

Allegato B2

Quadro degli obiettivi formativi specifici e delle propedeuticità

Corso di Laurea in Internet Of Things: Big Data & Web

Curriculum unico

Rau, art. 12

Insegnamento	Settore Scientifico Disciplin.	Obiettivi formativi specifici (ITA)	Specific educational objectives (ENG)	Propedeuticità obbligatorie
Elementi di Matematica	MAT/01	<p>Questo corso, primo fra i corsi di matematica che gli studenti del corso di IBW devono affrontare, ha l'obiettivo di 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. Il corso è suddiviso in quattro parti:</p> <p><u>1.Introduzione al linguaggio matematico elementare</u></p> <p>Obiettivo di questa parte è porre le basi per riuscire ad esprimersi correttamente nel linguaggio matematico. Verranno analizzate e messe a confronto diverse notazioni e definizioni insiemistiche e si svilupperanno dei primi ragionamenti astratti in questo ambito.</p> <p><u>2.Funzioni e relazioni</u></p> <p>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><u>3.Aritmetica e congruenze</u></p>	<p>This is the first course of mathematics for IBW students and its 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.</p> <p>The course is divided into four parts:</p> <p><u>1. Introduction to the elementary language of mathematics</u></p> <p>In this part, we do the groundwork needed for the correct use of the mathematical language. We analyze and compare different set notations and definitions, and we develop the first abstract reasonings using them.</p> <p><u>2.Functions and relations</u></p> <p>We introduce the first definitions using examples that allows overcoming the first difficulties we met when the level of abstraction increases, as when we consider the set of equivalence classes.</p> <p><u>3.Arithmetics and congruences</u></p> <p>The first training ground for developing reasoning skills. We use this material as an</p>	

	<p>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><u>4. Calcolo combinatorio</u></p> <p>Vengono analizzate diverse tecniche per il calcolo delle cardinalità di insiemi finiti.</p> <p>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>Capacità relative alle discipline</p> <p><i>1.1 Conoscenza e capacità di comprensione</i></p> <p>Conoscere i concetti di base della matematica elementare ed i suoi teoremi più significativi. Saper enunciare correttamente ed esemplificare i teoremi studiati.</p> <p><i>1.2 Conoscenza e capacità di comprensione applicate</i></p> <p>Saper sviluppare semplici dimostrazioni e modellare matematicamente semplici problemi combinatori.</p> <p>Capacità trasversali</p> <p><i>2.1 Autonomia di giudizio</i></p> <p>Saper riconoscere un ragionamento corretto. Saper valutare quale strumento utilizzare per risolvere uno specifico problema.</p> <p><i>2.2 Abilità comunicative</i></p> <p>Saper formalizzare le proprie intuizioni e spiegare in modo comprensibile una dimostrazione matematica.</p> <p><i>2.3 Capacità di apprendimento</i></p> <p>Essere in grado di approfondire autonomamente gli argomenti trattati nel corso.</p>	<p>example of the use of elementary mathematics in applications, for example in cryptography.</p> <p><u>4. Combinatorics</u></p> <p>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 reasonings skills acquired in the previous parts of the program.</p> <p>Sector-specific skills</p> <p><i>1.1 Knowledge and understanding</i></p> <p>To know the basic concept of elementary mathematics.</p> <p>To know how to apply the theoretical material acquired in the course to specific examples.</p> <p><i>1.2 Applying knowledge and understanding</i></p> <p>To know how to develop easy proofs and how to build mathematical models of easy combinatorial problems.</p> <p>Cross-sectoral Skills/soft skills</p> <p><i>2.1 Making judgements</i></p> <p>To be able to recognize a correct reasoning. To know what tool is needed to solve a specific problem.</p> <p><i>2.2 Communication skills</i></p> <p>To know how to communicate intuitions and explain in a clear way a mathematical proofs.</p> <p><i>2.3 Learning skills</i></p>	
--	--	---	--

