



Bollettino Notiziario - A.A. 2021/2022

LAUREA MAGISTRALE IN COMPUTER SCIENCE (ORD. 2021)

Curriculum: Corsi comuni

ADVANCED ALGORITHMS

Titolare: Prof. DAVIDE BRESOLIN

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Il corso richiede familiarità con alcuni concetti algoritmici di base, quali complessità asintotica, algoritmi di ricerca e ordinamento e strutture dati di base come alberi, liste, mappe, vettori e matrici. Non ci sono corsi propedeutici.

Conoscenze e abilità da acquisire:

Il corso si pone l'obiettivo di insegnare agli studenti a pensare "in modo algoritmico", fornendo le seguenti conoscenze e abilità attese: 1. Essere in grado di comprendere un problema da risolvere, di quali sono i dati disponibili e come deve essere fornita la soluzione. 2. Essere in grado di formulare un problema in termini di input e output trattabili da un calcolatore. 3. Essere in grado di definire un algoritmo e di analizzarne le proprietà di correttezza e complessità computazionale. 4. Essere in grado di implementare un algoritmo in un linguaggio di programmazione concreto. 5. Essere in grado eseguire l'implementazione su un dataset di media dimensione e di comprenderne ed analizzarne i risultati.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso è suddiviso in blocchi tematici. Ogni blocco inizia con una serie di lezioni frontali in aula durante le quali vengono affrontati i contenuti del corso. Terminata l'attività frontale, il blocco prosegue con una o più lezioni di laboratorio dove gli studenti divisi a gruppi implementeranno e testeranno la soluzione del problema reale su un dataset di media dimensione, e si conclude con una fase di confronto e discussione delle varie soluzioni realizzate.

Contenuti:

1) Algoritmi su grafi: Nozioni di base sui grafi e loro rappresentazione. Visite in ampiezza e in profondità di un grafo, con applicazioni. Componenti Connesse. Grafi pesati: cammini minimi e alberi di connessione minimi. Strutture dati per insiemi disgiunti. 2) Algoritmi di approssimazione: Problemi trattabili e problemi intrattabili. Problemi NP-completi e algoritmi di approssimazione. Approssimazione per i problemi del vertex cover e del set cover. Il problema del Commesso Viaggiatore: inapprossimabilità e algoritmi di approssimazione per casi speciali. 3) Algoritmi randomizzati: Tecniche principali e applicazioni: le disuguaglianze di Markov, i bound di Chernoff, analisi di Quicksort, algoritmi randomizzati per il problema del taglio minimo

Modalità di esame:

L'esame è diviso in una parte di teoria ed una parte pratica. La parte di teoria consiste in una prova scritta che si tiene nelle normali sessione d'esame. La parte pratica può essere svolta in due modalità diverse: - DURANTE IL CORSO: un gruppo di massimo tre studenti implementa tutti gli algoritmi visti nei laboratori e consegna il codice ed i risultati ottenuti. - DOPO IL CORSO: un unico progetto personale. In questo caso il progetto va svolto da soli, non in gruppo.

Criteri di valutazione:

I criteri di valutazione con cui verrà effettuata la verifica delle conoscenze e delle abilità acquisite sono: 1. Completezza delle conoscenze acquisite; 2. Proprietà della terminologia tecnica utilizzata; 3. Capacità di formalizzare un problema in termini algoritmici 4. Capacità di analizzare correttezza e complessità computazionale di un algoritmo 5. Capacità di implementare un algoritmo in modo coerente con la teoria 6. Capacità di utilizzare un algoritmo per risolvere un problema concreto

Testi di riferimento:

Cormen, Thomas H., Introduction to algorithms. Cambridge (Mass.): MIT Press, 2009

Eventuali indicazioni sui materiali di studio:

Il corso ha una sezione dedicata sul Moodle del Dipartimento di Matematica. Il Moodle raccoglierà possibili dispense, le specifiche dettagliate dei singoli laboratori, esercizi e loro soluzioni. Verrà usato anche per comunicazioni e aggiornamenti da parte dei docenti. Altri testi consigliati oltre al testo adottato: - J. Kleinberg and E. Tardos: Algorithm Design, Addison-Wesley, 2005 - V. Vazirani: Approximation Algorithms, Springer, 2001 - R. Motwani and P. Raghavan: Randomized Algorithms, Cambridge University Press, 1995

ADVANCED TOPICS IN COMPUTER AND NETWORK SECURITY

Titolare: Prof. MAURO CONTI

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Il corso non prevede propedeuticità. Tuttavia, sono consigliate conoscenze di base di reti, crittografia, e sistemi distribuiti (tipicamente acquisite nei corsi di Laurea in Informatica).

Conoscenze e abilità da acquisire:

Approfondire i concetti di sicurezza di base, analizzando le più recenti proposte di ricerca nell'ambito. Al termine del corso gli studenti saranno in grado non solo di analizzare con spirito critico un sistema software nel suo complesso, ma anche di aggiornare autonomamente le proprie competenze nel settore, anche tramite risultati recenti della ricerca nell'area.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali; discussione di articoli scientifici.

Contenuti:

Teoria: sicurezza RFID, captcha, sistemi di archiviazione non sicuri, sicurezza sugli smartphone, attacchi su smartPhone, protezione di password, attacchi Denial-of-Service distribuiti, deep learning, biometria, sicurezza VoIP, secure content delivery, comunicazioni anonime, rilevamento keylogger, anonimato in WSN, rilevamento di botnet, HW affidabile, sicurezza dei passaporti RFID, attacco di tipo node replication in WSN, aggregazione sicura dei dati in WSN, problemi di privacy nei social media, sicurezza smartphone Android Google, sistemi di votazione elettronica, rilevazione botnet P2P, meccanismi di taint analysis, sicurezza dei browser, privacy di servizi di localizzazione, Named Data Networking security, Named Data Networking privacy, sicurezza dei sistemi cloud, anonimato nella rete wireless, profilazione di utenti su smartphone, problemi di sicurezza SSL in Android, circumvent censorship, secure messaging, sicurezza tecnologica operativa, sicurezza dei sistemi cyber-fisici. Laboratorio: strumenti di sicurezza avanzati, inclusi: analisi del traffico con strumenti di apprendimento automatico, inferenza di dati, strumenti di sicurezza in Android, meccanismi di attacco e difesa per buffer overflow; analisi avanzata di sistema Malware e Advanced Persistent Threat; sicurezza web; strumenti di analisi di reti sociali, trusted platform modules.

Modalità di esame:

Progetto con relazione + esame orale.

Criteri di valutazione:

Conoscenza dei concetti studiati nel corso.

Testi di riferimento:

W. Stallings, L. Brown, Computer Security: Principles and Practice 2/E. : Prentice Hall, M. Bishop, Introduction to Computer Security. : Addison-Wesley Professional,

Eventuali indicazioni sui materiali di studio:

Libro (testo principale Computer Security: Principles and Practice 2/E) e articoli scientifici. Il corso sarà tenuto in Inglese. Il sito web del corso offrirà tutte le informazioni e materiale ulteriore: <http://www.math.unipd.it/~conti/teaching.html>

ADVANCED TOPICS IN COMPUTER SCIENCE

Titolare: Dott. LUCA PASA

Periodo: I anno, annuale

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A; 6,00

Prerequisiti:

Nessun prerequisito

Conoscenze e abilità da acquisire:

Capacità di affrontare in modo critico temi avanzati, al confine con la ricerca in ambito informatico. Capacità di consultare letteratura scientifica.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali

Contenuti:

Il corso si compone di cicli di lezioni monografici, che illustrano temi avanzati dell'informatica con il supporto di esperti internazionali.

Modalità di esame:

Lo studente dovrà approfondire un tema prescelto e mostrarne quindi la padronanza, discutendolo in forma seminariale o sviluppando un progetto correlato.

Criteri di valutazione:

L'esame valuta la capacità dello studente di confrontarsi con tematiche avanzate in informatica, da approfondire mediante la consultazione di articoli scientifici e letteratura specializzata.

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

Articoli scientifici e note fornite dal docente.

ADVANCED TOPICS IN PROGRAMMING LANGUAGES

Titolare: Prof.ssa SILVIA CRAFA

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Conoscenze di programmazione e di programmazione ad oggetti.

Conoscenze e abilità da acquisire:

Il corso presenta alcune tecniche avanzate dei moderni linguaggi di programmazione. Lo studente svilupperà la capacità di comprendere, ragionare e valutare alcune delle nuove tecniche di programmazione e svilupperà la capacità di analisi critica del ruolo della tecnologia nella società contemporanea.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali con esercizi ed approfondimenti di argomenti di ricerca tramite articoli scientifici.

Contenuti:

Il corso presenta alcune tecniche avanzate dei moderni linguaggi di programmazione, tra cui: l'uso dei sistemi di tipi per ragionare sui programmi, concetti avanzati di programmazione orientata agli oggetti (typing strutturale, type checking dinamico, mixins), l'integrazione della programmazione funzionale con la programmazione ad oggetti. Questi concetti saranno affrontati secondo l'approccio dei metodi formali e relativo rigore logico, e visti all'opera in uno studio ragionato dei linguaggi Scala, Java8 e Rust. Nella parte finale del corso gli studenti saranno esposti all'analisi e alla discussione di alcuni casi concreti di riflessione sull'impatto sociale delle nuove tecnologie.

Modalità di esame:

Sono previste una prova scritta e una seconda prova che consiste nella discussione orale di un tema di approfondimento o in alternativa nella realizzazione di un progetto software.

Criteri di valutazione:

La prova scritta valuta l'acquisizione dello studente degli aspetti fondamentali affrontati durante il corso (dimostrazioni formali, tecnica di induzione). La seconda prova valuta la capacità dello studente di analizzare e valutare aspetti avanzati delle tecnologie informatiche e del loro impatto sulla società.

Testi di riferimento:

M. Odersky, L. Spoon, B. Venners, Programming in Scala. : Artima, 2016 B.C. Pierce, Types and Programming Languages. : The MIT Press, 2002

ARTIFICIAL INTELLIGENCE

Titolare: Prof. ALESSANDRO SPERDUTI

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+8L; 6,00

Prerequisiti:

E' opportuno possedere le conoscenze di base relative al Calcolo delle Probabilità, alla Programmazione e agli Algoritmi.

Conoscenze e abilità da acquisire:

In questo insegnamento si presentano le tecniche fondamentali di alcuni degli approcci principali, all'interno della Intelligenza Artificiale, per la soluzione di problemi difficili. In particolare sono esaminate tecniche di Ricerca in uno Spazio di Soluzioni, di Ricerca con Avversari, di Rappresentazione e Manipolazione di Conoscenza con e senza incertezza, e le basi di Apprendimento Automatico, Sistemi con vincoli, Visione Artificiale, Analisi del Linguaggio Naturale. E' previsto lo sviluppo da parte del singolo studente, o di un gruppo di studenti, di un progetto di dimensioni ridotte.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali o in remoto nel caso si renda necessario per motivi di sicurezza sanitaria.

Contenuti:

La struttura e le tematiche dell'insegnamento saranno le seguenti: - Introduzione, Motivazioni, Architetture di Agenti Intelligenti; - Risoluzione di Problemi e Cenni di Sistemi con Vincoli; - Ricerca con avversari; - Trattamento della conoscenza tramite logica preposizionale e del primo ordine; - Trattamento dell'Incertezza e Ragionamento Probabilistico; - Basi di Apprendimento Automatico; - Basi di Visione Artificiale; - Basi di Elaborazione del Linguaggio Naturale.

Modalità di esame:

Lo studente deve superare un esame scritto. Inoltre lo studente deve sviluppare un progetto applicativo concordato con il docente.

Criteri di valutazione:

La valutazione dello studente si basa su una verifica dell'apprendimento dei concetti di base introdotti durante il corso e sulla capacità di analisi dello studente. La valutazione del progetto considera la capacità, da parte dello studente, di individuare un caso di studio adeguato e di svolgere in modo autonomo un'attività di progettazione e realizzazione qualitativamente appropriata.

Testi di riferimento:

Stuart Russell, Peter Norvig, Artificial Intelligence: A modern approach. : Prentice Hall, 2010

Eventuali indicazioni sui materiali di studio:

Materiale aggiuntivo sarà disponibile sul sito e-learning del corso.

