



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



Bollettino Notiziario - A.A. 2024/2025

LAUREA MAGISTRALE IN COMPUTER SCIENCE (ORD. 2021)

Curriculum: Corsi comuni

ADVANCED ALGORITHMS

Titolare: Prof. MICHELE SCQUZZATO

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Sebbene non vi siano prerequisiti formali, si presuppone l'aver precedentemente fatto un corso di algoritmi e una buona conoscenza della probabilità (discreta).

Conoscenze e abilità da acquisire:

Competenze avanzate nell'ambito del progetto e dell'analisi di algoritmi, tra cui tecniche per l'approssimazione e l'uso della randomizzazione.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento si basa su lezioni in aula. Le lezioni frontali, oltre a presentare i contenuti del corso, dedicano attenzione al processo di sviluppo delle idee e al loro uso innovativo in contesti diversi da quello in cui sono state apprese.

Contenuti:

Il corso copre tre argomenti principali: 1) Algoritmi su grafi per problemi fondamentali come componenti connesse, alberi di copertura minimi, cammini minimi e flussi massimi; 2) Algoritmi di approssimazione per problemi intrattabili come la copertura dei vertici, la copertura dell'insieme e il problema del commesso viaggiatore; 3) Algoritmi randomizzati: principali tecniche e applicazioni a problemi quali l'ordinamento e i tagli minimi.

Modalità di esame:

Esame scritto.

Criteri di valutazione:

La prova scritta contiene domande sugli argomenti affrontati a lezione ed esercizi volti a verificare la capacità dello studente di utilizzare le nozioni apprese durante il corso per risolvere nuovi problemi.

Testi di riferimento:

Mitzenmacher and Upfal, Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis. : Cambridge University Press, 2017 Cormen, Leiserson, Rivest, Stein, Introduction to Algorithms. : MIT press, 2022

Eventuali indicazioni sui materiali di studio:

La sezione dedicata sulla piattaforma Moodle raccoglierà diverso materiale come dispense, esercizi con soluzioni, e ulteriore materiale di approfondimento. Moodle sarà anche il metodo di comunicazione principale, e tutti gli annunci verranno pubblicati su Moodle.

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: Prof. CLAUDIO ENRICO PALAZZI

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:

Verifica su ogni modulo del corso.

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.

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 opzionale 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 può opzionalmente 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 opzionale 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: buone competenze relative al progetto e all'analisi di algoritmi e strutture dati, conoscenza delle nozioni fondamentali di calcolo delle probabilità e statistica, buone capacità di programmazione in Java o Python, e la capacità di usare Linux via command line.

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:

Il corso affronterà i seguenti argomenti: Introduction to the Big Data phenomenon. Distributed frameworks: MapReduce, Apache Spark. Clustering for data analysis and summarization. Analysis of data streams. Similarity Search.

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, 3rd Edition. : Cambridge University Press, 2020

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.ssa LAURA TREU

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 la predizione e annotazione genica, l'analisi del trascrittoma e la loro integrazione attraverso modelli computazionali del metabolismo. Inoltre viene discusso il ruolo della bioinformatica nell'analisi di genomi individuali e nella medicina personalizzata.

Modalità di esame:

L'esame si svolgerà in modalità scritta.

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:

Saranno distribuite delle note integrative con esercizi suggeriti e svolti. Tutto il materiale sarà pubblicato sul moodle del corso e alla pagina web: <http://www.math.unipd.it/~aldan/Computability>

CRYPTOGRAPHY

Titolare: Prof. CARLO MARICONDA

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 (CRYPTOGRAPHY, Prof. Mariconda; primo semestre, 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 nel I semestre e Prof. Migliardi nel II semestre; 6 CFU): OS, Programming.