Algebra Lineare	MAT/01	<p>In questo 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>Il corso è suddiviso in quattro parti. Nella prima parte si considerano il passaggio dai numeri reali ai numeri complessi (anche da un punto di vista storico) ed i paradossi relativi alla nozione di cardinalità per insiemi infiniti. Lo studente acquisisce così una visione della matematica meno statica di quella presentata in ambito scolastico.</p> <p>Nella seconda parte si studiano i 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>Nella terza parte le definizioni e le tecniche dell'algebra lineare (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>Nell'ultima parte del corso si introducono le trasformazioni lineari e la loro rappresentazione tramite matrici, considerando prima 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</p>	<p>In this 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 course is divided into four parts. In the first part 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. In this way, the student acquires an idea of mathematics, which is less static than the high school one. In the second part, linear systems and their matrix representation are considered. Here the student learn how a mathematical representation can be useful to represent and solve a problem concisely. In the third part, definitions and tools of linear algebra (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. In the last part of the course, linear transformations and their matrix representation are considered, starting from rotations and symmetries in plane and space and then generalizing to higher dimension. Here we exemplify how certain motion can be reproduced in the computer using their mathematical representation.</p> <p>Sector-specific skills</p> <p><i>1.1 Knowledge and understanding</i></p>	
-----------------	--------	---	--	--

		<p>computer attraverso la loro rappresentazione matematica.</p> <p>Capacità relative alle discipline</p> <p><i>1.1 Conoscenza e capacità di comprensione</i> Conoscere le nozioni ed i teoremi fondamentali dell'algebra lineare ed essere in grado di esemplificare tali nozioni.</p> <p><i>1.2 Conoscenza e capacità di comprensione applicate</i> Saper formalizzare un problema usando le tecniche dell'algebra lineare.</p> <p>Capacità trasversali</p> <p><i>2.1 Autonomia di giudizio</i> Saper quale strumento utilizzare per risolvere un problema affrontabile con le tecniche dell'algebra lineare.</p> <p><i>2.2 Abilità comunicative.</i> Saper comunicare in modo efficace le soluzioni di un problema risolto usando le tecniche dell'algebra lineare. Saper scrivere in modo conciso la soluzione di un problema.</p> <p><i>2.3 Capacità di apprendimento</i> Essere in grado di approfondire in maniera autonoma argomenti di algebra lineare.</p>	<p>To know the basic notions and the fundamental theorem of linear algebra and to be able to apply them to specific examples.</p> <p><i>1.2 Applying knowledge and understanding</i> To know how to formalize real problems using the techniques of linear algebra.</p> <p>Cross-sectoral Skills/soft skills</p> <p><i>2.1 Making judgements</i> To know what tool is needed to solve a specific problem.</p> <p><i>2.2 Communication skills</i> To know how to communicate the solution of a problem using linear algebra techniques.</p> <p><i>2.3 Learning skills</i></p>	
Analisi Matematica	MAT/05	<p>Il corso di Analisi Matematica vuole fornire una solida preparazione di base nel calcolo differenziale e integrale in una variabile. La 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</p>	<p>The Calculus course gives a solid grounding in the differential and integral calculus in one variable. There are two purposes: familiarizing 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</p>	

		<p>progressiva familiarità che gli studenti acquisiscono con il metodo logico-deduttivo. 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 <i>Conoscenza e capacità di comprensione:</i> conoscere il significato geometrico e algebrico di limiti, derivate, integrali. <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 <i>Autonomia di giudizio:</i> saper collegare fra loro idee algebriche, geometriche e computazionali. <i>Abilità comunicative:</i> saper scrivere in modo comprensibile la risoluzione di un problema matematico, e saper dimostrare oralmente un teorema. <i>Capacità di apprendimento:</i> saper leggere un testo scientifico che usa il metodo dimostrativo e gli strumenti di base dell'analisi matematica.</p>	<p>discovery of the logical-deductive method. 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 <i>Knowledge and understanding:</i> The algebraic and geometrical meaning of limit, derivative, integral. <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 <i>Making judgements:</i> Being able to make useful connections between algebraic, geometric and computational ideas. <i>Communication skills:</i> Being able to write a report on how to solve a mathematical problem, and to prove a theorem orally. <i>Learning skills:</i> Being able to read a scientific text that uses the demonstrative method and the basic notions of calculus.</p>	
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>The course aims at describing the structure and operations of the diverse components forming a computer, by illustrating the most popular technologies.</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), le architetture parallele (pipelining, superscalari, sistemi multiprocessore).</p> <p>Capacità relative alle discipline</p> <p>1.1 <i>Conoscenza e capacità di comprensione</i>: 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 <i>Conoscenza e capacità di comprensione applicate</i>: 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 <i>Autonomia di giudizio</i>: lo studente acquisisce gli strumenti per poter valutare e scegliere in maniera ponderata l'architettura hardware da utilizzare.</p> <p>2.2 <i>Abilità comunicative</i>: 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 <i>Capacità di apprendimento</i>: il corso mira a fornire allo studente un quadro concettuale</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), parallel architectures (pipelining, superscalar, multi-processor), 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 <i>Knowledge and understanding</i>: 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 <i>Applying Knowledge and understanding</i>: 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>Cross-sectoral skills/soft skills</p> <p>2.1 <i>Making Judgements</i>: students will learn tools for evaluating with ponder and choosing a hardware architecture.</p> <p>2.2 <i>Communication skills</i>: students will learn the exact meaning of the different words used</p>	
--	--	---	--

