare conoscenza e comprensione:</i> attraverso le attività pratiche in laboratorio lo studente acquisirà capacità nell'analizzare e risolvere i problemi tipici relativi alla gestione, amministrativa e della sicurezza di un sistema operativo, alla sincronizzazione, comunicazione tra processi.</p>	<p>Sector-specific skills</p> <p>1.1 <i>Knowledge and understanding:</i> the student acquires specific knowledge on the architecture, functioning and design principles of a modern operating system. He/she also learns to evaluate and choose the adoption of an operating system according to the context of use and the characteristics of the hardware at hand.</p> <p>1.2 <i>Applying knowledge and understanding:</i> through practical activities, in laboratory, the student will acquire skills in analyzing and solving typical problems related to management, administrative and security of an operating system, to synchronization and communication between processes.</p>	<p>CONSIGLIATA: Architettura degli elaboratori, Programmazione e laboratorio</p>
--	-------------------	--	---	--

		<p>Capacità trasversali/soft skills</p> <p>2.1 Autonomia di giudizio: lo studente acquisisce capacità di valutazione critica delle tecnologie, algoritmi e soluzioni impiegate nella realizzazione di un sistema operativo, anche in relazione al contesto applicativo (sistemi tradizionali, real-time, embedded, mobili, ecc.).</p> <p>2.2 Abilità comunicative: lo studente impara a descrivere, adottando una terminologia appropriata, le principali funzionalità di un moderno sistema operativo, le componenti ad esse preposte, le problematiche e le tecniche risolutive ad esse associate.</p> <p>2.3 Capacità di apprendimento: lo studente impara ad espandere autonomamente le proprie competenze sui sistemi operativi, scoprendo e sfruttando anche connessioni con altri insegnamenti riguardanti l'architettura dei calcolatori, le reti ed i sistemi distribuiti.</p>	<p>Cross-sectoral skills/soft skills</p> <p>2.1 Making judgements: the student acquires critical assessment skills of the technologies, algorithms, and solutions used in the design of an operating system, also in relation to the application context (traditional, real-time, embedded, mobile systems, etc.).</p> <p>2.2 Communication skills: the student learns to describe, by adopting an appropriate terminology, the main functionalities of a modern operating system, the corresponding system components, the problems and the solving techniques associated with them.</p> <p>3.2 Learning skills: the student learns to independently extend his/her skills, also discovering and exploiting connections with other notion concerning computer architecture, networks, and distributed systems.</p>	
Social Computing	ING-INF/05	L'obiettivo del corso è fornire allo studente le conoscenze fondamentali e le abilità pratiche relative al mondo dell'informatica sociale. Vengono trattati sia i cosiddetti Social media (Facebook, Twitter, ecc.) sia il fenomeno del Crowdsourcing: nel primo caso il comportamento sociale viene supportato da sistemi computazionali; nel secondo caso i sistemi computazionali vengono supportati dal comportamento sociale.	The aim of the course is to provide to the student the foundational knowledge and the practical skills concerning the area of social informatics. The course will discuss both the so-called Social Media (Facebook, Twitter, etc.) and the Crowdsourcing phenomenon. In the first case, social behavior is supported by computational systems; in the second case, computational systems are supported by social behavior.	VINCOLANTE: Tecnologie web