BIG DATA COMPUTING

Titolare: Prof. ANDREA ALBERTO PIETRACAPRINA

Mutuato da: Scuola Galileiana di Studi Superiori - Classe di Scienze Naturali

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Il corso ha i seguenti prerequisiti: competenze relative al progetto e all'analisi di algoritmi e strutture dati, conoscenza delle nozioni fondamentali di calcolo delle probabilità e statistica, e capacità di programmazione in Java o Python.

Conoscenze e abilità da acquisire:

In questo corso gli studenti imparano tecniche algoritmiche fondamentali per l'elaborazione efficiente ed efficace di insiemi di dati di grande dimensione. Inoltre, attraverso alcune attività pratiche, essi acquisiscono abilità relative allo sviluppo di applicazioni in Apache Spark, che è uno dei framework di programmazione più popolari e diffusi per big data computing.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali, uso di piattaforme di students engagement, seminari di esperti selezionati, e attività propedeutiche allo svolgimento degli homework.

Contenuti:

La valutazione finale è basata sui seguenti argomenti: Introduction to the Big Data phenomenon. Programming frameworks: MapReduce, Apache Spark Reducing input size (Case study: clustering) Reducing output size (Case study: frequent itemsets) Streaming framework.

Modalità di esame:

L'esame consiste in alcuni homework di programmazione, assegnati ogni 2-3 settimane e da svolgere in gruppi di 2-3 studenti, e in una prova scritta individuale comprendente domande teoriche ed esercizi.

Criteri di valutazione:

La valutazione finale è basata sugli homework e sulla prova scritta. Gli homework mirano a verificare la capacità degli studenti di programmare applicazioni big data in Apache Spark, mentre la prova scritta mira a verificare la loro conoscenza delle tecniche algoritmiche apprese durante il corso e la loro capacità di problem solving nel contesto big data.

Testi di riferimento:

J. Leskovec, A. Rajaraman and J. Ullman, Mining Massive Datasets. : Cambridge University Press, 2014

Eventuali indicazioni sui materiali di studio:

Il diario delle lezioni, il materiale didattico e le modalità d'esame dettagliate sono resi disponibili sul MOODLE del corso e sul MOODLE esami.

BIOINFORMATICS

Titolare: Prof. GIORGIO VALLE

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+8E; 6,00

Prerequisiti:

Non ci sono prerequisiti particolari, se non quanto ci si aspetta da uno studente magistrale di informatica. Una conoscenza di base della genetica e della biologia molecolare saranno comunque utili per meglio inquadrare le motivazioni biologiche che stanno alla base della bioinformatica. Il corso è in lingua inglese, quindi è necessario avere una buona conoscenza dell'inglese scritto e parlato.

Conoscenze e abilità da acquisire:

Il Corso è suddiviso in tre parti principali: la prima parte mette in relazione Biologia e Informazione; la seconda parte descrive i principali algoritmi utilizzati in bioinformatica per allineare sequenze biologiche e assemblare genomi; la terza parte tratta di problemi di bioinformatica relativi alla genomica funzionale. Inoltre il corso è accompagnato da esercitazioni pratiche in cui gli studenti applicheranno metodi bioinformatici per analizzare dati genomici. In considerazione della complessità della materia e in accordo con i descrittori di Dublino, particolare attenzione sarà dedicata affinché gli studenti acquisiscano la capacità di integrare le conoscenze e gestire la complessità dei problemi trattati, nonché di formulare giudizi sulla base di informazioni limitate e spesso frammentarie.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso sarà tenuto con lezioni frontali e con esercitazioni pratiche. L'attività didattica sarà supportata da risorse messe a disposizione sulla piattaforma e-learning "Moodle", che comprendono materiale per apprendimento remoto e per auto-valutazione. In questo modo si vuole promuovere un'attività di "blended learning" con cui lo studente, almeno in parte, impara autonomamente, seguendo un percorso che lo accompagna attraverso contenuti reperibili in rete. Dove possibile si applicherà il paradigma della "flipped classroom" che inverte lo schema tradizionale di insegnamento, prevedendo che prima lo studente impari la lezione autonomamente per poi discutere e approfondire gli argomenti in classe, con il docente e con gli altri studenti. Un'ampia raccolta di problemi, questionari ed esercizi viene messa a disposizione sulla piattaforma Moodle, sia per consentire l'autovalutazione, sia per stimolare argomenti di discussione da approfondire in classe.

Contenuti:

Questo è un corso di 6 crediti: cinque di lezioni ed uno di attività pratiche che consistono nell'implementazione di algoritmi oppure in un'approfondita indagine della letteratura, su argomenti assegnati. Le lezioni sono organizzate in tre parti. La prima parte è un'approfondita introduzione alla Biologia, presentata come una disciplina scientifica centrata sull'informazione. I meccanismi che facilitano la trasmissione e l'evoluzione dell'informazione biologica saranno presi come spunto per introdurre alcuni problemi della biologia che richiedono approcci computazionali e strumenti bioinformatici. La seconda parte del corso descrive i principali algoritmi utilizzati per allineare sequenze biologiche, inclusi quelli sviluppati per il sequenziamento di DNA di ultima generazione. Sono inoltre descritti gli algoritmi utilizzati per l'assemblaggio "de novo" di genomi. Infine, la terza parte del corso copre alcuni aspetti della bioinformatica relativi alla genomica funzionale, come l'analisi del trascrittoma, la predizione e annotazione genica, la ricerca di pattern e motivi per la predizione delle strutture proteiche. Inoltre viene discusso il ruolo della bioinformatica nell'analisi di genomi individuali e nella medicina personalizzata.

Modalità di esame:

L'esame si articola in tre parti: 1) una sessione pratica nella quale lo studente deve descrivere un progetto di analisi di dati che deve essere consegnato almeno due giorni prima della data dell'esame, 2) una sessione di quiz su Moodle, che si svolgerà all'inizio dell'appello d'esame e 3) una discussione orale in cui lo studente deve descrivere il progetto che ha realizzato e rispondere a domande sui contenuti del corso. Un continuo monitoraggio sarà attuato durante l'intera durata del corso per verificare la comprensione degli studenti.

Criteri di valutazione:

Nell'esame finale gli studenti dovranno dimostrare una comprensione sistematica del settore e dovranno sapersi destreggiare con i metodi della ricerca associati ad esso. Inoltre gli studenti dovranno essere capaci di analisi critica, di valutare e sintetizzare idee nuove e complesse, integrando gli argomenti di questo corso con altre conoscenze.

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

Non sono previsti libri ufficiali di testo e gli studenti saranno stimolati a trovare le informazioni su fonti multiple. Il materiale didattico sarà messo a disposizione sulla piattaforma e-learning Moodle.

COMPUTABILITY

Titolare: Prof. PAOLO BALDAN

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Il corso richiede familiarità con alcuni concetti matematici di base, quali relazioni, funzioni, insiemi, cardinalità, ordini parziali, principi di induzione. Non ci sono corsi propedeutici.

Conoscenze e abilità da acquisire:

Obiettivo del corso è quello di avvicinare lo studente ai temi classici della teoria della calcolabilità, completando e approfondendo alcune conoscenze algoritmiche acquisite nella laurea di primo livello. Partendo dall'esame matematico del concetto di procedimento effettivo, si studiano i limiti che tale nozione impone sulla classe delle funzioni effettivamente calcolabili da un algoritmo, con lo sviluppo di una teoria dell'indecidibilità e della ricorsione.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso prevede lezioni frontali ed esercizi.

Contenuti:

Saranno sviluppati i seguenti temi: - Algoritmi ed il concetto di procedimento effettivo. Macchine a registri (URM). Funzioni parziali ricorsive. Equivalenze tra modelli di calcolo. Universalità dei modelli di calcolo. Tesi di Church. - Enumerazione delle funzioni calcolabili. Esistenza di funzioni non calcolabili: il metodo della diagonalizzazione. Il teorema del parametro. Programmi universali. - Problemi decidibili, indecidibili e semidecidibili. Indecidibilità del problema della fermata. Metodo di riduzione. Esempi di altri problemi indecidibili. - Insiemi ricorsivi e ricorsivamente enumerabili. Teoremi di Rice e di Rice-Shapiro. - Funzionali. Definizioni ricorsive. Ordinamenti parziali, funzioni monotone e punti fissi. Funzionali ricorsivi. Il teorema di Myhill-Shepherdson. Primo teorema di ricorsione. Secondo teorema di ricorsione.

Modalità di esame:

Esame scritto e prova orale.

Criteri di valutazione:

La prova scritta contiene esercizi atti a verificare la capacità dello studente di utilizzare nozioni e tecniche dimostrative apprese durante il corso, per la soluzione di problemi nuovi.

Testi di riferimento:

Nigel Cutland, Computability: An Introduction to Recursive Function Theory. : Cambridge University Press, 1980

Eventuali indicazioni sui materiali di studio:

Pagina web: <http://www.math.unipd.it/~baldan/Computabilita>

CRYPTOGRAPHY

Titolare: Prof. ALESSANDRO LANGUASCO

Mutuato da: Laurea magistrale in Cybersecurity (Ord. 2020)

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+8E; 6,00

Prerequisiti:

Per la prima parte (Prof. Languasco; 6 CFU): Gli argomenti dei corsi di Algebra (congruenze, gruppi e gruppi ciclici, campi finiti), Analisi I (calcolo differenziale ed integrale, serie numeriche) del corso di studi in Matematica. Per la seconda parte (Prof. Conti and Prof. Migliardi; 6 CFU): OS, Programming.

Conoscenze e abilità da acquisire:

Per la prima parte (Prof. Languasco; 6 CFU): Lo scopo della prima parte del corso e' quello di offrire una panoramica delle basi teoriche necessarie per permettere uno studio critico dei protocolli crittografici usati oggi in molte applicazioni (autenticazione, commercio digitale). Nella prima parte verranno esposti gli strumenti matematici di base (essenzialmente dalla teoria elementare ed analitica dei numeri) necessari per comprendere il funzionamento dei moderni metodi a chiave pubblica. Nella seconda parte vedremo come applicare queste conoscenze per studiare in modo critico alcuni protocolli crittografici. For the second part (Prof. Conti; 3 CFU): Students will be able to identify, classify, describe, explain, and correlated the key concepts of cybersecurity attacks and defenses. For the second part (Prof. Migliardi; 3 CFU): Assess the risks to which an IT system is exposed, Explain how an attack works, Describe, explain and generalize software vulnerabilities, Avoid software pitfalls.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali in classe, se possibile. In presenza di una continuata emergenza sanitaria, mediante teleconferenza.

Contenuti:

Per la prima parte (Prof. Languasco; 6 CFU): First Part: Basic theoretical facts: Modular arithmetic. Prime numbers. Little Fermat theorem. Chinese remainder theorem. Finite fields: order of an element and primitive roots. Pseudoprimality tests. Agrawal-Kayal-Saxena's test. RSA method: first description, attacks. Rabin's method and its connection with the integer factorization. Discrete logarithm methods. How to compute the discrete log in a finite field. Elementary factorization methods. Some remarks on Pomerance's quadratic sieve. Protocols and algorithms. Fundamental crypto algorithms. Symmetric methods (historical ones, DES, AES) . Asymmetric methods. Attacks. Digital signature. Pseudorandom generators (remarks). Key exchange, Key exchange in three steps, secret splitting, secret sharing, secret broadcasting, timestamping. Signatures with RSA and discrete log. Per la seconda parte (Prof. Conti and Prof. Migliardi; 6 CFU): Introduction to Cybersecurity, User Authentication, Access Control, Database Security, Malicious Software, Denial-of-Service Attacks, Intrusion Detection, Firewalls and Intrusion Prevention Systems, Operating System Security, Trusted Computing and Multilevel Security. The execution environment of a program and the vulnerabilities resulting from the threat model of the time. Languages and threat models. Control hijacking: attack. Control hijacking: defense. Security of operating systems and principle of least privilege necessary (and examples of privilege escalation). Sandboxing and interaction with legacy code. Flaw search techniques.

Modalità di esame:

Per la prima parte (Prof. Languasco; 6 CFU): Esame scritto svolto in presenza se possibile; se non possibile per questioni sanitarie, esame scritto svolto in teleconferenza. Per la seconda parte (Prof. Conti and Prof. Migliardi; 6 credits): Esame scritto, progetti assegnati da svolgere a casa, esame orale.