		sufficientemente rigoroso completo da permettergli di affrontare con sicurezza lo studio dettagliato delle varie componenti di un calcolatore.	for properly describing characteristics and specifications of a computer. 2.3 <i>Learning skills</i> : the course aims at providing a conceptual framework, precise enough to secure a detail study of the diverse components finding place in a computer.	
Programmazione e laboratorio	INF/01	<p>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, 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>The course introduces the scientific and methodological foundations of computer programming, in order for students to understand the nature of programming and to 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: - Expressions and procedures;</p>	

		<p>I. Astrazione procedurale:</p> <ul style="list-style-type: none"> - Procedure come astrazione di espressioni; - Ricorsione, ricorsione ad albero e ricorsione di coda; - 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>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</p> <p>Lo/la studente/essa dovrà:</p> <p>1.1. <i>Conoscenza e capacità di comprensione</i></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; - Conoscere le relazioni fra ricorsione e iterazione; - Conoscere le relazioni fra i concetti di classe e oggetto. <p>1.2 <i>Capacità di applicare conoscenza e comprensione</i></p> <ul style="list-style-type: none"> - Saper codificare semplici algoritmi ricorsivi e imperativi; 	<ul style="list-style-type: none"> - 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; - 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. <i>Knowledge and understanding</i></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 <i>Applying knowledge and understanding</i></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; 	
--	--	---	---	--

		<ul style="list-style-type: none"> - 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. <p>Capacità trasversali / soft skills Lo/la studente/essa dovrà:</p> <p><i>2.1 Autonomia di giudizio</i></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><i>2.2 Abilità comunicative.</i></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><i>2.3 Capacità di apprendimento</i></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. 	<ul style="list-style-type: none"> - 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>Cross-sectoral skills/soft skills</p> <p><i>2.1 Making judgements</i></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><i>2.2 Communication skills.</i></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><i>2.3 Learning skills</i></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. 	
Tecnologie Web e Laboratorio	ING-INF/05	Il corso si propone di presentare agli studenti il World-Wide Web, le tecnologie informatiche e della comunicazione (ITC) che sono coinvolte; le implicazioni che da esse traggono origine. Si	The course is aimed at introducing the students to the World Wide Web, its design principles and architecture and the main	

	<p>fornisce una panoramica generale, corredata da esercitazioni di laboratorio, che anticipa i contenuti di corsi successivi e inserisce lo studente nelle problematiche dell'intero corso di laurea. Vengono introdotti i principi fondamentali del Web e i criteri di progettazione che ne derivano; la infrastruttura di rete sui quali si realizza il servizio WWW; i linguaggi di rappresentazione (XML, HTML5) e presentazione (CSS3); i concetti di base e le tecnologie abilitanti del web dinamico e del web semantico. Alla fine del corso lo/la studente/essa dovrà:</p> <p>Capacità relative alle discipline <i>Conoscenza e capacità di comprensione</i> Conoscere la terminologia e le concettualizzazioni di base per analizzare, descrivere, usare e valutare tecnologie web Conoscere i principi, le metodologie, le tecniche, gli strumenti per lo sviluppo di applicazioni web <i>Conoscenza e capacità di comprensione applicate</i> Saper applicare le conoscenze e le metodologie di base per realizzare siti e portali web di tipo statico</p> <p>Capacità trasversali/soft skills Alla fine del corso lo/la studente/essa dovrà essere in grado di: <i>Autonomia di giudizio</i> Confrontare e valutare criticamente due o più applicazioni web da diversi punti di vista (tecnico, comunicativo) e a diversi livelli di aggregazione e astrazione <i>Abilità comunicative</i> Saper descrivere le caratteristiche principali di una applicazione web</p>	<p>technologies supporting it. At the end of the course, the student should be able</p> <p>Sector-specific skills</p> <p>1. <i>Knowledge and understanding</i> To know the main conceptualizations and the technical vocabulary for the analysis, description and evaluation of web technologies; To know basic principles, methods and tools for the development of web applications</p> <p>2. <i>Applying knowledge and understanding</i> To apply conceptual and methodological knowledge for developing static web sites</p> <p>Cross-sectoral skills/soft skills</p> <p>3. <i>Making judgments</i> To compare, evaluate and criticize two or more web applications from different points of view and at different aggregation levels</p> <p>4. <i>Communication skills</i> To describe the main features of a web application and to explain how it achieves the intended goals and requirements</p> <p>5. <i>Learning skills</i> To generalize/specialize concepts, models and methods</p>	
--	---	--	--