		<p>La trattazione riguarda sia gli aspetti concettuali, sia i fondamenti teorici, sia le applicazioni pratiche. Il corso è diviso nelle quattro parti seguenti.</p> <p><i>1. Introduzione</i></p> <ul style="list-style-type: none"> ○ Esempi di social media, importanza per la scienza dei dati (fonte di dati, di utenti, luogo in cui accadono fenomeni interessanti) ○ Esempi di Crowdsourcing di successo e non <p><i>2. I Social media</i></p> <ul style="list-style-type: none"> ○ Concetti: Definizioni. Esempi. Classificazione (generalisti, verticali, privati) ○ Fondamenti: Cenni storici (analisi reti sociali, scienza delle reti). Cenni di Scienza delle reti ○ Applicazioni: API per accesso ai dati dei social (casi di studio: Twitter o Facebook) <p><i>3. Il Crowdsourcing</i></p> <ul style="list-style-type: none"> ○ Concetti: Definizioni, Esempi. Intelligenza collettiva ○ Fondamenti: Il caso generale della Human computation. Caratteristiche necessarie per il successo. Cenni di Computabilità 	<p>The course will deal with conceptual topics, theoretical foundations, and practical applications. The course is divided into the following four parts:</p> <p><i>1. Introduction</i></p> <ul style="list-style-type: none"> ○ Examples of social media, relevance to data science (socials are a source of data and users, and a ground where interesting phenomena happen) ○ Examples of Crowdsourcing. Success stories and failures. <p><i>2. Social media</i></p> <ul style="list-style-type: none"> ○ Concepts: Definition. Examples. Classification (generalist, verticals, private). ○ Foundations: Historical background (social network analysis, network science). Elementary network science. ○ Applications. APIs to access data from socials (case studies: Twitter or Facebook or Telegram) <p><i>3. Crowdsourcing</i></p> <ul style="list-style-type: none"> ○ Concepts: Definitions, Examples. Collective intelligence ○ Foundations: The general case of Human computation. Characteristics needed for successful Crowdsourcing. Computability (brief account) 	<p>per il Cloud e laboratorio.</p> <p>CONSIGLIATE: Fondamenti di scienza dei dati e laboratorio, Statistica e laboratorio</p>
--	--	---	--	---

	<ul style="list-style-type: none"> ○ Applicazioni: uso di una piattaforma di crowdsourcing (Crowdfower o Amazon's Mechanical Turk) per progettazione ed esecuzione di esperimenti. Analisi dei risultati <p>4. Approfondimenti</p> <ul style="list-style-type: none"> ○ Aspetti etici, morali, legali. ○ Aspetti economici (SEO, modelli di business, crowdfunding) ○ Programmazione social-aware (Sistemi multiagente, society design; algoritmi genetici; map/reduce) ○ Sistemi ibridi <p>Capacità relative alle discipline</p> <p>1.1 Conoscenza e capacità di comprensione: lo studente acquisisce conoscenze specifiche del Social computing e delle problematiche connesse. Egli inoltre impara ad analizzare, comprendere e valutare i fenomeni principali del mondo social, per quanto concerne sia i social media sia il crowdsourcing e la human computation.</p> <p>1.2 Conoscenza e capacità di comprensione applicate: lo studente acquisisce conoscenze sull'uso delle API per l'accesso ai dati dei social media e impara a scegliere fra le varie alternative di progetto per esperimenti di raccolta dati tramite crowdsourcing.</p> <p>Capacità trasversali/soft skills</p> <p>2.1 Autonomia di giudizio: lo studente acquisisce una capacità di valutazione critica sulle tecnologie, gli algoritmi e le tecniche di programmazione che possono influire positivamente o negativamente sull'uso corretto ed efficace dei social e del crowdsourcing in diversi contesti applicativi.</p>	<ul style="list-style-type: none"> ○ Applications: usage of a Crowdsourcing platform (Crowdfower or Amazon's Mechanical Turk) to design and run experiments. Analysis of results. <p>4. Case studies and specific issues</p> <ul style="list-style-type: none"> ○ Ethical, moral, legal aspects ○ Economic aspects (SEO, business models, crowdfunding) ○ Social-aware programming (multiagent systems, society design; genetic algorithms; map/reduce) ○ Hybrid systems <p>Sector-specific skills</p> <p>1.1 Knowledge and understanding: the student acquires specific knowledge on Social computing and related issues. The student will also learn to analyze, understand, and evaluate the main phenomena of the social computing world, with respect to social media, crowdsourcing, and human computation.</p> <p>1.2 Applying knowledge and understanding: the student acquires knowledge on API usage to access social media data, as well as learns to choose among the various design choices for data gathering experiments by mean of crowdsourcing.</p> <p>Cross-sectoral skills/soft skills</p> <p>2.1 Making judgements: the student acquires critical evaluation skills on technologies, algorithms, and programming techniques that can affect, either positively or negatively, the correct and effective usage of socials and crowdsourcing in different application contexts.</p>	
--	--	---	--