Criteri di valutazione:

Per la prima parte (Prof. Languasco; 6 CFU): Durante la prova scritta lo studente dovrà rispondere ad alcune domande relative al programma svolto dimostrando di aver compreso gli argomenti del corso. Il massimo dei voti (30/30) verrà assegnato in presenza di un compito privo di errori. Il docente si riserva di fare alcune domande orali nel caso in cui sia necessario investigare ulteriormente la preparazione del candidato. L'orale sarà in presenza se possibile o in videoconferenza se l'emergenza sanitaria sarà ancora presente. Per la seconda parte (Prof. Conti and Prof. Migliardi; 6 CFU): Evaluation of both theoretical competence and operational ability to apply what has been learned to a real case.

Testi di riferimento:

Pfleeger, Charles P.; Pfleeger, Shari Lawrence, Security in Computing. : Prentice Hall; 5 edition, 2015 Wenliang Du, Computer Security: a hands-on approach. : Create Space Independent Publishing Platform, 1 ed, 2017 Stallings, William; Brown, Lawrie, Computer security principles and practice. Boston [etc.]: Pearson, 2015 Knospe, Heiko, A Course in Cryptography. Providence: American Mathematical Society, 2019 A. Languasco; A. Zaccagnini, Manuale di Crittografia. Milano: Hoepli, 2015

Eventuali indicazioni sui materiali di studio:

Per la prima parte (6 CFU): Utilizzeremo i seguenti testi: 1) A.Languasco, A.Zaccagnini - Manuale di Crittografia - Hoepli Editore, 2015. (italian). 2) N. Koblitz - A Course in Number Theory and Cryptography -Springer, 1994. 3) H. Knospe - A Course in Cryptography - American Mathematical Society, 2019. 4) R. Crandall, C.Pomerance - Prime numbers: A computational perspective - Springer, 2005. 5) B. Schneier - Applied Cryptography - Wiley, 1994.

DATA MINING

Titolare: Prof.ssa ANNAMARIA GUOLO

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 34A+16L; 6,00

Prerequisiti:

Conoscenze di Informatica di base, Basi di Dati. Conoscenze di base di Probabilità e Statistica sono utili anche se non indispensabili.

Conoscenze e abilità da acquisire:

Ci si attende che lo studente acquisisca le seguenti conoscenze e abilità: - padronanza dei principi di data mining per l'analisi di insiemi di dati anche ad elevata dimensionalità; - costruzione di modelli appropriati per l'analisi di insiemi di dati e la previsione; - analisi dei dati tramite il software R, considerando sia analisi grafiche sia analisi di modellazione; - valutazione ed interpretazione critica dei risultati ottenuti; - capacità di comunicazione delle analisi dei dati effettuate e dei risultati conseguiti.

Attività di apprendimento previste e metodologie di insegnamento:

L'attività prevede 1) lezioni frontali, nelle quali vengono illustrati i contenuti del corso tramite l'utilizzo di slides relative agli aspetti teorici e all'analisi di insiemi di dati, con l'obiettivo di promuovere la discussione e la riflessione critica in aula; 2) esercitazioni in laboratorio informatico, per introdurre gli studenti all'analisi di insiemi di dati reali tramite l'utilizzo del software R.

Contenuti:

- Introduzione al corso: l'analisi dei dati come strumento di supporto per le decisioni. Motivazioni e contesto per il data mining. - Modello lineare semplice e multiplo: stima, intervalli di confidenza, test di verifica d'ipotesi, livello di significatività osservato, previsione, selezione del modello, analisi dei residui, correlazione spuria, multicollinearità - Metodi di classificazione: regressione logistica, analisi discriminante lineare ed estensioni - Criteri per la selezione del modello: validazione incrociata, R2 aggiustato, AIC, BIC, selezione automatica - Regolarizzazione: regression ridge e lasso - Regressione con componenti principali - Regressione semiparametrica: splines di regressione, splines di liscio, modelli additivi generalizzati

Modalità di esame:

L'esame consiste in due prove scritte. 1) La prima prova si svolge in laboratorio informatico (durata 35 minuti) riguarda il modello di regressione lineare e consiste in una serie di domande a risposta multipla e a risposta aperta da svolgersi su Moodle. Le domande si riferiscono all'analisi di un insieme di dati reali e comprendono analisi numeriche, interpretazione di output da analisi effettuate in R e commenti di analisi grafiche. La prima prova si svolgerà dopo la prima metà del corso. Durante la prova lo studente può consultare il materiale del corso (libro di testo, slides, dispense di laboratorio). 2) La seconda prova si svolge in laboratorio informatico (durata 2 ore) ed è costituita dall'analisi di un insieme di dati reali tramite il software R. Si richiede di riportare i principali risultati dell'analisi condotta, con adeguati commenti, in un breve report. Durante la prova lo studente può consultare il materiale del corso (libro di testo, le slides, le dispense di laboratorio). La valutazione finale della prova è una media dei risultati delle due prove scritte. Gli studenti che non effettueranno la prima prova a metà del corso potranno sostenerla immediatamente dopo la seconda prova in laboratorio.

Criteri di valutazione:

La prova d'esame d'esame ha lo scopo di 1) valutare le conoscenze acquisite relativamente alla costruzione e selezione di un modello di regressione lineare ed alla interpretazione critica dei risultati grafici e analitici conseguiti; 2) valutare le conoscenze acquisite relativamente all'applicazione di tecniche di modellazione appropriate per l'analisi di insieme di dati reali di natura diversa e per la previsione, con attenzione al caso di dati ad alta dimensionalità; 3) valutare le capacità di utilizzo delle funzionalità del software R per condurre una completa analisi di insiemi di dati reali; 4) valutare le capacità di interpretazione e comunicazione dei risultati dell'analisi di un insieme di dati reali.

Testi di riferimento:

James, Gareth; Witten, Daniela; Hastie, Trevor; Tibshirani, Robert, An introduction to statistical learning: with applications in R. New York: Springer, 2013

Eventuali indicazioni sui materiali di studio:

Libro di testo. Sulla piattaforma Moodle sarà reso disponibile del materiale aggiuntivo, che include 1) le slides del corso; 2) le dispense di laboratorio informatico con R; 3) articoli e note dalla letteratura statistica e di data mining.

DEEP LEARNING

Titolare: Prof. ALESSANDRO SPERDUTI

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

E' opportuno possedere le conoscenze di base relative al Calcolo delle Probabilità, alla Programmazione e agli Algoritmi.

Conoscenze e abilità da acquisire:

L'insegnamento introduce i concetti di base relativi al Deep Learning, cioè all'apprendimento automatico tramite reti neurali. Verranno richiamati i concetti matematici necessari per una piena comprensione della materia. Si tratteranno le reti neurali feedforward deep e le relative tecniche di regolarizzazione e di ottimizzazione dell'apprendimento. Verranno introdotti i concetti di base relativi alle reti convolutive. Per quanto riguarda il trattamento di sequenze, saranno presentate le reti neurali ricorrenti, con particolare enfasi all'utilizzo di unità LSTM e analoghe. Infine si tratteranno autoencoder e modelli generativi deep. Inoltre, per quanto riguarda l'implementazione dei modelli trattati nel corso, si introdurrà la piattaforma TensorFlow.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali o in remoto nel caso si renda necessario per motivi di sicurezza sanitaria.

Contenuti:

La tematica dell'insegnamento saranno le seguenti: - Introduzione ai contenuti dell'insegnamento; - Reti Neurali Feedforward profonde (deep); - Regolarizzazione per l'apprendimento deep; - Ottimizzazione per l'apprendimento di modelli deep; - Concetti di base per reti neurali convolutive; - Reti neurali ricorrenti e Transformers per la modellazione di sequenze; - Autoencoder; - Modelli generativi deep; - TensorFlow.

Modalità di esame:

Lo studente deve superare un esame scritto. Inoltre lo studente deve sviluppare un notebook concordato con il docente.

Criteri di valutazione:

La valutazione dello studente si basa su una verifica dell'apprendimento dei concetti di base introdotti durante il corso e sulla capacità di analisi dello studente. La valutazione del progetto considera la capacità, da parte dello studente, di individuare un caso di studio adeguato e di svolgere in modo autonomo un'attività di progettazione e realizzazione qualitativamente appropriata.

Testi di riferimento:

Eventuali indicazioni sui materiali di studio:

Materiale aggiuntivo sarà disponibile sul sito e-learning del corso.

ECONOMICS AND MANAGEMENT OF INNOVATION

Titolare: Prof.ssa KATIA CALDARI

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Nessun prerequisito richiesto

Conoscenze e abilità da acquisire:

Il corso si propone di fornire gli elementi conoscitivi di base per comprendere il ruolo dell'innovazione nel processo economico capitalistico. A tal fine saranno prese in esame le principali innovazioni che hanno segnato la storia del sistema capitalistico e le teorie economiche che hanno evidenziato l'importanza e il ruolo delle innovazioni soprattutto per i processi di crescita e sviluppo economici. Saranno poi analizzati il processo di creazione, diffusione e gestione delle innovazioni a livello di impresa e il loro ruolo nel contesto competitivo.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali e attività di discussione in aula su argomenti specifici. Saranno inseriti sulla piattaforma Moodle (con libero accesso da parte dello studente) dispense ed altro materiale di approfondimento, considerati parte del programma d'esame (e sono quindi da studiare) assieme la manuale indicato.

Contenuti:

il corso si propone di affrontare e approfondire i seguenti temi: - Le principali innovazioni che hanno influenzato l'evoluzione del sistema capitalistico. - L'innovazione nelle teorie economiche (Smith, Ricardo, Marx, la teoria neoclassica, Schumpeter, Nelson e Winter) - fonti dell'innovazione - tipologie di innovazione - innovazione e strategie d'impresa - innovazione e concorrenza

Modalità di esame:

L'esame finale sarà scritto e composto da domande aperte. Per gli studenti frequentanti si terrà anche conto delle loro presentazioni in aula (tale presentazioni sono facoltative)

Criteri di valutazione:

Gli studenti frequentanti saranno valutati sugli argomenti svolti in aula e sulle loro presentazioni di approfondimento tematico che saranno svolte nel corso delle lezioni. Tali presentazioni saranno su base volontaria. Gli studenti non frequentanti saranno valutati sulla base della comprensione del manuale e degli articoli messi a disposizione sulla piattaforma moodle.

Testi di riferimento:

Melissa A. Schilling, Strategic Management of Technological Innovation. New York: McGraw-Hill, 2020

Eventuali indicazioni sui materiali di studio:

Dispense ed altro materiale di approfondimento, considerati parte del programma d'esame (e sono quindi da studiare) saranno resi disponibili sulla piattaforma Moodle durante il corso

ENGLISH LANGUAGE B2 (PRODUCTIVE SKILLS)

Titolare: Prof. MICHELE SCQUIZZATO

Periodo: I anno, annuale

Indirizzo formativo: Corsi comuni

Tipologie didattiche: ; 3,00

FORMAL METHODS FOR CYBER-PHYSICAL SYSTEMS

Titolare: Prof. DAVIDE BRESOLIN

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Il corso richiede familiarità con alcuni concetti matematici e informatici di base, quali teoria degli automi e della computabilità, analisi matematica. Non ci sono corsi propedeutici.

Conoscenze e abilità da acquisire:

Un sistema cyber-fisico consiste in una collezione di dispositivi informatici in grado di interagire in modo continuo con il mondo fisico tramite sensori e attuatori. Tali sistemi sono sempre più diffusi nelle società moderne, dagli edifici intelligenti ai dispositivi medici alle automobili. Questo corso offre un'introduzione ai principi di progettazione, specifica, modellazione e analisi dei sistemi ciberfisici, fornendo le seguenti conoscenze e competenze: 1. Capacità di modellare un sistema ciberfisico. 2. Capacità di formulare le proprietà che il sistema dovrebbe rispettare in modo matematicamente rigoroso. 3. Capacità di progettare e implementare un algoritmo di verifica per i sistemi ciberfisici. 4. Capacità di eseguire l'implementazione su un test case e di

comprenderne e analizzarne i risultati.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso è suddiviso in blocchi tematici, ognuno dei quali affronta un problema reale e lo risolve seguendo l'approccio del pensiero algoritmico. Ogni blocco inizia con una serie di lezioni frontali in aula durante le quali vengono affrontati i contenuti del corso. Terminata l'attività frontale, il blocco prosegue con una o più lezioni di laboratorio dove gli studenti divisi a gruppi implementeranno e testeranno la soluzione del problema reale su un dataset di media dimensione, e si conclude con una fase di confronto e discussione delle varie soluzioni realizzate.