		<p>Saper argomentare e spiegare le motivazioni che stanno alla base delle scelte di progetto e di implementazione di una applicazione web</p> <p>Saper costruire mappe concettuali del dominio di studio</p> <p><i>Capacità di apprendimento</i></p> <p>Saper identificare problemi e proporre soluzioni</p> <p>Saper generalizzare/specializzare concetti, modelli e metodi</p> <p>Saper stabilire analogie tra concettualizzazioni e trasferire concetti e metodi tra domini differenti</p> <p>Saper riflettere su scelte e decisioni</p>		
Algoritmi e strutture dati e laboratorio	INF/01	<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 di ricerca/inserimento/cancellazione</i> 	<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> 	

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

		<p>Capacità trasversali / soft skills</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>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 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>	
Programmazione orientata agli oggetti e laboratorio	ING-INF/05	Il corso prevede di formare lo studente relativamente a questi argomenti:	The course is going to deal with the following topics:	

		<ul style="list-style-type: none"> • 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 (singleton, factory). <p>Lo studente/la studentessa dovrà:</p> <p>Capacità relative alla disciplina</p> <p><i>Conoscenza e capacità di comprensione</i></p> <ul style="list-style-type: none"> • conoscere i concetti di base della orientazione agli oggetti • conoscere alcuni pattern di design orientato agli oggetti • conoscere alcuni principi di progettazione (singola responsabilità, Liskov, inversione dipendenza) e di programmazione (clean code) <p><i>Capacità di applicare conoscenza e comprensione</i></p> <ul style="list-style-type: none"> • saper utilizzare Java per la programmazione orientata agli oggetti • saper leggere codice Java di tipici progetti open source 	<ul style="list-style-type: none"> • "In the large" vs "in the small" programming: the quality role in software programming. • Abstraction types: procedural, parametric, per specific abstractions. • Clean code approach • Programs as total or partial functions; exceptions; defensive and assertive programming. • Different types of abstract data • Hierarchies of types; interfaces; abstract classes; the reversal of dependency principle. Liskov principle. • Generical types. • Anonymous classes. • Introduction to object oriented programming: CRC technique, the single responsibility principle, some design patterns (singleton, factory). <p>At the end of the course the student is expected to:</p> <p>Sector specific skills</p> <p><i>Knowledge and understanding</i></p> <ul style="list-style-type: none"> • Know the basic concepts of object orienting • Know some patterns of object oriented design • Know some design (single responsibility, Liskov, reversal of dependency) and programming principles (clean code) <p><i>Applying knowledge and understanding</i></p> <ul style="list-style-type: none"> • Know how to use Java for object oriented programming 	
--	--	--	--	--