		<p>2.2 Abilità comunicative: lo studente impara a descrivere in modo tecnicamente corretto ed usando la terminologia appropriata lo stato dell'arte del Social computing. Impara inoltre a comunicare in modo corretto usando i Social.</p> <p>2.3 Capacità di apprendimento: lo studente impara ad essere autonomo nell'espandere le proprie conoscenze sul Social computing oltre le nozioni e gli esempi appresi ed analizzati durante il corso, scoprendo e sfruttando anche le connessioni con altri corsi riguardanti le tecnologie web per il cloud, la scienza dei dati, l'internet of things, il machine learning e la programmazione.</p>	<p>2.2 Communication skills: the student learns to describe in a technically appropriate way, and using the correct terminology, the state of the art of Social computing. The student also learns to communicate in an appropriate way using the social media.</p> <p>2.3 Learning skills: the student learns to be autonomous in expanding his/her own knowledge on Social computing beyond the notions and examples learnt and analyzed during the course. The student is also able to discover and exploit connections with other courses about web / cloud technologies, data science, internet of things, machine learning, and programming.</p>	
<p>Statistica e laboratorio Statistics and laboratory</p>	<p>SECS-S/01</p>	<p>Il corso si pone come obiettivo l'introduzione dei concetti di base dell'analisi dei dati, con particolare riferimento alle procedure proprie della statistica descrittiva e della statistica inferenziale. Le nozioni saranno introdotte considerando anche contesti applicativi legati all'ambito dell'informatica e della "scienza dei dati", per fornire agli studenti competenze utilizzabili per la propria attività professionale.</p> <p>Programma del corso: 1) Statistica descrittiva: caratteri statistici; serie, seriazioni e distribuzioni di frequenza; indici di posizione e di variabilità; analisi bivariate. 2) Probabilità: probabilità elementare; variabili casuali; modelli probabilistici; variabili casuali multivariate e convergenza; campionamento. 3) Inferenza statistica: stima puntuale; intervalli di confidenza; verifica di ipotesi.</p> <p>Capacità relative alle discipline</p>	<p>The course introduces students to the fundamental concepts of descriptive and inferential statistics. These notions will be presented focusing also on applications, with particular regard to data science and computer science applications.</p> <p>Course contents: 1) Descriptive statistics: variables; frequency distributions; location and variability indicators; bivariate analysis. 2) Probability: basic concepts; random variables; probabilistic models; random vectors and convergence concepts; sampling theory. 3) Inference: point estimation; confidence intervals; hypothesis testing.</p> <p>Sector-specific skills</p> <p>The aim of the course is to introduce the fundamental concepts of descriptive and inferential statistics, as basic tools for research and data analysis. The students are expected to acquire the fundamental skills in order to</p>	<p>CONSIGLIATA: Analisi matematica</p>

		<p>Il corso ha l'obiettivo di introdurre i concetti fondamentali della statistica descrittiva e della statistica inferenziale, come strumenti di base per la ricerca e l'analisi dei dati. Si prevede che lo studente, alla fine del corso, sia in grado di predisporre ed eseguire autonomamente analisi di dati reali basate su semplici tecniche statistiche.</p> <p>1.1 Conoscenza e capacità di comprensione Conoscenza e comprensione della statistica descrittiva e delle principali procedure per sintetizzare i dati, del concetto di incertezza e degli elementi di base della teoria della probabilità, dei concetti fondamentali della teoria dei campioni e della statistica inferenziale, delle procedure per analizzare le relazioni tra variabili e degli elementi costitutivi della verifica di ipotesi.</p> <p>1.2 Capacità di applicare conoscenza e comprensione Comprensione dei metodi statistici come strumenti di ricerca utili nelle scienze applicate e capacità di usare la statistica descrittiva e inferenziale per sintetizzare informazioni, per analizzare e interpretare relazioni tra variabili e per i test di ipotesi.</p> <p>Capacità trasversali/soft skills</p> <p>2.1 Autonomia di giudizio Autonomia di giudizio nella scelta dei modelli e dei metodi statistici più appropriati per analizzare uno specifico dataset e nell'interpretazione dei risultati sperimentali.</p> <p>2.2 Abilità comunicative Abilità comunicative nel presentare in modo convincente e corretto un'analisi statistica,</p>	<p>perform autonomously the analysis of real datasets, involving the use of basic statistical procedures.</p> <p>1.1 Knowledge and understanding Knowledge and understanding of descriptive statistics and of how to summarize data, of the notion of uncertainty and of the basics in probability theory and sampling theory, of the fundamental concepts of inferential statistics, of the procedures for studying the relationships between variables and of the basic elements of hypothesis testing.</p> <p>1.2 Applying knowledge and understanding Understanding of statistical methods as useful instruments for research in data science and computer science applications and ability to use descriptive and inferential statistics in order to summarize information, to analyze and interpret relationships between variables and to test hypotheses.</p> <p>Cross-sectoral skills/soft skills</p> <p>2.1 Making judgements Making judgements on the appropriate statistical models and methods to be used for analyzing a specific dataset and on the interpretation of the experimental results.</p> <p>2.2 Communication skills Communication skills in order to present a statistical analysis, including both the methodology and the final conclusions, in a consistent and convincing way.</p>	
--	--	---	--	--