Conoscenze e abilità da acquisire:

Per la prima parte A (Prof. Mariconda; 6 CFU): Lo scopo della prima parte del corso è 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. La seconda parte è suddivisa in due moduli: Modulo B: nel primo modulo (Prof. Conti; 3 CFU, I semestre): gli studenti saranno in grado di identificare, classificare, descrivere, spiegare e correlare i concetti chiave degli attacchi e delle difese informatiche. Modulo C: nel secondo modulo della seconda parte (Prof. Migliardi; 3 CFU, II semestre): Valutare i rischi a cui è esposto un sistema IT, Spiegare come funziona un attacco, Descrivere, spiegare e generalizzare le vulnerabilità del software, Evitare le insidie del software.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali in classe. Per la prima parte (CRYPTOGRAPHY) sono previste attività in aula, partecipazione attiva e saranno disponibili i video delle lezioni.

Contenuti:

La prima parte (Prof. Mariconda; 6 CFU) costituisce anche l'insegnamento di CRYPTOGRAPHY per i corsi ICT FOR INTERNET AND MULTIMEDIA, COMPUTER ENGINEERING, COMPUTER SCIENCE, MATHEMATICS, International Cybersecurity and Cyberintelligence. Fatti teorici di base: Aritmetica modulare. Numeri primi. Teorema piccolo di Fermat. Teorema del resto cinese. Campi finiti: ordine di un elemento e radici primitive. Test di pseudoprimalità. Test di Agrawal-Kayal-Saxena. Metodo RSA: prima descrizione, attacchi. Metodo di Rabin e la sua connessione con la fattorizzazione degli interi. Metodi di logaritmo discreto. Come calcolare il logaritmo discreto in un campo finito. Metodi elementari di fattorizzazione. Alcune osservazioni sul setaccio quadratico di Pomerance. Protocolli e algoritmi. Algoritmi crittografici fondamentali. Metodi simmetrici (storici, DES, AES). Metodi asimmetrici. Attacchi. Firma digitale. Generatori pseudocasuali (osservazioni). Scambio di chiavi, scambio di chiavi in tre passaggi, divisione del segreto, condivisione del segreto, trasmissione del segreto, marcatura temporale. Firme con RSA e logaritmo discreto. Simboli di Legendre e di Jacobi, legge di reciprocità quadratica e applicazioni alla crittografia. Per la seconda parte (Prof. Conti and Prof. (da determinare); 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:

gli studenti dell'insegnamento di CRYPTOGRAPHY devono sostenere solo la prima parte A, quelli di CYBERSECURITY AND CRYPTOGRAPHY: PRINCIPLES AND PRACTICES devono sostenere i moduli A, B, C. Per la prima parte A (CRYPTOGRAPHY, Prof. Mariconda; 6 CFU): Esame scritto, prova orale se ritenuta necessaria. In alternativa all'esame finale sono proposte prove intermedie e lavori individuali (esercizi, peer review). Per la seconda parte - modulo B (Prof. Conti, 3 crediti): Esame scritto, progetti assegnati da svolgere a casa, esame orale. - modulo C (Prof. Migliardi, 3 crediti): Esame scritto. Per ogni modulo lo studente può scegliere tra 5 date possibili per sostenere l'esame, senza nessun vincolo di esclusione tra l'una e l'altra, anche se la consegna del compito ad una prova successiva annulla la precedente. - **Modulo A:** 2 prove nella sessione invernale, 1 esame a giugno-luglio, 2 esami ad agosto-settembre. - **Modulo B:** 2 prove nella sessione invernale, 2 esami a giugno-luglio, 1 esame ad agosto-settembre. - **Modulo C:** 2 prove a giugno-luglio, 1 esame a settembre, 2 esami a gennaio-febbraio (dell'anno successivo). Il voto finale per gli studenti del corso CYBERSECURITY AND CRYPTOGRAPHY: PRINCIPLES AND PRACTICES è determinato dalla media ponderata dei tre esami parziali—A (Prima parte del corso, 6 crediti, I semestre), B (Primo modulo della seconda parte del corso, 3 crediti, I semestre) e C (Secondo modulo della seconda parte del corso, 3 crediti, II semestre)—in proporzione ai rispettivi crediti. Tutti e tre gli esami parziali devono essere completati all'interno dello stesso anno accademico (in particolare, gli esami dei Moduli A e B devono essere completati entro la fine di settembre, mentre l'esame del Modulo C può essere completato nella sessione invernale dell'anno successivo). Al termine di ogni sessione d'esame, i voti degli studenti che hanno superato tutte e tre le parti vengono automaticamente registrati su Uniweb (non è richiesta alcuna registrazione). Di norma, salvo altre indicazioni della commissione, gli studenti che rifiutano il voto finale dovranno ripetere tutti e tre i moduli.