Contenuti:

Sistemi ciberfisici: definizione e caratteristiche chiave. Modelli formali per sistemi ciberfisici: modelli sincroni e asincroni, modelli temporizzati e ibridi. Analisi dei sistemi ciberfisici: proprietà di sicurezza e vivacità, sistemi dinamici e proprietà di controllo.

Modalità di esame:

Esame orale e/o progetto.

Criteri di valutazione:

I criteri di valutazione sono i seguenti: 1. Completezza delle conoscenze acquisite; 2. Proprietà della terminologia tecnica utilizzata; 3. Capacità di modellare un sistema ciberfisico e le proprietà desiderate 3. Capacità di utilizzare strumenti di verifica formale per i sistemi ciberfisici 4. Capacità di progettare e implementare algoritmi di verifica per sistemi ciberfisici

Testi di riferimento:

Alur, Rajeev, Principles of cyber-physical systems. Cambridge: MS, MIT, 2015

Eventuali indicazioni sui materiali di studio:

Il corso ha una sezione dedicata sul Moodle del Dipartimento di Matematica. Il Moodle raccoglierà le dispense del corso, le specifiche dettagliate delle attività di laboratorio, gli esercizi e le loro soluzioni. Verrà usato anche per comunicazioni e aggiornamenti da parte del Docente.

FUNCTIONAL LANGUAGES

Titolare: Dott. ALVISE SPANO'

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+10L; 6,00

Prerequisiti:

Programmazione imperativa ed orientata agli oggetti

Conoscenze e abilità da acquisire:

Apprezzare il paradigma funzionale di programmazione. In particolare viene studiato il linguaggio Haskell. Gli studenti imparano ad apprezzare l'importanza dei tipi come fondamentale mezzo per scoprire gli errori e imparano concetti quali il polimorfismo parametrico e la valutazione lazy. Viene introdotto anche il concetto di Monade e il suo uso per la programmazione generica e modulare. Imparano anche a capire la gestione run time dei programmi.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso prevede 48 ore totali. Di queste, circa 35 sono di lezioni frontali. Le restanti ore sono impiegate per la soluzione interattiva di esercizi e per la presentazione e discussione del progetto.

Contenuti:

Nel corso viene illustrato il linguaggio funzionale Haskell. In particolare si studiano i seguenti aspetti: Pattern matching. Curryficazione e funzioni di ordine superiore. Inferenza di tipo: cos'è e come viene fatta. Input e output. Polimorfismo parametrico. Valutazione pigra. Funtori, funtori applicativi e monadi. Supporto run-time. Parsing con le Monadi.

Modalità di esame:

L'esame consiste di una parte scritta (80% del voto finale) e di una orale (20% del voto finale). L'esame scritto verte sulle nozioni generali e gli esercizi visti nel corso, mentre la parte orale è una discussione sul progetto che consiste nella realizzazione di un parser per un linguaggio funzionale.

Criteri di valutazione:

L'esame mira a valutare il grado di comprensione dei concetti insegnati, raggiunto dagli studenti, e la capacità di usare questi concetti per affrontare problemi relativi ai linguaggi funzionali e al loro uso per definire programmi modulari e generici.

Testi di riferimento:

Miran Lipovaca, Learning you a Haskell for great good. : , Graham Hutton, Programming in Haskell. : Cambridge University Press,

GAME THEORY

Titolare: ELVINA GINDULLINA

Mutuato da: Scuola Galileiana di Studi Superiori - Classe di Scienze Naturali

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Un corso anche basilare di teoria della probabilità.

Conoscenze e abilità da acquisire:

L'insegnamento prevede l'acquisizione delle seguenti conoscenze e abilità, suddivise in due insiemi. 1: parte base. Apprendere e padroneggiare concetti teorici di base e avanzati della teoria dei giochi e saper risolvere problemi generali multi-obiettivo multi-agente con tecniche della teoria dei giochi. 2: parte applicativa. Sapere applicare i concetti della teoria dei giochi a scenari pratici, specialmente di tipo ICT; in questo contesto, e' di particolare interesse l'abilità di contestualizzare la teoria dei giochi come strumento di valutazione per l'efficacia della risoluzione tramite procedure multi-agente distributed.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni convenzionali con il supporto di slide. Prevista interazione su piattaforma moodle. Caricamento video via Kaltura.

Contenuti:

Concetti base di teoria dei giochi Utilità, mercato, fattore di sconto Giochi statici in forma normale Dominanza, Equilibri di Nash Efficienza, prezzo dell'anarchia Giochi a somma zero, giochi minimax Strategie miste, equilibri misti Teorema di Nash, il teorema minimax The tragedy of the commons Giochi dinamici Strategie e sottogiochi Backward utility Equilibri di Stackelberg Giochi ripetuti, collaborazione Duopoli dinamici, collusione Cooperazione, pricing Informazione incompleta/imperfetta Giochi bayesiani, signaling, beliefs Principio di rivelazione Teoria dei giochi assiomatica Fictitious play Best response dynamics Ottimizzazione distribuita Game theory algoritmica Calcolo, complessità, e completezza dell'equilibrio Aste, bargaining Aste di primo e secondo prezzo Criterio VCG Giochi cooperativi, il nucleo, il valore di Shapley Allocazione delle risorse Utilità, scelte e paradossi Giochi potenziali, coordinazione Algoritmi bio-inspired Giochi evolutivi Reti cognitive Selfish routing Sistemi multi-input con teoria dei giochi

Modalità di esame:

In qualunque caso l'esame comprende un test scritto obbligatorio a libro aperto, dove vengono sottoposti diversi problemi di game theory allo studente su argomenti toccati durante il corso. Per ogni esercizio, vengono poste piu' domande, tipicamente tre. Per frequentanti, l'esame puo' coinvolgere lo sviluppo di un progetto in gruppi di 1-3 persone, su argomenti del corso applicati alle ICT. L'adesione a questa modalita' e l'argomento del progetto sono concordati con il docente durante il corso. Se il test scritto e' sufficiente, si puo' registrare il voto conseguito come voto finale dell'esame. Si puo' ulteriormente discutere il progetto sviluppato durante il corso con un esame orale, da svolgersi dopo l'esame scritto. Questi esami orali si svolgono nella stessa giornata di un esame scritto, ma non necessariamente bisogna presentarsi nella stessa giornata per l'esame scritto e la discussione orale del progetto. La discussione orale integra il voto dello scritto.

Criteri di valutazione:

Ogni domanda nei test scritti viene valutata fino a un massimo di 3 punti. La discussione del progetto viene valutata fino a 10 punti. Il voto finale e' la somma numerica dei punteggi individuali delle domande e della discussione del progetto (se presente), limitata a 30. Un punteggio di 30 e lode e' assegnato agli studenti il cui punteggio numerico e' superiore a 31. Nella valutazione di ogni domanda scritta vengono tenuti in considerazione: - la pertinenza, la correttezza, e la completezza della risposta; - l'utilizzo appropriato delle terminologie, metodologie, e rappresentazioni formali tipiche della teoria dei giochi - l'acquisita capacita' di problem solving - la capacita' di discussione e verifica ex-post della soluzione trovata Nella valutazione del progetto (se presente) vengono tenuti in considerazione: - l'originalita' della proposta e la pertinenza sia con le tematiche del corso che con le metodologie ingegneristiche tipiche dell'ICT - la qualita' dell'esposizione orale - la capacita' di lavoro di gruppo e la presenza di singoli contributi attribuibili ai partecipanti al progetto - la capacita' di trarre conclusioni significative dal punto di vista scientifico grazie alle metodologie apprese nel corso

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

Diversi libri forniscono una trattazione generale di teoria dei giochi. A mero titolo di suggerimento, si può usare il libro di Tadelis come riferimento in senso generale. Questa parte comunque dovrebbe essere integrata con materiale per le applicazioni. Il libro di MacKenzie e DaSilva è un buon esempio, anche se non è obbligatorio usare un libro per questo scopo (si può fare riferimento anche a materiale trovato in rete). In ogni caso, il docente fornirà agli studenti tutte le dispense delle lezioni e appunti aggiuntivi.

IT SERVICE MANAGEMENT

Titolare: Dott. FRANCESCO CLABOT

Periodo: 1 anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+8E; 6,00

Prerequisiti:

Il corso non prevede particolari prerequisiti.

Conoscenze e abilità da acquisire:

Il corso si propone di presentare agli studenti l'organizzazione di un dipartimento ICT di una grande azienda. In particolare, il corso tratta le tematiche legate alle metodologie consolidate per l'impostazione dei processi ICT (ITIL), le motivazioni che sono alla base delle scelte dei prodotti e tecnologie adottate (ROI, SLA, etc.), esempi concreti di architetture informatiche basilari oltre a vari case studies. Livello Bloom's taxonomy: 1-2 (Knowledge-Comprehension) per il corso, 3-4 (Application-Analysis) per le esercitazioni e laboratori.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali, esercizi in classe e discussione sui casi di studio proposti.

Contenuti:

Il corso descrive i processi ed i ruoli che permettono al dipartimento IT di una grossa azienda di funzionare: l'approccio ai problemi, le responsabilità associate ad ogni ruolo, i compiti e come questi debbano venire svolti. Nel corso vengono trattate le relazioni che è necessario instaurare e gestire non solo all'interno dell'azienda, ma anche con i clienti ed i fornitori, attraverso analisi e molti casi reali. Alla base del corso, la metodologia ITSM denominata "ITIL" (IT Infrastructure Library) oramai diventata un MUST per l'organizzazione aziendale, un insieme di practices per la gestione e l'erogazione dei servizi IT. L'obiettivo, oltre al fornire conoscenze specifiche tratte da ITIL tanto da mettere gli studenti in condizione di superare agevolmente il livello di certificazione "ITIL Foundation", vuole insegnare anche cosa è un reparto IT, quale sia la sua importanza all'interno delle aziende di oggi e di come la sua attività sia profondamente integrata nel processo produttivo. E', quindi, necessario trasmettere agli studenti quale sia lo schema mentale necessario per poter operare nell'ambito IT, con particolare attenzione alla cultura dell'organizzazione, alla prevenzione dei problemi, alla gestione ed al controllo delle attività, tutti ambiti fondamentali per poter svolgere un lavoro efficace e professionale. Durante il corso i concetti espressi saranno affiancati da numerosi casi di studio reali (esempi) molto concreti tratti anche da esperienze dirette del docente.

Modalità di esame:

Test a scelta multipla (crochette) e produzione di una piccola tesina su un caso reale.

Criteri di valutazione:

La prova scritta valuta l'acquisizione dello studente degli aspetti fondazionali affrontati durante il corso [Livello Bloom's taxonomy: 1-2 (Knowledge-Comprehension)]. La seconda prova valuta la capacità dello studente di analizzare e valutare aspetti concreti dalla metodologia insegnata e la loro applicazione nei casi di vita reali [Livello Bloom's taxonomy: 3-4 (Application-Analysis)].

Testi di riferimento:

Pierre Bernard, Foundations of ITIL. : Van Haren, 2012

Eventuali indicazioni sui materiali di studio:

Sul sito web del corso (link da <http://www.netadm.it>) e/o su Moodle sono presenti molti documenti scaricabili in formato digitale: slide delle lezioni, video pillole, case study, articoli divulgativi, etc.

KNOWLEDGE REPRESENTATION AND LEARNING

Titolare: da definire

Mutuato da: Laurea magistrale in Data Science (Ord. 2017)

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Suggested basic knowledge of logics and statistics.

Conoscenze e abilità da acquisire:

Introduce the students to the principles for logics for knowledge representation and reasoning, statistical relational learning, and the combination of the two in order to build system for learning and reasoning in hybrid domains.

Attività di apprendimento previste e metodologie di insegnamento:

Lectures supported by exercises and lab.

Contenuti:

(A) Logics for knowledge representation: (A.i) introduction to propositional logics, syntax, semantics, decision procedure. Satisfiability, weighted satisfiability, and best satisfiability. (A.ii) First order logics, syntax, semantics, resolution and unification. (A.iii) Fuzzy logics, syntax, semantics, and reasoning. (B) statistical relational learning: (B.i) Graphical models (B.ii) Markov Logic Networks (B.iii) Probabilistic prolog, (B.iii) Logic Tensor Networks

Modalità di esame:

Critical knowledge of the course topics. Ability to present and apply the studied material.

Criteri di valutazione:

Critical knowledge of the course topics. Ability to present and apply the studied material.