		<ul style="list-style-type: none"> saper applicare alcuni pattern di design. Questi obiettivi saranno supportati da attività guidata di laboratorio. <p>Capacità trasversali/soft skills</p> <p><i>Autonomia di giudizio</i></p> <ul style="list-style-type: none"> saper valutare la qualità del codice prodotto <p><i>Abilità comunicative</i></p> <ul style="list-style-type: none"> essere in grado di descrivere e motivare le scelte fatte <p><i>Capacità di apprendimento</i></p> <ul style="list-style-type: none"> essere in grado di approfondire autonomamente gli argomenti toccati a lezione 	<ul style="list-style-type: none"> Know how to read typical open source design codes in Java Know how to use some design patterns <p>These objectives will be supported by guided laboratory activities.</p> <p>Cross-sectoral skills/soft skills</p> <p><i>Making judgements</i></p> <ul style="list-style-type: none"> Be able to assess the quality of the produced code <p><i>Communication skills</i></p> <ul style="list-style-type: none"> Be able to explain and give reasons for decisions <p><i>Learning skills</i></p> <ul style="list-style-type: none"> Be able to autonomously study in depth the topics explained during the lessons 	
Sistemi operativi e laboratorio	ING-INF/05	<p>Capacità relative alla disciplina</p> <p><i>Conoscenza e capacità di comprensione:</i> lo studente acquisisce conoscenze specifiche sull'architettura, il 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.</p> <p><i>Capacità di applicare conoscenza e comprensione:</i> grazie all'attività di laboratorio, lo studente acquisisce specifiche capacità di applicare a casi reali le conoscenze maturate sui sistemi operativi.</p> <p>Capacità trasversali/soft skills</p> <p><i>Autonomia di giudizio:</i> lo studente acquisisce una capacità di valutazione critica sulle tecnologie, gli algoritmi e le soluzioni che possono influire positivamente o negativamente sulle prestazioni</p>	<p>Sector-specific skills</p> <p><i>Knowledge and understanding:</i> the student will acquire specific knowledge about the architecture, the operation and the underlying design principles of a modern operating system. She/He will also learn how to evaluate and how to choose among several operating systems, depending on the purpose of the implementation and on the hardware features.</p> <p><i>Applying knowledge and understanding:</i> thanks to the lab activities, the student will get specific skills in the application of the acquired knowledge of operating systems to real case-studies.</p> <p>Cross-sectoral skills/soft skills</p> <p><i>Making judgements:</i> the student will acquire a critical appraisal skill in appropriately selecting</p>	Architettura degli Elaboratori e laboratorio, Programmazione e laboratorio. Computer Systems Architecture, Programming and laboratory.

		<p>e sul corretto funzionamento di un sistema operativo in diversi contesti applicativi.</p> <p>Abilità comunicative: lo studente impara a descrivere in modo tecnicamente corretto ed usando la terminologia appropriata un sistema operativo.</p> <p><i>Capacità di apprendimento:</i> lo studente impara ad essere autonomo nell'espandere le proprie conoscenze sui sistemi operativi oltre le nozioni e gli esempi appresi ed analizzati durante il corso, scoprendo e sfruttando anche le connessioni con altri corsi riguardanti l'architettura dei calcolatori, le reti ed i sistemi distribuiti.</p>	<p>technologies, algorithms and solutions according to their outcome on the performance and on the correct operation of an operating system in different application contexts.</p> <p><i>Communication skills:</i> the student will learn how to describe an operating system using a technically correct and appropriate terminology.</p> <p><i>Learning skills:</i> the student will learn how to be autonomous in expanding his knowledge of operating systems beyond the notions and examples learned and analyzed during the course, discovering and exploiting connections with other courses such as computer architecture, computer networks, and distributed systems.</p>	
<p>Statistica Applicata Applied Statistics</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>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</p>	

	<p>Capacità relative alle discipline 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>and data analysis. The students are expected to acquire the fundamental skills in order to 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>2.3 Learning skills</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>Learning skills based on the prerequisites that are required for understanding autonomously a statistical report and for applying more advanced statistical techniques.</p>	
Tecnologie Web per il Cloud	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 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</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. 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</p> <p>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.</p>	

		<p>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 scientifiche per l'approfondimento autonomo delle tematiche studiate a lezione Saper affrontare lo studio di nuove tecnologie lato server 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 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 Be able to identify new web services, including cloud services, that could be useful to develop his/her own applications.</p>	
Fondamenti di Scienza dei Dati e Laboratorio	INF/01	<p>In questo corso lo studente imparerà i fondamenti della scienza dei dati utilizzando l'ambiente R. Il corso è suddiviso in cinque blocchi principali: 1. Esplorare. L'obiettivo di questa parte è di imparare gli strumenti di base per l'esplorazione dei dati il più rapidamente possibile. In particolare, in questa parte lo studente imparerà a visualizzare, trasformare e analizzare in modo</p>	<p>In this course, you will learn the fundamentals of data science using R environment. The course is divided in five main chunks: 1. Explore. The goal of this part is to learn the basic tools of data exploration as quickly as possible. In particular, in this part you will learn the basics of visualization, data transformation, and exploratory data analysis. Moreover, you will move the first steps in the R environment.</p>	