Criteri di valutazione:

Per la prima parte (Prof. Mariconda; 6 CFU): Si prevedono due percorsi possibili: per chi frequenta e studia regolarmente è previsto un bonus costituito dagli esiti di valutazioni in itinere su lavori a casa singoli o di gruppo da utilizzare in un appello nella prima sessione di esami dopo il corso o nei tre esami parziali durante il corso, altrimenti la prova è costituita dal solo appello finale. Durante la prova scritta finale lo studente dovrà rispondere ad alcune domande relative al programma svolto e risolvere alcuni esercizi dimostrando di aver compreso gli argomenti del corso. Il voto costituisce l'esito finale per l'insegnamento di Cryptography. Per la seconda parte (Prof. Conti and Prof. Migliardi; 3+3 CFU): Valutazione sia della competenza teorica che della capacità operativa di applicare quanto appreso a un caso reale. Il voto per gli studenti che seguono l'intero insegnamento di CYBERSECURITY AND CRYPTOGRAPHY: PRINCIPLES AND PRACTICES è dato dalla media pesata in proporzione ai CFU dei voti della prima parte e della seconda parte.

Testi di riferimento:

Hoffstein J., Pipher J., Silverman J., An introduction to mathematical cryptography (2nd ed.). New York: Springer, 2014 Stallings, William; Brown, Lawrie, Computer security principles and practice. Boston [etc.]: Pearson, 2015 Pflieger, Charles P.; Pflieger, 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

Eventuali indicazioni sui materiali di studio:

Per la prima parte (6 CFU) il testo di riferimento è: Hoffstein J., Pipher J. e Silverman J. - An introduction to mathematical cryptography. 2nd ed. Undergraduate Texts in Mathematics. New York, NY: Springer (2014) Per la prima parte sono testi di consultazione e approfondimento: 1) N. Koblitz - A Course in Number Theory and Cryptography -Springer, 1994. 2) H. Knospe - A Course in Cryptography - American Mathematical Society, 2019. 3) R. Crandall, C. Pomerance - Prime numbers: A computational perspective - Springer, 2005. 4) B. Schneier - Applied Cryptography - Wiley, 1994. 5) A. Languasco, A.Zaccagnini - Manuale di Crittografia - Hoepli Editore, 2015. (Italian).

CYBERPHYSICAL SYSTEMS AND IOT SECURITY

Titolare: Dott. ALESSANDRO BRIGHENTE

Mutuato da: Laurea magistrale in International Cybersecurity and Cyberintelligence (Ord. 2023)

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Conoscenze base di architettura degli elaboratori e principali protocolli di rete (TCP, UDP, IP). Capacità di analizzare codice, capire il suo funzionamento, e modificarlo in base alle necessità.

Conoscenze e abilità da acquisire:

Alla fine del corso, lo studente sarà in grado di - Analizzare un flusso di controllo e capirne le operazioni fondamentali, con particolare riferimento al protocollo CAN. Capacità di implementare attacchi a livello controllo e a livello rete. Capacità di analizzare il traffico CAN bus e inferire informazioni sul suo funzionamento. - Implementare semplici controllori e testarne la sicurezza. - Analizzare un programma ladder logic per PLC e capirne il funzionamento. Implementare attacchi in grado di alterarne il funzionamento e progettare programmi sicuri. - Comprendere il funzionamento dei principali protocolli industriali, implementare attacchi ad integrity ed availability, e sviluppare contromisure. - Comprendere il funzionamento dei protocolli di posizionamento di droni e procedure di fail safe. Implementare attacchi di GPS spoofing per deviare le traiettorie. - Implementare protocolli di remote attestation per dispositivi IoT ed analizzarne le performance.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali comprendenti sia teoria che laboratori. In particolare, l'attività di laboratorio ha l'obiettivo di fornire allo studente le conoscenze basilari sugli attacchi e tecniche di prevenzione.