		<p>motivando i risultati ottenuti e giustificando la metodologia adottata.</p> <p>2.3 Capacità di apprendimento permanente Capacità di apprendimento utilizzando strumenti utili per riuscire a capire autonomamente i contenuti di un report statistico e per acquisire tecniche statistiche più avanzate.</p>	<p>2.3 Learning skills Learning skills based on the prerequisites that are required for understanding autonomously a statistical report and for applying more advanced statistical techniques.</p>	
<p>Tecnologie Web e Laboratorio Web Technologies and Laboratory</p>	<p>ING-INF/05</p>	<p>Il corso si propone di fornire agli studenti una panoramica generale, corredata da esercizi di laboratorio, del World-Wide Web. In particolare, vengono introdotti i principi fondamentali del Web e i criteri di progettazione che ne derivano; la infrastruttura di rete su cui il servizio web viene erogato, i linguaggi di strutturazione e presentazione (XML, HTML5, CSS3); i concetti di base e le tecnologie abilitanti del web dinamico e del web semantico.</p> <p>Alla fine del corso lo/la studente/essa dovrà:</p> <p>Capacità relative alla disciplina</p> <p>1.1 Conoscenza e capacità di comprensione Conoscere la terminologia e le concettualizzazioni di base per descrivere, analizzare e valutare tecnologie web. Conoscere i principi, i metodi, le tecniche e gli strumenti per lo sviluppo di applicazioni web.</p> <p>1.2 Conoscenze e capacità di comprensione applicate Saper applicare le conoscenze concettuali e metodologiche di cui sopra per realizzare siti e portali web di tipo statico.</p> <p>Capacità trasversali/soft skills</p>	<p>The course is aimed at introducing the students to the basics of the World Wide Web. Main topics are: the technical infrastructure (Internet); fundamental principles of web design and development; technologies supporting the dynamic and semantic web; languages for organization and presentation of information (i.e., XML, HTML5, CSS3). A laboratory activity is aimed at the application of conceptual knowledge to concrete development of web sites.</p> <p>At the end of the course the student should be able:</p> <p>Sector-specific skills</p> <p>1.1 Knowledge and understanding To know the main conceptualizations and the technical vocabulary for the analysis, description and evaluation of web technologies. To know the basic principles, methods and tools for the development of a web application.</p> <p>1.2 Applying knowledge and understanding To effectively exploit the aforementioned knowledge and methods for developing (static) web sites or portals.</p> <p>Cross-sector skills/ soft skills</p>	