	<p>esplorativo i dati. Inoltre, muoverà i primi passi nell'ambiente R.</p> <p>2.Riordinare. In questa parte lo studente imparerà ad ottenere i dati in R in una forma utile per la visualizzazione e la modellazione. Ci concentreremo principalmente su dati strutturati (relazionali) e in misura minore su dati semi-strutturati e non strutturati.</p> <p>3.Programmare. In questa parte lo studente potrà migliorare le abilità di programmazione. In particolare, lo studente imparerà le strutture di dati, l'iterazione, le funzioni e la concatenazione di istruzioni (pipe) in R.</p> <p>4.Modellare. I modelli sono uno strumento utile per il partizionamento dei dati in struttura e in ciò che rimane (rumore). Saranno spiegate le basi della modellazione e come sia possibile applicarle al fine di dare un senso ad insiemi di dati complessi.</p> <p>5.Comunicare. Verrà insegnato R Markdown, uno strumento per l'integrazione di prosa, codice e risultati.</p> <p>Capacità relative alle discipline</p> <p><i>Conoscenza e capacità di comprensione:</i> lo studente deve aver acquisito le conoscenze necessarie per importare, riordinare, trasformare, visualizzare e modellare i dati, così come comunicare i risultati delle analisi. Gli strumenti si concentreranno principalmente su dati strutturati (in formato relazionale), e in misura minore su dati semi-strutturati (XML) e non strutturati (testo).</p> <p><i>Conoscenza e capacità di comprensione applicate:</i> lo studente deve aver imparato l'analisi dei dati e la visualizzazione in ambiente R, così come la</p>	<p>2. Wrangle. In this part, you will learn about data wrangling, the art of getting your data into R in a useful form for visualization and modelling. In particular, you will learn about tidy data, a consistent way of storing your data that makes transformation, visualization, and modelling easier. You will mainly focus on structured (relational) data, and to a lesser degree on semi structured and unstructured data.</p> <p>3. Programming. In this part, you will improve your programming skills. In particular, you will learn data structures, iteration, functions, and pipes using R.</p> <p>4. Modelling. Models are a useful tool for partitioning data into structure and what remains (noise). Here you will learn the basics of modelling and how you can apply them to make sense of complex datasets.</p> <p>5. Communication. You will learn about R Markdown, a tool for integrating prose, code, and results.</p> <p>Sector-specific skills</p> <p><i>Knowledge and understanding:</i> the student must have acquired the necessary knowledge to import, tidy, transform, visualize, and model data as well as communicate the results of the analysis. The method will mainly focus on relational data, although semi structured as well as unstructured data will also be touched.</p> <p><i>Applying knowledge and understanding:</i> the student must have learned R environment for data analysis and visualization, as well as R Markdown language for communication of results of the analysis. This includes the skills</p>	
--	---	--	--

		<p>comunicazione dei risultati utilizzando il linguaggio R Markdown. Questo include le competenze di programmazione efficiente in R utilizzando strutture di dati adeguate.</p> <p>Capacità trasversali/soft skills</p> <p><i>Autonomia di giudizio:</i> lo studente deve essere in grado di interpretare i risultati sperimentali delle analisi e trarre conclusioni efficaci relative al dominio del discorso.</p> <p><i>Abilità comunicative:</i> lo studente deve essere in grado di comunicare in modo efficace i risultati delle analisi. Questo include la comunicazione analista-analista e quella analista-decisore.</p> <p><i>Capacità di apprendere:</i> lo studente deve dimostrare di aver imparato la capacità di scegliere un sufficientemente ricco caso di studio, analizzare i dati per estrarre informazioni significative, trarre conclusioni e saperle comunicare efficacemente.</p>	<p>of efficient programming in R using suitable data structures.</p> <p>Cross-sectoral skills/soft skills</p> <p><i>Making judgments:</i> the student must be able to interpret the experimental results of the analysis and draw effective conclusions relevant to the domain of discourse.</p> <p><i>Communication skills:</i> the student must be able to communicate effectively the results of the analysis. This includes both analyst-to-analyst communication and analyst-to-decision-maker communication.</p> <p><i>Learning skills:</i> the students must demonstrate that they have learned the ability to choose a sufficiently rich row data set, analyze the data to extract meaningful information, draw and communicated conclusions.</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</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</p>	

		<p>apprendimento automatico attraverso una corretta formulazione del problema, una scelta critica dell’algoritmo di apprendimento e 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>1.2 Conoscenza e capacità di comprensione applicate: 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.</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>algorithm and an experimental analysis of the obtained results. The course includes laboratory activities so that students can 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.</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>	
Internet of Things	INF/01	L’obiettivo del corso è fornire allo studente le conoscenze fondamentali e le abilità pratiche	The aim of the course is to provide students with basic knowledge and practical skills about	