Contenuti:

Fundamentals - What is a Cyber-Physical System - Security Requirements in CPS Automotive Security - The CAN bus protocol - Error handling in CAN bus and bus-off attack - Network attacks on CAN bus - Keyless cars security and attacks to distance bounding protocols Autonomous Driving - Introduction to controllers - Levels of automation and modes of operation - Attacks on controllers and countermeasures Industrial Control Systems - Industrial Control Network Protocols - PLC and their functioning - Attacks and countermeasures to industrial control systems Drones - Drone components and basic functioning - Protocols for drone location and fail-safe procedures - Drone detection systems Internet of Things - Network protocols for the internet of things - Remote attestation - Intrusion and anomaly detection

Modalità di esame:

La totalità dei punti dell'esame, è suddivisa secondo i seguenti criteri: 40%: report di metà corso su un lavoro di implementazione di attacchi e contromisure su un topic a scelta della prima parte del corso 40%: report di fine corso su un lavoro di implementazione di attacchi e contromisure su un topic a scelta della seconda parte del corso 20%: esame teorico finale (10 domande a scelta multipla)

Criteri di valutazione:

I criteri di valutazione riflettono la padronanza delle conoscenze e abilità acquisite dallo studente secondo la sezione "Conoscenze e abilità d'acquisire". L'esame finale valuta la conoscenza dei concetti base introdotti durante il corso.

Testi di riferimento:

Walid M., et al., Cyber-Physical Systems: A Model-Based Approach. : Springer, 2021 E.D. Knapp, Industrial Network Security. : Elsevier, 2011

Eventuali indicazioni sui materiali di studio:

Durante le lezioni verranno forniti articoli scientifici a supporto della didattica. Additional Material Saqib Ali, Taiseera Al Balushi, Zia Nadir, Omar Khadeer Hussain: Cyber Security for Cyber Physical Systems. Springer International Publishing (2018) Edward J. M. Colbert, Alexander Kott: Cyber-security of SCADA and Other Industrial Control Systems. Springer International Publishing (2016)

DATA MINING

Titolare: Prof. LIVIO FINOS

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 32A+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 dati reali tramite l'utilizzo del software R. Sebbene non sia vincolante, gli studenti sono invitati a partecipare alle attività di gruppo (Team-based Learning) proposte durante il corso.

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, previsione, selezione del modello - Metodi di classificazione: regressione logistica, analisi discriminante lineare ed estensioni - Criteri per la selezione del modello: validazione incrociata, R2 aggiustato, AIC, BIC, selezione automatica - Linear Model Selection and Regularization - Regressione semiparametrica: splines di regressione, splines di liscio, modelli additivi generalizzati - Tree-Based Methods

Modalità di esame:

Esame scritto e progetto

Criteri di valutazione:

La prova d'esame d'esame ha lo scopo di 1) 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; 2) valutare le capacità di utilizzo delle funzionalità del software R per condurre una completa analisi di insiemi di dati reali; 3) 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

DEEP LEARNING (MATICOLE DISPARI)

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, e il Transformer. Infine si tratteranno autoencoder e modelli generativi deep. Inoltre, gli studenti dovranno consegnare homework pratici che coprono i contenuti dell'insegnamento.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali.

Contenuti:

La tematiche 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;

Modalità di esame:

Lo studente deve superare un esame scritto. Per essere ammesso all'esame lo studente deve aver consegnato ed ottenuto valutazione positiva tutti gli homework previsti dall'insegnamento.

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.