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

Lecture notes and slides for the part not covered by textbooks will be provided.

LANGUAGES FOR CONCURRENCY AND DISTRIBUTION

Titolare: Prof. PAOLO BALDAN

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Il corso non ha propedeuticità.

Conoscenze e abilità da acquisire:

L'enorme diffusione dei sistemi concorrenti, distribuiti e mobili rende inadeguati i paradigmi di specifica e programmazione classici ed apre sfide complesse e affascinanti. Appare necessario un ripensamento, che parta dalle stesse fondamenta e che adotti un approccio rigoroso, formale, disciplinato. Il corso si propone di avvicinare lo studente a tematiche di interesse in questo ambito, utilizzando come strumenti sistemi di tipi, calcoli di processo e in generale linguaggi di modellazione. Parte da argomenti fondazionali oramai classici (come il Calculus of Communicating Systems ed il pi-calculus) e giunge ad illustrare alcuni argomenti di punta della ricerca nell'area. Vengono discussi alcuni linguaggi che traducono in pratica gli sviluppi teorici descritti, quali linguaggi evoluti per la concorrenza (Google Go, Erlang, Clojure), linguaggi di orchestrazione (ORC) e linguaggi per programmazione service oriented (Jolie).

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni in classe e uso di strumenti di verifica automatica.

Contenuti:

La struttura e le tematiche del corso saranno le seguenti: - Introduzione alla concorrenza e mobilità?: dagli automi ai sistemi reattivi e concorrenti. - Calculus of Communicating Systems (CCS), un linguaggio minimale per la descrizione di sistemi concorrenti. Equivalenza di processi: Sistemi di transizione e bisimulazione. - Logica di Hennessy-Milner e strumenti per la verifica. Mutua esclusione, deadlock, fairness. Proprieta? di safety e liveness. Verifica con strumenti automatici. Il Concurrency Workbench e CAAL. - Dai linguaggi di specifica ai linguaggi di programmazione: linguaggi avanzati per la concorrenza (Google Go e channel-based concurrency, Erlang e modello ad attori, Clojure e functional concurrency) - Linguaggi di orchestrazione (ORC) e

linguaggi per programmazione orientata ai servizi (Jolie).

Modalità di esame:

Esercizi in classe, soluzione e discussione orale di esercizi avanzati, presentazione di un tema scelto dallo studente. Tra le opzioni ci sarà anche la realizzazione di un piccolo progetto.

Criteri di valutazione:

Lo studente è valutato rispetto alla sua capacità di risolvere semplici esercizi, verificando così l'acquisizione di nozioni e tecniche discusse durante il corso. Alcuni esercizi avanzati sono finalizzati a verificare la capacità di mettere a frutto quanto appreso per la soluzione di problemi nuovi. La presentazione verifica l'abilità dello studente di approfondire, autonomamente, tematiche di ricerca nell'area di interesse per il corso, e di esporre in modo efficace quanto appreso.

Testi di riferimento:

Luca Aceto, Reactive systems modelling, specification and verification. Cambridge: Cambridge University Press, 2007

Eventuali indicazioni sui materiali di studio:

Il libro di testo è complementato con articoli di ricerca e altre risorse disponibili online. Pagina web: <http://www.math.unipd.it/~baldan/Global>

MACHINE LEARNING

Titolare: Prof. FABIO AIOLLI

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+8L; 6,00

Prerequisiti:

È opportuno avere familiarità con le conoscenze matematiche relative al Calcolo delle Probabilità e all'Analisi di funzioni multivariate. Inoltre è consigliabile avere conoscenze di base relative alla Programmazione e all'Intelligenza Artificiale. L'insegnamento non prevede propedeuticità.

Conoscenze e abilità da acquisire:

In questo insegnamento si presentano alcuni dei concetti fondamentali che caratterizzano l'Apprendimento Automatico, cioè quella classe di tecniche ed algoritmi che a partire da dati empirici permettono di acquisire nuova conoscenza, oppure di correggere e/o raffinare conoscenza già disponibile. Tali tecniche sono particolarmente utili per problemi per cui è impossibile o molto difficile pervenire ad una formalizzazione utilizzabile per la definizione di una soluzione algoritmica ad-hoc. Esempi di tali problemi sono compiti percettivi, come il riconoscimento visivo di cifre manoscritte, e problemi in cui i dati sono corrotti dal rumore e/o incompleti. L'insegnamento tratta principalmente metodi numerici. Sono previste esercitazioni in laboratorio informatico che consentono allo studente di sperimentare le conoscenze acquisite mediante l'applicazione a piccoli esempi pratici.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali ed esercitazioni in laboratorio informatico. Le esercitazioni in laboratorio informatico consistono nella sperimentazione da parte degli studenti delle tecniche viste a lezione sotto vari scenari operativi. In questo modo gli studenti possono verificare sperimentalmente i concetti appresi e acquisire sia capacità di applicazione dei concetti appresi che di giudizio critico.

Contenuti:

La struttura e le tematiche dell'insegnamento saranno le seguenti: - Introduzione: Quando Applicare le Tecniche Proprie dell'Apprendimento Automatico; Paradigmi di Apprendimento Automatico; Gli ingredienti Fondamentali dell'Apprendimento Automatico. - Apprendimento supervisionato: Complessità dello Spazio delle Ipotesi; Misure di Complessità; Esempi di Algoritmi di Apprendimento Supervisionato; - Alberi di Decisione: Apprendimento di Alberi di Decisione; Trattamento di Dati Numerici, di Dati Mancanti, di Costi; Tecniche di Pruning e Derivazione di Regole di Decisione. - Apprendimento Probabilistico: Apprendimento Bayesiano; Esempi di Applicazione al Paradigma Supervisionato e al Paradigma Non-Supervisionato (clustering); Classificatore Ottimo di Bayes; EM. - Reti Neurali e Support Vector Machines: Cenni di Reti Neurali; Margine di Classificazione; Support Vector Machines per Classificazione e Regressione; Funzioni Kernel. - Aspetti Applicativi: Pipeline di Classificazione; Rappresentazione dei dati e Selezione di Variabili; Model Selection; Clustering; Comitati; Sistemi di Raccomandazione.

Modalità di esame:

Lo studente deve superare un esame scritto e, se ritenuto necessario dal docente, un esame orale.

Criteri di valutazione:

Il testo dell'esame scritto contiene alcune domande che consentono di valutare il livello di apprendimento delle nozioni impartite durante l'insegnamento e la capacità dello studente nell'analizzarle criticamente. Sono poi presenti domande in cui si richiede allo studente di mostrare di aver compreso gli aspetti applicativi trattati all'interno delle attività svolte in laboratorio informatico. Tali domande hanno lo scopo di valutare se lo studente ha sviluppato la capacità di applicare le nozioni apprese durante l'insegnamento. Nel caso in cui la valutazione dello scritto non risulti soddisfacente per lo studente, il docente può integrare l'esame scritto con un esame orale per meglio verificare la preparazione dello studente.

Testi di riferimento:

Alpaydin, Elman, Introduction to machine learning. Cambridge: The MIT press, 2010 Mitchell, Tom M., Machine learning. New York: McGraw-Hill, 1998

Eventuali indicazioni sui materiali di studio:

Vengono rese disponibili, come riferimento, i lucidi utilizzati a lezione.

MATHEMATICAL MODELS AND NUMERICAL METHODS FOR BIG DATA

Titolare: Prof. WOLFGANG ERB

Mutuato da: Laurea magistrale in Data Science (Ord. 2017)

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Conoscenze di base di analisi matematica e di algebra lineare. Conoscenze di base di algebra lineare e delle matrici. - Tipo di matrici: diagonale, triangolare, simmetrica, definita positiva - Spettro di una matrice: autovalori e autovettori - Fattorizzazione di una matrice: LU, Cholesky, QR, SVD, Schur
Conoscenze di base di analisi matematica. - norme e metriche in spazi vettoriali - continuita' di funzioni e di operatori - successioni e convergenza
Conoscenze di base di programmazione (Matlab o Python).

Conoscenze e abilità da acquisire:

In questo corso si apprenderanno gli strumenti matematici e computazionali d'avanguardia per l'analisi numerica di megadati soprattutto in vista delle applicazioni nel machine learning. Usando pacchetti di software moderni per dati grandi e sparsi, si impara come implementare questi metodi in maniera efficiente e come applicarli in problemi di vita reale.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni in aula, tutoraggi ed esercitazioni a casa

Contenuti:

1. Numerical methods for large eigenvalue problems • The power method • Krylov subspace iterations for sparse systems of equations • Krylov-type methods for large eigenvalue problem: Arnoldi (and sketches of Lanczos) • Singular value decomposition (SVD) VS eigenvalue decomposition • Best rank-k approximation
2. Numerical analysis of big data: graph and network clustering • Graphs, the graph Laplacian, graph signals, the Cheeger constant • Spectral graph theory and the graph Fourier transform • Spectral clustering • K-Means algorithm and K-center clustering • Principal component analysis and dimensionality reduction
3. Numerical analysis of big data: network centralities • Perron-Frobenius Theorem • Centrality based on eigenvectors (HITS and Pagerank) • Centrality based on matrix functions
4. Multiway Data Analysis • Tensor decompositions: Tucker decomposition, multilinear singular value decomposition, canonical polyadic decomposition • Numerical methods for tensor decompositions and tensor eigenproblems • Applications: Face Recognition Using Tensor SVD and Tensor Data Fusion • Kernel methods for data analysis

Modalità di esame:

Esame scritto alla fine del corso

Criteri di valutazione:

La prova scritta alla fine del corso mira a verificare la comprensione dei fondamenti teorici dei metodi numerici.

Testi di riferimento:

Eldén, Lars, Matrix methods in data mining and pattern recognition. : Society for Industrial and Applied Mathematics, 2019 Saad, Yousef, Numerical methods for large eigenvalue problems. Philadelphia: SIAM, 2011

Eventuali indicazioni sui materiali di studio:

Lezioni estese scritte e codice verra' fornito sul sito web del corso

METHODS AND MODELS FOR COMBINATORIAL OPTIMIZATION

Titolare: Prof. LUIGI DE GIOVANNI

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 32A+4E+12L; 6,00

Prerequisiti:

Elementi di ricerca operativa, elementi di programmazione lineare, elementi di base di programmazione.

Conoscenze e abilità da acquisire:

Uso di metodologie quantitative di supporto alle decisioni per la modellazione e la soluzione di problemi di ottimizzazione combinatoria. Il corso intende fornire strumenti matematici e algoritmici per la soluzione di problemi pratici di ottimizzazione con l'utilizzo dei pacchetti software e delle librerie di ottimizzazione più diffusi.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali, esercitazioni in laboratorio, discussione di esempi notevoli, realizzazione di progetti individuali o di gruppo con stesura di relazione finale. Le esercitazioni in laboratorio consistono nell'implementazione di algoritmi di ottimizzazione combinatoria sia esatti (con l'uso di librerie di programmazione lineare intera) sia euristici.

Contenuti:

1. Approfondimenti e applicazioni di Programmazione Lineare e dualità : metodo del simpleso primale-duale, tecniche di generazione di colonne, applicazioni a problemi di ottimizzazione su grafo. 2. Metodi avanzati di Programmazione Lineare Intera (PLI): Branch & Bound e tecniche di rilassamento, formulazioni alternative di modelli PLI, metodo dei piani di taglio e tecniche di Branch & Cut, applicazioni ad esempi notevoli: commesso viaggiatore, problemi di localizzazione, problemi di network design etc. 3. Meta-euristiche di Ottimizzazione Combinatoria: ricerca di vicini e varianti, algoritmi evolutivi, metodi data-driven (integrazione di tecniche da Machine Learning e Data Science). 4. Applicazione di metodi di modellazione e ottimizzazione su grafo. 5. Laboratori: utilizzo di software e librerie di ottimizzazione.

Modalità di esame:

Esame orale sui contenuti del corso e su esercizi di applicazione di metodi di ottimizzazione a problemi realistici. Realizzazione facoltativa di un progetto individuale su un caso di studio riguardante la soluzione di un problema, reale o realistico, di ottimizzazione combinatoria (definizione del problema, modellazione, applicazione di un metodo di soluzione esatto e/o euristico).

Criteri di valutazione:

L'esame verifica il livello di apprendimento degli argomenti svolti e la capacità dello studente di applicarli per la soluzione di problemi reali di ottimizzazione combinatoria.