		<p>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 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 	<p>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. 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> <ol style="list-style-type: none"> 1) Hardware: rapid prototyping platforms (Arduino), single-board computers (Raspberry), sensors, wireless communication technologies (RFID, Bluetooth LE, Wi-Fi) 2) M2M communication protocols (DDS, AMQP, MQTT, JMS, REST, CoAP etc.) 3) Stream Processing Algorithms 4) Programming issues related to IoT: concurrency, real-time computing, synchronization, error handling 5) Frameworks for building IoT applications <p>Sector-specific skills</p>	
--	--	---	--	--

		<p>computing, sincronizzazione, gestione degli errori</p> <ul style="list-style-type: none"> • 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 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>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 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>	
--	--	--	--	--

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 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>	<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 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”. 	
----------------------------	--------	--	---	--

		<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. • 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. 	<ul style="list-style-type: none"> • 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. <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. 	
--	--	--	--	--

		<ul style="list-style-type: none"> • 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. 	<ul style="list-style-type: none"> • 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. 	
Ingegneria del Software	ING-INF/05	<p>Gli obiettivi del corso sono i seguenti:</p> <ul style="list-style-type: none"> - Introdurre ai concetti di base dell'Ingegneria del Software (IS), ossia il settore ingegneristico dell'informatica dedicato allo studio delle metodologie, delle tecniche e degli strumenti utilizzati in tutti gli aspetti della produzione industriale del software. - Capire quali sono i principali problemi che si incontrano nello sviluppo del software a livello professionale ed industriale. 	<p>The course objectives are:</p> <ul style="list-style-type: none"> - Introducing basic concepts, principles, and techniques of Software Engineering (SE), i.e. the discipline of Computer Science devoted to the professional and industrial development of software. - Understanding the main problems arising during the software development process. - Knowing how the software development process is organized 	

		<p>- Conoscere come è organizzato il processo di produzione e di sviluppo del software</p> <p>Capacità relative alle discipline Lo studente dovrà:</p> <ol style="list-style-type: none"> 1. <i>Conoscenza e capacità di comprensione:</i> acquisire specifiche conoscenze dei principali concetti e principi dell'IS. Conoscere molteplici modelli del processo di sviluppo del software e le varie fasi del ciclo di vita, con riferimento sia al software tradizionale, che al software avanzato. Conoscere diverse tecniche di analisi e di progettazione del software, ivi inclusi i linguaggi: UML, Data Flow Diagram e Reti di Petri. Conoscere i maggiori fattori di criticità nello sviluppo del software. Conoscere le principali tecniche di verifica e validazione del software, ivi incluse le tecniche di testing e di software inspection. Conoscere i principali parametri di qualità ed i principali standard del software. Conoscere le nuove tecniche agili di sviluppo del software. 2. <i>Capacità di applicare conoscenza e comprensione:</i> sapere come analizzare e rappresentare i risultati dell'analisi e della progettazione in uno specifico dominio applicativo. Saper selezionare le metodologie e le tecniche di sviluppo più adeguate alle varie situazioni. <p>Capacità trasversali / soft skills Lo studente dovrà:</p> <ol style="list-style-type: none"> 1. <i>Autonomia di giudizio:</i> saper valutare indipendentemente le caratteristiche ed i requisiti di un'applicazione informatica ed 	<p>Sector-specific skills The student will have to:</p> <ol style="list-style-type: none"> 1. <i>Knowledge and understanding:</i> acquiring specific knowledge of the main concepts and basic principles of SE. Knowing the various development models exploited for producing software, the various phases of the software life cycle, both for traditional software and for innovative software. Knowing the different techniques for analysis and design, including languages such as UML, Data Flow Diagrams, and Petri Nets. Knowing the major critical problems encountered in software development. Knowing the main verification and validation techniques, including testing and software inspection. Knowing the main quality criteria and standards for software. Knowing the new agile methodologies for software development. 2. <i>Ability to apply knowledge and understanding:</i> knowing how to analyze and represent the results of analysis and design in a specific application context. Knowing how to select the most appropriate technique for the specific situation considered. <p>Cross-sectoral skills/soft skills The student will have to:</p> <ol style="list-style-type: none"> 1. <i>Autonomy of judgment:</i> be able to independently evaluate the characteristics of a computer application 	
--	--	---	--	--