		<p>2.1 Autonomia di giudizio Saper analizzare, confrontare e valutare in maniera critica due o più siti web da diversi punti di vista (tecnico, strutturale, funzionale, comunicativo, sociale/culturale) e a diversi livelli di aggregazione (singole pagine, aree, intero sito) e di astrazione (aspetti quantitativi e qualitativi).</p> <p>2.2 Abilità comunicative Saper descrivere le caratteristiche fondamentali di un sito web. Saper spiegare e argomentare le decisioni tecniche e di progetto che sono alla base di una applicazione web.</p> <p>2.3 Capacità di apprendimento Saper identificare problemi e proporre soluzioni. Essere capaci di generalizzare (specializzare) concetti, modelli e metodi. Saper stabilire analogie tra concettualizzazioni diverse e saper trasferire concetti e metodi tra domini differenti. Essere capaci di riflettere su scelte e decisioni (meta-riflessione).</p>	<p>2.1 Making judgements To compare, evaluate and criticize two or more web applications from different perspectives (e.g., technical, structural, functional, communicative and socio-cultural points of view), at different aggregation levels (i.e., single pages, areas and the whole site) and at different abstraction levels (e.g., focusing on quantitative and qualitative features).</p> <p>2.2 Communication skills To describe the main characteristics of a web site. To be able to explain and motivate the technical and design decisions that are at the base of a web application.</p> <p>2.3 Learning skills To identify problems and generate solutions. To generalize (or specialize) concepts, models and methods. To establish analogies between different conceptualizations, in order to transfer concepts and methods from an application domain to another. To be able to reflect on personal (technical and design) decisions and choices (meta-reflection).</p>	
Tecnologie web per il cloud e laboratorio	ING-INF/05	<p>Capacità relative alle discipline Lo/la studente/essa dovrà:</p> <p>1.1. Conoscenza e capacità di comprensione Conoscere i concetti base della programmazione lato server Conoscere la sintassi e le principali istruzioni e strutture del linguaggio PHP Conoscere i principi della Service Orientation Conoscere le caratteristiche specifiche ed i protocolli relativi ai servizi Web di tipo SOAP e di tipo REST, con attenzione al formato JSON</p>	<p>Sector-specific skills</p> <p>1.1. Knowledge and understanding To know the main concepts of server-side programming To know syntax and main structures and statements of PHP To know the principles of Service Orientation. To know the specific features and protocols for SOAP- and REST-based web services, with attention to the JSON format.</p>	CONSIGLIATE: Tecnologie web e laboratorio, Programmazione e laboratorio

	<p>Conoscere i principi del Cloud ed i tipi di servizi forniti dai cloud provider</p> <p>1.2 Capacità di applicare conoscenza e comprensione Saper implementare un'applicazione Web interattiva Saper implementare un'applicazione che utilizza servizi web esterni Saper implementare un provider di servizi Web Saper utilizzare alcuni servizi cloud nelle proprie applicazioni sostenere argomentazioni che per risolvere problemi nel proprio campo di studi;</p> <p>Capacità trasversali / soft skills Lo/la studente/essa dovrà:</p> <p>2.1 Autonomia di giudizio Saper valutare quale linguaggio lato server utilizzare per lo sviluppo di un'applicazione Web Saper selezionare le librerie ed i servizi web necessari per lo sviluppo di un determinato progetto Saper identificare le funzioni da fornire tramite web services Saper identificare quali, tra le risorse fornite dai vari cloud provider, siano le più adeguate alla soluzione di un determinato problema implementativo, sia dal punto di vista delle funzionalità che dei costi.</p> <p>2.2 Abilità comunicative. Essere in grado di motivare, a voce e per iscritto, le scelte implementative effettuate nello sviluppo di un progetto basato su tecnologie lato server, web services o cloud.</p> <p>2.3 Capacità di apprendimento Saper reperire ed utilizzare risorse informatiche e</p>	<p>To know principles of Cloud computation and service types as offered by the main cloud providers.</p> <p>1.2 Applying knowledge and understanding Be able to implement an interactive web application. Be able to implement an application that calls external web services. Be able to implement a web service provider. Be able to adopt some cloud services in his/her own applications.</p> <p>Cross-sectoral skills/soft skills 2.1 Making judgements Be able to evaluate which server-side language to choose for a specific web application. Be able to select libraries and web services needed for a specific project. Be able to identify functions to be exposed via web services. Be able to identify which resources available from cloud providers are adequate to solve a specific implementation problem, from both functionality and cost points of view.</p> <p>2.2 Communication skills. Be able to discuss the implementation choices made in the development of a project based on server-side technologies, web services, or cloud.</p> <p>2.3 Learning skills Be able to find and use informatic and scientific resources for an autonomous close examination of the course topics. Be able to learn new server-side technologies</p>	
--	---	--	--

		<p>scientifiche per l'approfondimento autonomo delle tematiche studiate a lezione</p> <p>Saper affrontare lo studio di nuove tecnologie lato server</p> <p>Saper individuare nuovi servizi web, inclusi quelli di tipo cloud, che possano essere utili per lo sviluppo di applicazioni di interesse</p>	<p>Be able to identify new web services, including cloud services, that could be useful to develop his/her own applications.</p>	
--	--	---	--	--