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:
Dispense fornite dal docente. Articoli scientifici.

MOBILE AND IOT SECURITY

Titolare: da definire

Mutuato da: Laurea magistrale in Cybersecurity (Ord. 2020)

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Qualsiasi linguaggio di programmazione orientato agli oggetti.

Conoscenze e abilità da acquisire:

Acquisizione dei concetti fondamentali di sicurezza del sistema operativo Android e dei protocolli Bluetooth/Bluetooth Low-Energy. Alla fine del corso, gli studenti avranno acquisito le conoscenze necessarie per analizzare un dispositivo mobile o un'applicazione mobile e identificarne le possibili vulnerabilità.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni; esercitazioni pratiche.

Contenuti:

Teoria: modello di sicurezza di Android, autorizzazioni, gestione dei pacchetti, gestione degli utenti, provider crittografici, sicurezza della rete, archiviazione delle credenziali, gestione degli account online, sicurezza dei dispositivi, SELinux, aggiornamenti del sistema e accesso root, protocolli di comunicazione Bluetooth/Bluetooth Low-Energy. Esercitazione: sicurezza delle applicazioni, superficie di attacco di Android, debugging e analisi delle vulnerabilità, sfruttamento del software dello spazio utente, attacchi a Bluetooth/Bluetooth Low-Energy.

Modalità di esame:

Gli studenti hanno due opzioni. (Opzione 1) Esame pratico, in cui gli studenti risolvono esercizi sulla sicurezza Android; (Opzione 2) Progetto, in cui gli studenti affrontano un tema di ricerca assegnato dal docente e illustrano i risultati ottenuti in una presentazione orale.

Criteri di valutazione:

Conoscenza dei concetti studiati durante il corso.

Testi di riferimento:

CONTENUTO NON PRESENTE

MOBILE PROGRAMMING E MULTIMEDIA

Titolare: Prof.ssa OMBRETTA GAGGI

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Sistemi Operativi, Tecnologie Web

Conoscenze e abilità da acquisire:

Il corso fornisce le competenze necessarie alla realizzazione di un'applicazione in ambito mobile, dalla progettazione, allo sviluppo fino al deployment e al suo mantenimento. Inoltre illustra le principali tecnologie per la codifica, memorizzazione e diffusione di informazioni multimediali.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali

Contenuti:

- Introduzione alla Programmazione mobile. Sistemi operativi: diffusione, storia, caratteristiche. Acquisizione di dati tramite sensori. Caratteristiche delle applicazioni per cellulari. Design dell'interfaccia di un'applicazione mobile. Pubblicazione e distribuzione. - Framework per cross-platform mobile development. Storia, classificazione e caratteristiche. Importanza della scelta di un framework corretto. Consumi energetici. - Introduzione: Sistemi multimediali e ipermediali. I formati dei media. Media e modelli dei dati. Classificazione dei media. Audio, immagini statiche, video. Media statici, media continui, media temporizzati. - Le immagini. Rappresentazione digitale delle immagini. Risoluzione e profondità di colore. Percezione umana del colore. Modelli per la codifica dei colori. Tecniche di riduzione dei colori. Formati standard per la rappresentazione delle immagini: GIF, PNG, JPEG. Il formato JPEG2000. Le immagini vettoriali. - L'audio. Rappresentazione digitale delle informazioni audio. Campionamento e quantizzazione. Teorema di Nyquist. Rapporto segnale-rumore. Dimensione dei dati e banda di trasmissione. Formati standard per la codifica dell'audio: WAV, u-Law. I sistemi MIDI. - Il video. Rappresentazione del segnale video analogico. Standard NTSC e PAL. Il video digitale. Rappresentazione del colore. Sottocampionamento cromatico. Standard H261, H263, MPEG. - La compressione dei dati. Compressione reversibile e compressione irreversibile. Compressione entropica. Compressione LZW. Compressione dei dati acustici. Elementi di psicoacustica. Bande critiche. Mascheramento spaziale e temporale. Compressione MP3. Compressione JPEG delle immagini. Compressione video. Codifica predittiva. Vettori di movimento. Compressione MPEG.

Modalità di esame:

L'esame consiste nello sviluppo di un progetto e una prova orale.

Criteri di valutazione:

L'esame verifica l'effettivo apprendimento dei concetti esposti durante l'insegnamento. Questo può avvenire in forma di discussione orale, oppure

applicando quanto appreso nella progettazione e realizzazione di una applicazione per smartphone.

Testi di riferimento:

Josh Clark., Designing for Touch. : A Book Apart, 2015 Li, Ze-Nian; Drew, Mark S., Fundamentals of multimedia. Upper Saddle River: Pearson Prentice Hall, 0

Eventuali indicazioni sui materiali di studio:

Le slide del corso sono fornite sul sito web del corso

PROCESS MINING

Titolare: Prof. MASSIMILIANO DE LEONI

Mutuato da: Laurea magistrale in Data Science (Ord. 2017)

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Conoscenze di base di algoritmi, strutture dati e programmazione, come acquisiti nel corso "Fundamental of Information Systems"

Conoscenze e abilità da acquisire:

Lo scopo del corso e' quello di introdurre e studiare metodi e concetti che riguardano la modellazione e l'analisi di processi di business. In dettaglio, il corso inizia con illustrare i principi di modellazione di processi come ci si attende che vengano eseguiti (i cosiddetti "TO-BE processes"), usando i più comuni linguaggi di modellazione (BPMN, Petri Nets e Declare), e con descrivere le tecniche di analisi della loro correttezza. Successivamente, il corso si focalizza sul Process Mining, che si pone l'obiettivo di utilizzare lo storico dei dati transazionali (i log degli eventi) per analizzare e migliorare i processi come sono eseguiti (noti come "AS-IS processes"), che possono differire da quelli attesi. Un enfasi speciale verrà messa sugli algoritmi e sulla loro implementazione in piattaforme software open-source e commerciali. Sessioni pratiche complementano il corso affinché gli studenti possano toccare con mano i problemi di modellazione e mining di processi aziendali in contesti reali/realistici, utilizzando dati reali.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso consiste di lezioni, esercitazioni e sessioni di laboratorio. Le sessioni di laboratorio creeranno l'opportunità di "sporcarsi le mani" su dati reali, così da comprendere la complessità di eseguire un progetto di Process Mining e di lavorare con dati di processo reali.

Contenuti:

Il corso affronterà i seguenti argomenti: 1. MODELLAZIONE ED ANALISI ATTRAVERSO IL LINGUAGGIO BPMN - Modellazione di processo base ed avanzata in BPMN - Analisi Qualitativa - Analisi Quantitativa 2. MODELLAZIONE ED ANALISI TRAMITE RETI DI PETRI - Concetti di Base di Reti di Petri - Uso delle Reti di Petri per la Modellazione dei Processi Aziendali - Analisi Strutturale di Reti di Petri - Correttezza di Modelli di Processi Aziendali basati su Petri Nets: Principi e Verifica. 3. PROCESS MINING - Introduzione su Process Mining e il log degli eventi - Tecniche di Base per Process Discovery e loro Limiti - Tecniche Avanzate per Process Discovery: Heuristic Miner and Inductive Miner - Verifica di Conformità basata su Token Replay - Verifica di Conformità basata sugli Allineamenti - Il Mining delle Prospettive Aggiuntive su Decisioni, Tempistiche e Risorse - Analisi delle Reti Sociali 4. TECNICHE PREDITTIVE E PROCESSI DI BUSINESS - Tecniche di Base di Predictive Process Monitoring - Tecniche Avanzate di Predictive Process Monitoring

Modalità di esame:

Esame scritto e progetto.

Criteri di valutazione:

Il lavoro di progetto e l'esame scritto saranno valutati sulla base dei seguenti criteri: i) conoscenza dello/a studente/ssa di concetti, metodi e tecnologie; ii) abilità dello/a studente/ssa di padroneggiare le tecnologie di implementazione; iii) le capacità di sintesi, precisione, chiarezza e astrazione dello/a studente/ssa dimostrate nell'esame scritto e nella presentazione di progetto. Il voto finale è ottenuto come la somma pesata dei voti dell'esame scritto (60%) e del progetto (40%).

Testi di riferimento:

Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A. Reijers, Fundamentals of Business Process Management. : Springer, 2013 Wil M.P. van der Aalst and Christian Stahl, Modeling Business Processes: A Petri Net-Oriented Approach.. : Information Systems, 2011 Wil M.P. van der Aalst, Process Mining: Data Science in Action. Berlin: Springer-Verlag, 2016

Eventuali indicazioni sui materiali di studio:

Slides, Eserciziari, Esercitazioni di Laboratori Autoguidate e Articoli Scientifici.

REAL-TIME KERNELS AND SYSTEMS

Titolare: Prof. TULLIO VARDANEGA

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+8E; 6,00

Prerequisiti:

L'insegnamento assume familiarità con l'architettura degli elaboratori tradizionali, con la struttura e le attività dei loro sistemi operativi, particolarmente per quanto attiene a concorrenza, sincronizzazione e gestione dell'I/O. L'insegnamento non prevede propedeuticità.

Conoscenze e abilità da acquisire:

Il corso si propone di esaminare la struttura dei sistemi software embedded soggetti a vincoli temporali, con l'obiettivo di evidenziarne le caratteristiche che più li differenziano dagli altri sistemi di calcolo. Attenzione sarà posta su alcuni paradigmi di progettazione e programmazione di tali sistemi, che ne facilitano l'analisi e la verifica.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso esamina la struttura dei sistemi software embedded soggetti a vincoli di tempo reale, illustrando le principali problematiche nella loro progettazione, realizzazione e validazione. In particolare vengono affrontate: - caratterizzazione architetturale (livello hardware, software, e sistema) - controllo e gestione del tempo e delle interfacce hardware - progettazione e programmazione di software real-time - tecniche e approcci per la modellazione e l'analisi di sistemi real-time - problematiche di verifica e validazione. Nell'ambito del corso, il docente propone allo studente esercizi da realizzare in proprio in laboratorio per sperimentare direttamente le problematiche progettuali e realizzative e i paradigmi di soluzione illustrati a lezione, oltre a familiarizzare gli studenti con i più recenti sviluppi della teoria real-time intorno a tematiche di particolare interesse. I metodi di insegnamento previsti includono: 1) insegnamento d'aula, sia sincrono che asincrono; 2) lezioni "rovesciate", svolte in modalità "flipped classroom", su argomenti selezionati, centrati prevalentemente su strumenti e tecniche di sviluppo collaborativo; 3) esercitazione guidate su temi di teoria; 4) attività pratiche con feedback di progresso.

Contenuti:

- Introduzione: cenni storici e visione architetturale - Cenni sulla affidabilità e la tolleranza ai guasti - Il problema dell'ordinamento, tassonomia di algoritmi - Politiche di sincronizzazione nella gestione delle risorse condivise - Problematiche di sistema: una visione d'insieme della pila tecnologica - Estensione ai sistemi distribuiti - Estensione ai sistemi multiprocessore

Modalità di esame:

L'esame si svolge in una di due modalità a scelta dello studente. Una modalità richiede la redazione e la presentazione di una relazione tecnica sulle problematiche incontrate nell'adattamento a principi di progettazione e programmazione real-time di un piccolo sistema concorrente e distribuito individuato congiuntamente dallo studente e dal docente. L'altra modalità prevede lo studio critico e la presentazione di un lavoro di ricerca recente, che sviluppa qualcuno dei temi toccati in aula, scelto dallo studente tra un insieme di lavori individuati dal docente.

Criteri di valutazione:

Lo sviluppo della prova d'esame scelta dallo studente, indipendentemente dalle sue specifiche modalità, viene accompagnato da intenso dialogo con il docente, che consente allo studente di approfondire le principali problematiche affrontate a lezione e associate alla realizzazione del progetto. La presentazione e discussione da effettuare in sede d'esame consente di completare la valutazione il grado di apprendimento complessivo dello studente rispetto ai principali temi della materia.

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

Il docente pubblica regolarmente tutte le diapositive utilizzate a lezione e anche materiale supplementare utile per l'approfondimento dei temi trattati in aula. Tutto il materiale didattico dell'insegnamento è reso disponibile attraverso la piattaforma Moodle, per la parte di risorse online accessibile con autenticazione, e una apposita pagina-calendario statica dell'insegnamento, che fornisce materiale di apprendimento, di pubblico dominio.