		<p>essere in grado di rappresentarne l'analisi e la progettazione.</p> <p>2. <i>Abilità comunicative</i>: saper illustrare con rigore logico e terminologico, a voce e per iscritto mediante varie tecniche di rappresentazione, gli elementi tecnici rilevanti di un progetto software.</p> <p>3. <i>Capacità di apprendimento</i>: essere in grado di apprendere le nozioni base dell'IS, al fine poi all'occorrenza di raffinare e di approfondire specifici settori della disciplina.</p>	<p>and to be able to represent the results of analysis and design.</p> <p>2. <i>Communicative skills</i>: acquiring the ability to describe effectively and through appropriate models the most relevant aspects of a software project</p> <p>3. <i>Learning skills</i>: be able to learn the basics of SE, in order to possibly later refine and deepen specific areas of the discipline.</p>	
<p>Interazione Uomo-Macchina Human-Computer Interaction</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>Capacità relative alle discipline</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>Sector-specific skills</p> <p>1.1 <i>Knowledge and understanding</i>: Students acquire specific multidisciplinary knowledge</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, 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.</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>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</p>	<p>on the characteristics of users and human-computer interfaces. They also learn to evaluate and choose from various interaction techniques, 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.</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 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>	
--	--	--	--	--

		autonomamente problemi inerenti alla progettazione e valutazione di interfacce utente.		
Reti di calcolatori	INF/01	<p>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 affrontate le problematiche di sicurezza delle comunicazioni tra calcolatori; a questo scopo, verranno introdotte anche alcune nozioni di crittografia.</p> <p>Capacità relative alle discipline</p> <p><i>1.1 Conoscenza e capacità di comprensione:</i> 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 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.</p> <p><i>1.2 Conoscenza e capacità di comprensione applicate:</i> grazie ad una serie di esempi presentati a lezione e esercizi mirati alla risoluzione di casi pratici, lo studente acquisisce specifiche capacità di applicare a casi reali le conoscenze maturate sui vari aspetti della disciplina.</p>	<p>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</p> <p><i>1.1 Knowledge and understanding:</i> 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.</p> <p><i>1.2 Applied knowledge and understanding:</i> 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 real-life cases the knowledge about the various aspects of the discipline.</p> <p>Cross skills / soft skills</p>	

		<p>Capacità trasversali/soft skills</p> <p><i>2.1 Autonomia di giudizio:</i> 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><i>2.2 Abilità comunicative:</i> 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><i>2.3 Capacità di apprendimento:</i> 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><i>2.1. 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><i>2.2 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><i>2.3 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>	
Social Computing	ING-INF/05	<p>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. 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>	<p>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. 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"> • 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 • Applicazioni: uso di una piattaforma di crowdsourcing (Crowdfunder o Amazon's Mechanical Turk) per progettazione ed esecuzione di esperimenti. Analisi dei risultati <p><i>4. Approfondimenti</i></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>a.Examples of social media, relevance to data science (socials are a source of data and users, and a ground where interesting phenomena happen)</p> <p>b.Examples of Crowdsourcing. Success stories and failures.</p> <p><i>2.Social media</i></p> <p>a.Concepts: Definition. Examples. Classification (generalist, verticals, private)</p> <p>bFOUNDATIONS: Historical background (social network analysis, network science). Elementary network science.</p> <p>c.Applications. APIs to access data from socials (case studies: Twitter or Facebook or Telegram)</p> <p><i>3.Crowdsourcing</i></p> <p>a.Concepts: Definitions, Examples. Collective intelligence</p> <p>b.FOUNDATIONS: The general case of Human computation. Characteristics needed for successful Crowdsourcing. Computability (brief account)</p> <p>c.Applications: usage of a Crowdsourcing platform (Crowdfunder or Amazon's Mechanical Turk) to design and run experiments. Analysis of results.</p> <p><i>4.Case studies and specific issues</i></p> <p>a.Ethical, moral, legal aspects</p> <p>b.Economic aspects (SEO, business models, crowdfunding)</p> <p>c.Social-aware programming (multiagent systems, society design; genetic algorithms; map/reduce)</p> <p>d.Hybrid systems</p> <p>Sector-specific skills</p>	
--	--	--	---	--

		<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> <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,</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 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</p>	
--	--	--	--	--

		l'internet of things, il machine learning e la programmazione.	science, internet of things, machine learning, and programming.	
--	--	--	---	--