RUNTIMES FOR CONCURRENCY AND DISTRIBUTION
--

Titolare: Prof. TULLIO VARDANEGA

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

L'insegnamento assume familiarità con l'architettura degli elaboratori tradizionali, con la struttura e le attività dei loro sistemi operativi, particolarmente per quanto attiene la concorrenza, sincronizzazione e gestione dell'I/O, e dei protocolli di rete. Purtroppo, l'insegnamento non prevede propedeuticità, perché le attività d'aula sono disegnate per aiutare lo studente a colmare le proprie lacune e rinfrescare le proprie conoscenze di dominio.

Conoscenze e abilità da acquisire:

Il corso si propone di: 1) illustrare problematiche e modelli di base e avanzati di concorrenza intesa come parallelismo potenziale, realizzata a software, illustrando e studiando, dal punto di vista del sistema a run-time, le soluzioni proposte da Ada e Java e altri linguaggi dotati di supporto diretto alla concorrenza (p.es. Go e Rust), come strumenti di sperimentazione e di confronto; 2) analizzare i principi costruttivi e i paradigmi architetturali e realizzativi che stanno alla base dei sistemi distribuiti, nella loro evoluzione da sistemi multiprocessori omogenei a sistemi multicomputer eterogenei, su larga o larghissima scala fino al paradigma del Cloud Computing, anch'essi studiati dal punto di vista del run-time support.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso si compone di due segmenti principali e di un terzo e ultimo tratto. - Nel primo segmento si prendono in esame modelli e paradigmi di programmazione concorrente, concentrando sulla concorrenza direttamente esprimibile a linguaggio (ossia senza ricorso a librerie esterne), utilizzando Java e Ada come linguaggi di sperimentazione, ma estendendo l'attenzione ad altri paradigmi linguistici. - Nel secondo segmento si affronta invece l'evoluzione architetturale tecnologica dei sistemi distribuiti, toccando CORBA come paradigma storico (anche se ormai tecnicamente obsoleto) di interconnessione di sistemi eterogenei secondo il modello cliente-servente. - Nel terzo e ultimo tratto, l'insegnamento culmina con l'illustrazione di approcci particolarmente avanzati come virtualizzazione e Cloud Computing. Nell'ambito di ciascun segmento del corso, il docente propone allo studente esercizi da realizzare in proprio in laboratorio per sperimentare direttamente le problematiche progettuali e realizzative e i paradigmi di soluzione illustrati a lezione. I metodi di insegnamento previsti includono: 1) insegnamento d'aula, sia sincrono che asincrono; 2) lezioni "rovesciate", svolte in modalità "flipped classroom", su argomenti selezionati, centrati prevalentemente su strumenti e tecniche di sviluppo collaborativo; 3) esercitazione guidate su temi di teoria; 4) attività pratiche con feedback di progresso.

Contenuti:

Problematiche di concorrenza - Introduzione storica e metodologica - Nozione di processo e modalità di sincronizzazione - Un modello concreto e sue progressive estensioni - La dimensione temporale - Cenni sulla virtualizzazione Problematiche di distribuzione - Definizioni fondamentali - Comunicazione e sincronizzazione in distribuito - Il sistema dei nomi e la nozione di statelessness - Soluzioni concrete: Java RMI, Ada DSA, CORBA La frontiera del Cloud Computing - Origini del paradigma - Caratteristiche di applicazioni cloud-native - Le dimensioni della scalabilità.

Modalità di esame:

L'esame di profitto consiste nella redazione e nella discussione di una relazione scritta che illustri le problematiche affrontate nello svolgimento di un tema didattico concordato con lo studente, e le soluzioni tecniche adottate per risolverle. Il compito assegnato include sempre una componente di studio e una di sperimentazione pratica.

Criteri di valutazione:

Il lavoro assegnato in sede d'esame viene accompagnato da intenso dialogo con il docente, che consente allo studente di approfondire le principali problematiche affrontate a lezione e associarle alla realizzazione del progetto assegnato. La stesura della relazione tecnica mette alla prova la capacità di sintesi e di astrazione dello studente; le argomentazioni tecniche sostenute nella relazione consentono di completare la valutazione il grado di apprendimento complessivo dello studente rispetto ai principali temi della materia.

Testi di riferimento:

Alan Burns and Andy Wellings, Concurrent and Real-Time Programming in Ada. : Cambridge University Press, 2007 Andrew S Tanenbaum, Maarten van Steen, Distributed Systems - Principles and paradigms. : Pearson Education International, 2006

Eventuali indicazioni sui materiali di studio:

Tutto il materiale didattico dell'insegnamento è reso disponibile attraverso la piattaforma Moodle, per la parte di risorse online accessibile con autenticazione, e una apposita pagina-calendario statica dell'insegnamento, che fornisce materiale di apprendimento, di pubblico dominio.

SOFTWARE VERIFICATION

Titolare: Prof. FRANCESCO RANZATO

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Conoscenze di base dei linguaggi di programmazione. L'insegnamento non prevede propedeuticità.

Conoscenze e abilità da acquisire:

Il corso mira ad introdurre metodi e strumenti per la specifica del comportamento run-time dei programmi, l'analisi statica e la verifica automatica dei programmi e, più in generale, dei sistemi software. In particolare, il corso fornisce una introduzione alla semantica formale dei linguaggi di programmazione ed ai metodi formali per la loro analisi statica e verifica.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali (o in modalità telematica) e la risoluzione in modo indipendente a casa di vari esercizi e/o lo sviluppo di un progetto di verifica del software. Sono previste lezioni invitate di ospiti ricercatori su tematiche avanzate di verifica del software.

Contenuti:

- Semantica dei programmi: Modellazione del comportamento (in particolare il comportamento input/output) dei programmi mediante la teoria dell'ordinamento e dei punti fissi. (cf. [https://en.wikipedia.org/wiki/Semantics_\(computer_science\)](https://en.wikipedia.org/wiki/Semantics_(computer_science))) - Analisi statica e verifica di programmi mediante interpretazione astratta: L'interpretazione astratta è una notoria tecnica basata su una approssimazione della semantica dei programmi che permette di specificare le proprietà dei programmi deducibili mediante analisi statica e di provarne la correttezza. (cf. https://en.wikipedia.org/wiki/Abstract_interpretation) - Analisi statica dataflow di programmi: tecnica per dedurre staticamente informazioni sull'insieme dei possibili valori delle variabili nei vari punti del programma. Un grafo di flusso del controllo è utilizzato per determinare le parti di un programma a cui un particolare valore assegnato ad una variabile potrebbe propagarsi. Le informazioni raccolte sono spesso utilizzate dai compilatori (come gcc e javac) per ottimizzare un programma. (cf. https://en.wikipedia.org/wiki/Data-flow_analysis) - Strumenti di verifica del software: ad esempio, Clousot (Microsoft, USA), Interproc (INRIA, Francia), Jandom (Università di Pescara) (cf. https://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis)

Modalità di esame:

Esame orale e/o progetto software, possibilmente suddivisi in parti distinte.

Criteri di valutazione:

L'esame orale verte su vari esercizi che lo studente deve svolgere in modo indipendente a casa. Il progetto di laboratorio verte su qualche tool di verifica del software.

Testi di riferimento:

H. Riis Nielson, F. Nielson, Semantics with Applications: A Formal Introduction. : Wiley, 1992 Antoine Minè, Tutorial on static inference of numeric invariants by abstract interpretation. : Now, The Essence of Knowledge, 2017

Eventuali indicazioni sui materiali di studio:

Le slide utilizzate a lezione verranno distribuite.

START-UP IN ICT

Titolare: Dott. FABIO D'ALESSI

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Nessuno.

Conoscenze e abilità da acquisire:

Le conoscenze e abilità sulle quali si focalizza il corso sono (1) l'individuazione di vettori di innovazione e la comprensione del loro potenziale di impatto sul business; (2) i processi dell'innovazione, attraverso le fasi principali di start-up e scale-up, con l'analisi del ruolo che in esse giocano investitori e

finanziatori; (3) attività di laboratorio per attraversare, in breve, i passaggi da (1) verso l'avvio di (2).

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali, visite a realtà significative, testimonianze in aula, attività di laboratorio.

Contenuti:

- Introduzione all'innovazione e l'imprenditorialità. - Vettori di innovazione: Internet of things. Blockchain. Augmented Reality. Cyber physical systems (edge, fog, cloud). Esempi di applicazione al business - I processi dell'innovazione: come si deve governare la fase di start-up, la più rischiosa e intangibile? Governance della fase di start-up. Come si calcola il valore di una impresa in fase di start up? - Investitori istituzionali: business angels e venture capital. Come funzionano, quali sono i loro obiettivi, cosa cercano? - Il sostegno pubblico alla fase di start up, normativa e strumenti disponibili. Perché? il pubblico sostiene l'innovazione in fase di start up? Programmi di sostegno in Italia e all'estero. - Gestione dell'equity di una start up. L'equity di una start up serve a motivare il team, a raccogliere le risorse finanziarie necessarie ad aprire collaborazioni. - Illustrazione di temi di progetti di gruppo (casi di studio) e modalità di svolgimento. Approfondimenti ed esercizi sui casi di studio. Laboratorio su progetti di gruppo (anche con visite presso aziende o stakeholder dell'innovazione) - I processi dell'innovazione: quando termina la fase di start up? Quali sono gli obiettivi della fase successiva, quella di scale up? - I processi dell'innovazione: la fase di scale up. Come si struttura l'organizzazione nella fase di scale up? (testimonial, portare manager di imprese informatiche) - I processi dell'innovazione: come finanziare la fase di scale up; il sostegno pubblico alla fase di scale up. (testimonial dal mondo del debito e delle banche)

Modalità di esame:

Gli studenti si costituiranno in piccoli gruppi (orientativamente 3 persone gruppo), per lavorare sull'individuazione di un vettore di innovazione di interesse, all'interno di un insieme di temi indicato dal docente, e su di esso costruire una proposta di valore da sottoporre al docente e ad esperti esterni per un giudizio finale. L'attività inizia a metà del programma e si svolge, anche in aula, in modo da favorire l'interazione con il docente.

Criteri di valutazione:

Valutazione dei docenti e degli esperti esterni.

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

L'insegnamento non adotta un testo di riferimento. Le attività frontali saranno accompagnate da diapositive e da riferimento a materiale di approfondimento nel pubblico dominio.

STRUCTURAL BIOINFORMATICS

Titolare: Prof. DAMIANO PIOVESAN

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 32A+16E; 6,00

Prerequisiti:

Conoscenze base di algoritmi di ottimizzazione e machine learning. Linguaggio di programmazione: Python.

Conoscenze e abilità da acquisire:

Il corso intende comunicare le conoscenze di base sulla struttura e funzione della materia vivente nonché i principali metodi computazionali per il loro studio. Inoltre intende permettere allo studente lo svolgimento autonomo di un progetto di ricerca in bioinformatica strutturale, definendo lo stato dell'arte per un problema aperto e un tentativo di risolverlo con lo sviluppo di software che estenda librerie esistenti e la valutazione critica dei risultati ottenuti.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso si compone di lezioni frontali, esercitazioni pratiche al computer, sviluppo di un progetto e presentazione dello stesso con discussione critica. Le esercitazioni servono per familiarizzare lo studente con le librerie software da usare per un progetto bioinformatico relativo ad un problema attuale diverso per ogni gruppo. La presentazione del progetto richiede una discussione in cui far emergere i punti di forza e debolezza del software implementato.

Contenuti:

Il corso si compone di due parti: 1) Introduzione alla materia vivente (2 CFU): 1.1) Cenni di chimica organica, interazioni deboli ed energetica 1.2) Struttura e funzione di DNA e proteine 1.3) Lipidi, membrane e trasporto cellulare 1.4) Metodi sperimentali per la determinazione strutturale 2) Biochimica computazionale (4 CFU): 2.1) Banche dati biologiche 2.2) Librerie software e concetti per allineamenti di sequenza e ricerca in banche dati 2.3) Relazione sequenza – struttura nelle proteine e classificazione strutturale 2.4) Metodi per la predizione della struttura delle proteine da sequenza, l'esperimento CASP 2.5) Metodi per la predizione di funzione ed interazioni delle proteine, l'esperimento CAFA 2.6) Proteine non globulari, disordine e ripetizioni strutturali

Modalità di esame:

L'esame si compone di tre parti separate, che devono essere superate tutte: (i valori tra parentesi indicano i pesi per il voto complessivo) 1) Test scritto sulle nozioni di biochimica (ca. 30%) 2) Progetto software (ca. 40%) 3) Presentazione del progetto con valutazione critica (ca. 30%)

Criteri di valutazione:

Viene valutata: 1) la comprensione di concetti e gli algoritmi presentati a lezione 2) la capacità di applicare le nozioni fornite a lezione su problemi reali 3) la capacità critica di saper utilizzare i metodi nei modi più opportuni, scegliendo tra le alternative possibili 4) la capacità di sviluppare software riutilizzabile estendendo librerie esistenti 5) la capacità espositiva e di discussione critica

Testi di riferimento:

K.C. Mathews, K.E. Van Holde, K.G. Ahern, Biochimica 3° edizione. : Casa Editrice Ambrosiana, 2004 Pascarella, Stefano; Paiardini, Alessandro, Bioinformatica dalla sequenza alla struttura delle proteine. Bologna: Zanichelli, 2015 Gu, Jenny; Bourne, Philip E., Structural bioinformatics (Second edition). : Hoboken: Wiley-Blackwell, 2009 K.C. Mathews, K.E. Van Holde, K.G. Ahern, Biochimica 3° edizione. : Casa Editrice Ambrosiana, 2004

Eventuali indicazioni sui materiali di studio:

Sul sito E-learning vengono resi disponibili molti materiali per il corso. Questi comprendono i lucidi del corso (appena disponibili), le dispense e la letteratura usata per i progetti. Le dispense scaricabili in formato PDF contengono oltre 300 pagine per facilitare lo studio.

TYPE THEORY

Titolare: Prof.ssa MARIA EMILIA MAIETTI

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

E' caldamente suggerito, ma non strettamente necessario, aver seguito un corso di introduzione alla logica matematica.

Conoscenze e abilità da acquisire:

Lo scopo di questo corso è quello di fornire una introduzione teorica alla teoria dei tipi per poter apprezzare le sue applicazioni in informatica (correttezza dei programmi funzionali e loro verifica in proof-assistant) e in matematica (sviluppo di prove costruttive con eventuale verifica formale al calcolatore ed estrazione del loro contenuto computazionale).

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali e laboratorio sull'apprendimento di un proof-assistant.

Contenuti:

Nel corso verranno introdotti i principali concetti di teoria dei tipi al fine di apprezzarne alcune rilevanti applicazioni in ambito matematico, informatico e anche filosofico. Lo studente verrà introdotto a comprendere i seguenti aspetti della sua multiforme natura: 1) La natura computazionale della teoria dei tipi vista come lambda-calcolo tipato a' la Church che la rende un paradigma di un linguaggio di programmazione funzionale, in quanto permette di tipare i programmi con la loro specifica e di verificarne la correttezza in modo interattivo per mezzo di un proof-assistant. 2) La natura insiemistica della teoria dei tipi che la rende adatta a formalizzare le dimostrazioni in matematica costruttiva in modo da estrarne il contenuto computazionale. 3) La natura predicativa delle costruzioni di tipo dipendente a' la Martin-Löf tramite generazione induttiva. Verranno illustrati esempi di costruzioni non predicative tramite l'utilizzo di paradossi. 4) La possibilità di presentare versioni intensionali o estensionali della teoria dei tipi al fine di enucleare importanti proprietà di decidibilità del type-checking utili a costruire un proof-assistant affidabile che permetta di formalizzare le prove matematiche in modo interattivo. Il corso includerà una parte in laboratorio che introdurrà gli studenti all'utilizzo di un proof-assistant (il francese Coq o il bolognese Matita o lo svedese Agda).

Modalità di esame:

L'accertamento di profitto avverrà con una prova orale dopo il completamento di esercitazioni personali da parte dello studente.

Criteri di valutazione:

L'esame intende valutare le conoscenze acquisite dallo studente sui temi del corso e le sue capacità di svolgere del lavoro autonomo su di essi.

Testi di riferimento:

Bengt Nordström, Kent Petersson, Jan M. Smith, Programming in Martin-Loef's Type Theory. : Oxford University Press, 1990 P. Martin-Löf, Intuitionistic type theory. Notes by G. Sambin of a series of lectures given in Padua, June 1980. : Bibliopolis, 1984

Eventuali indicazioni sui materiali di studio:

Appunti forniti dal docente.

VISION AND COGNITIVE SYSTEMS

Titolare: Prof. LAMBERTO BALLAN

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 32A+16L; 6,00

Prerequisiti:

Lo studente deve avere conoscenze di base di programmazione e algoritmi. È inoltre consigliabile conoscere i concetti di base in termini di probabilità e di analisi delle funzioni multivariate.

Conoscenze e abilità da acquisire:

Questo corso insegna i concetti, i metodi e le tecnologie alla base della visione artificiale e dei sistemi cognitivi, incluso i moderni servizi cognitivi, vale a dire API e servizi tipicamente disponibili su cloud, che aiutano gli sviluppatori a creare applicazioni di intelligenza artificiale. Esempi di funzioni intelligenti che possono essere aggiunte ad un'applicazione tramite l'utilizzo di servizi cognitivi sono: il riconoscimento visuale; il rilevamento delle emozioni da video ed il riconoscimento facciale; comprensione linguistica e del parlato. Il corso insegna inoltre le competenze e le abilità specifiche necessarie per applicare tali concetti alla progettazione e all'implementazione di applicazioni di intelligenza artificiale. Gli studenti dovranno affrontare esercizi pratici in laboratorio informatico, in modo da provare l'applicazione delle conoscenze acquisite a piccoli esempi pratici.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso consiste in lezioni e esercizi in laboratorio informatico. Gli esercizi in laboratorio informatico consentono agli studenti di sperimentare, in diversi scenari operativi, le tecniche introdotte a lezione. In questo modo gli studenti possono verificare sperimentalmente i concetti appresi in classe, acquisire la capacità di applicare i concetti appresi e di esprimere un giudizio critico.

Contenuti:

Il corso comprende gli argomenti elencati di seguito: - Introduzione: Dalla cognizione umana all'intelligenza artificiale e ai sistemi cognitivi; breve introduzione ai paradigmi di intelligenza artificiale e apprendimento automatico; la rivoluzione dell'IA: attuali tendenze e applicazioni, le principali sfide. - Servizi cognitivi: Concetti basilari; servizi linguistici, vocali e di visione; principali provider e API (IBM Watson, AWS, Google Cloud); tecnologie abilitanti. -

Apprendimento automatico ed applicazioni: Classificazione; introduzione al deep learning e all'apprendimento di rappresentazioni; fasi di addestramento e test; misure di valutazione; il bias negli algoritmi. - Visione ed elaborazione di immagini: Percezione nelle macchine; formazione dell'immagine, campionamento, filtraggio e operatori lineari; gradiente dell'immagine, edge e corner; progettare descrittori visuali efficaci (SIFT e feature basate sul gradiente); confronto tra immagini. - Riconoscimento visivo e oltre: "Insegnare ai computer a vedere": bag-of-feature, piramidi spaziali e pooling; apprendimento di rappresentazioni per la visione, reti neurali convoluzionali; R-CNN e segmentazione; descrizione di immagini, scenari multi-modali e uno sguardo oltre al paradigma di apprendimento supervisionato. - Esercizi pratici: Cosa c'è nella scatola? Come costruire una pipeline di riconoscimento visivo; utilizzare i servizi cognitivi per il riconoscimento / comprensione delle immagini; combinare diversi servizi e modalità.

Modalità di esame:

Lo studente deve sviluppare, in accordo con il docente, un piccolo progetto applicativo. Inoltre, lo studente deve presentare una relazione scritta sul progetto svolto, in cui si discutono criticamente tutte le questioni trattate durante la sua realizzazione. L'esame consisterà prevalentemente in una breve presentazione e discussione del progetto svolto, in cui il docente potrà anche chiedere dettagli e/o altri contenuti visti a lezione.

Criteri di valutazione:

Il lavoro di progetto e l'esame orale saranno valutati sulla base dei seguenti criteri: a) conoscenza da parte dello studente dei concetti, dei metodi e delle tecnologie alla base dei servizi cognitivi (con particolare enfasi sulle tematiche di visione artificiale); b) capacità dello studente di padroneggiare la tecnologia di implementazione; c) capacità di sintesi, chiarezza e astrazione dello studente, come dimostrato dalla relazione scritta e dal progetto.

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

Le presentazioni mostrate durante le lezioni sono rese disponibili su Moodle come materiale di riferimento.

WEB INFORMATION MANAGEMENT

Titolare: Prof. MASSIMO MARCHIORI

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

E' opportuno avere familiarità con gli elementi di base del web, così come forniti nel corso di "Tecnologie Web", in particolare HTML, CSS, XML, XSLT.

Conoscenze e abilità da acquisire:

L'obiettivo principale del corso è quello introdurre alcune tra le principali tecnologie web di livello avanzato relative alla gestione dell'informazione, in modo da avere una visione ad alto livello del web attuale e del suo futuro. Guarderemo oltre la superficie, mostrando le connessioni profonde, e spesso sorprendenti, tra il mondo tecnologico e quello sociale.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali, con esempi illustrativi mostrati anche tramite connessione diretta al web.

Contenuti:

+ Web Usability Usabilità ed interazione con gli utenti, analisi multi-livello, come costruire un sito web di successo. Confronto tra il mondo desktop e quello mobile. + E-commerce Il caso studio dei siti di e-commerce, specializzazione dell'interazione col cliente. + Web Advertisement La pubblicità nei siti web, tecniche d'uso ed errori da evitare. + Web Search Web Site Search, Search Engine Optimization, testo ed ipertesto, il bene ed il male del web, i Social Information Systems. + Web Naming I nomi del web, loro usi ed abusi. + Il Web della Conoscenza Fondamenti del web semantico, rappresentazione della conoscenza, ontologie, semantic querying, syntactic querying, web reasoning, complex systems.

Modalità di esame:

Lo studente deve superare uno scritto, e consegnare un progetto. Sopra una certa soglia minima di punteggio lo studente può opzionalmente richiedere un ulteriore esame orale.

Criteri di valutazione:

Il criterio di valutazione principale è la comprensione delle tecnologie web mostrate durante il corso. Questo significa quindi conoscere il funzionamento, i punti deboli ed i punti di forza delle tecnologie, la loro interazione nel contesto.

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

Il materiale di studio per l'esame è fornito tramite il sito web del corso attraverso risorse online.

WIRELESS NETWORKS FOR MOBILE APPLICATIONS

Titolare: Prof. CLAUDIO ENRICO PALAZZI

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+8L; 6,00

Prerequisiti:

Reti di Calcolatori

Conoscenze e abilità da acquisire:

Questo corso offre una panoramica delle problematiche inerenti sistemi e servizi basati su reti wireless. A questo scopo, sono analizzati i principali

problemi e soluzioni protocollari disponibili per ambienti wireless. Inoltre, sono discussi la terminologia, il funzionamento e le possibili alternative allo stato dell'arte nelle comunicazioni wireless. Attraverso l'analisi dei servizi che possono essere offerti su tecnologia wireless, lo studente diventerà consapevole delle possibili evoluzioni ed utilizzi futuri dei sistemi wireless. Infine, il corso si conclude con alcune nozioni utili all'implementazione di un elaborato volto all'analisi e alla progettazione di protocolli/applicazioni wireless.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali e la realizzazione di un progetto.

Contenuti:

Introduzione alle reti wireless. Problematiche relative alle reti wireless: perdite per errore e collisione, equità e ritardi di trasmissione, handoff Standard MAC: 802.11 a/b/g/n/p/s Protocolli di trasporto in ambiente wireless: TCP Vegas, TCP Westwood, TCP Hybla, CUBIC. Reti ad hoc e protocolli di routing: MANET, VANET, DSDV, AODV, DSR. Applicazioni e servizi su reti mobili.

Modalità di esame:

Gli studenti sono valutati attraverso progetti individuali o di squadra ed attraverso un esame orale sulle tematiche discusse in aula.

Criteri di valutazione:

L'esame orale finale e il progetto realizzato consentono di valutare il livello di apprendimento delle nozioni discusse in classe e l'abilità dello studente nel maneggiare concetti in modo pratico.

Testi di riferimento:

William Stallings, Wireless Communications & Networks (2nd Edition). : Prentice Hall, 2005

Eventuali indicazioni sui materiali di studio:

Vengono rese disponibili le trasparenze utilizzate in aula.