Testi di riferimento:

Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron, Deep Learning. Cambridge: MA, MIT Press, 2016

Eventuali indicazioni sui materiali di studio:

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

DEEP LEARNING (MATRICOLE PARI)

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, e il Transformer. Infine si tratteranno autoencoder e modelli generativi deep. Inoltre, gli studenti dovranno consegnare homework pratici che coprono i contenuti dell'insegnamento.

Attività di apprendimento previste e metodologie di insegnamento:

L'insegnamento prevede lezioni frontali.

Contenuti:

La tematiche 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.

Modalità di esame:

Lo studente deve superare un esame scritto. Per essere ammesso all'esame lo studente deve aver consegnato ed ottenuto valutazione positiva tutti gli homework previsti dall'insegnamento.

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.

Testi di riferimento:

Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron, Deep Learning. Cambridge: MA, MIT Press, 2016

Eventuali indicazioni sui materiali di studio:

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

ECONOMICS AND MANAGEMENT OF INNOVATION

Titolare: Prof.ssa SILVIA RITA SEDITA

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 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 al manuale indicato.

Contenuti:

il corso si propone di affrontare e approfondire i seguenti temi: - Il ruolo dell'innovazione nell'evoluzione del sistema capitalistico. - Gli ecosistemi innovativi - 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 - Seventh edition. New York: McGraw-Hill, 2023

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

FINAL EXAM

Titolare: da definire

Periodo: Il anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: ; 33,00

FORMAL METHODS FOR CYBER-PHYSICAL SYSTEMS

Titolare: Prof. DAVIDE BRESOLIN

Mutuato da: Laurea magistrale in International Cybersecurity and Cyberintelligence (Ord. 2023)

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, logica. 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, e di comprenderne e analizzarne i risultati. 4. Capacità di utilizzare algoritmi e strumenti per la sintesi automatica di controllori, e di comprenderne e analizzarne i risultati.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso comprende lezioni frontali, attività di laboratorio e assignment da svolgere fuori dalle ore di lezione. Le lezioni frontali introducono le conoscenze di base e gli argomenti teorici. Le attività di laboratorio permettono di provare le metodologie e gli strumenti appresi su semplici casi di studio. Gli assignment richiedono l'implementazione di soluzioni originali e la loro applicazione a casi di studio di media complessità.

Contenuti:

Sistemi ciberfisici: definizione e caratteristiche chiave. Modelli formali per sistemi ciberfisici: modelli sincroni e asincroni. Verifica dei sistemi ciberfisici: proprietà di sicurezza e liveness, model checking, algoritmi enumerativi e tecniche simboliche. Sintesi di controllori per sistemi ad eventi discreti.

Modalità di esame:

Esame scritto per la parte di teoria. Per la parte pratica, due assignment da svolgere e consegnare durante il semestre di lezione, o in alternativa, un 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 5. Capacità di utilizzare strumenti per la sintesi automatica di controllori

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 STEM. 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 dei Docenti.

FUNCTIONAL LANGUAGES

Titolare: Dott. FRANCESCO GAVAZZO

Periodo: I anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Conoscenza base della logica formale e maturità matematica. Esperienze pregresse di programmazione funzionale possono aiutare, ma non sono necessarie.

Conoscenze e abilità da acquisire:

Il corso si focalizza sull'analisi matematica dei linguaggi di programmazione funzionali. Dopo una breve introduzione pratica alla programmazione funzionale, analizzeremo formalmente linguaggi funzionali core, studiandone matematicamente la sintassi e la semantica dinamica (semantica operativa) e statica (tipaggio). Un focus particolare sarà sugli strumenti matematici necessari all'analisi e design dei linguaggi (induzione, sistemi di inferenza, algebra, punti fissi, riscrittura)

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali alla lavagna per Parte 1-4 e live demo per la Parte 0

Contenuti:

Parte 0: - Crash course su Standard ML Parte 1: - Analisi della sintassi del frammento funzionale core di SML - Matematica della Sintassi (algebra, induzione, punti fissi) - Sintassi con cattura di variabili Parte 2 - Semantica operativa base - Cenni di teoria della riscrittura - Equivalenza tra programmi Parte 3 - Teoria dei tipi - Tipi semplici - Tipi polimorfi e ricorsivi Parte 4 (tempo permettendo) - Aspetti avanzati - Tipi lineari e sotto-strutturali - Semantica meccanizzata

Modalità di esame:

Esercizi per casa Esame orale

Criteri di valutazione:

Apprendimento teorico e pratico attraverso esercizi per casa e discussioni orali

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

Tutto il materiale è disponibile gratuitamente sul web e verrà comunicato a inizio corso

GAME THEORY

Titolare: Dott. GIOVANNI PERIN

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 basilare di teoria della probabilità.

Conoscenze e abilità da acquisire:

L'insegnamento prevede l'acquisizione delle seguenti conoscenze e abilità: - 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. - sapere applicare i concetti della teoria dei giochi a scenari pratici in ambito informatico e di analisi dei dati, soprattutto per metodologie con intelligenza computazionale, multi-agente e distribuite.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni convenzionali con il supporto di slide. Prevista interazione su piattaforma moodle. Possibile utilizzo di video registrati.

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 (cenni) Giochi cooperativi, il nucleo, il valore di Shapley (cenni) Allocazione delle risorse Utilità, scelte e paradossi Giochi potenziali, coordinazione Algoritmi bio-inspired Giochi evolutivi Reti cognitive Selfish routing Age of information Sistemi multi-input con teoria dei giochi

Modalità di esame:

Test scritto obbligatorio a libro aperto, con diversi problemi di game theory su argomenti toccati durante il corso. Questo determina il punteggio base della votazione per un massimo di 27 punti. I frequentanti possono in aggiunta sviluppare un progetto in gruppi di 1-3 persone, su argomenti del corso. Adesione al progetto e argomento dello stesso sono concordati con il docente durante il corso. Il progetto va consegnato entro un mese dalla fine delle lezioni. Se lo scritto è sufficiente (maggiore di 15) si può registrare il voto finale: - senza progetto: punteggio conseguito allo scritto + 3 punti - con progetto: punteggio conseguito allo scritto + valutazione del progetto che va da 0 a 8 punti

Criteri di valutazione:

Ogni domanda nei test scritti ha valutazione numerica comunicata in anticipo. Il punteggio finale dello scritto è la somma numerica dei punteggi individuali delle domande (max 27). Studenti che non svolgono il progetto ricevono ulteriori 3 punti per arrivare fino a un massimo di 30. Il progetto viene invece valutato da 0 a 8 punti, sommati al punteggio dello scritto. Un punteggio di 30 e lode è assegnato agli studenti il cui punteggio totale è maggiore o uguale di 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 capacità di problem solving - la capacità di discussione e verifica ex-post della soluzione trovata Nella valutazione del progetto (se presente) vengono tenuti in considerazione: - l'originalità della proposta e la pertinenza con le tematiche del corso - la capacità di lavoro di gruppo e la presenza di singoli contributi attribuibili ai partecipanti al progetto - la capacità di trarre conclusioni significative dal punto di vista scientifico grazie alle metodologie apprese nel corso

Testi di riferimento:

S. Tadelis, Game Theory: An Introduction. : Princeton, 2013 A. MacKenzie, L. Da Silva, Game Theory for Wireless Engineers. Morgan & Claypool: 2006, N. Nisan, T. Roughgarden, E. Tardos, V. V. Vazirani, Algorithmic game theory. : Cambridge Univ. Press, 2007 R. Lucchetti, A Primer in Game Theory. Bologna: Esculapio, L. Badia, T. Marchioro, Game theory. A handbook of problems and exercises. Bologna: Esculapio, 2022

Eventuali indicazioni sui materiali di studio:

Il corso segue la trattazione standard in ambito accademico per la presentazione delle basi della teoria dei giochi. Pertanto, il libro di Tadelis (integrato con materiale fornito a lezione/moodle) è sufficiente come riferimento generale, ma è equivalente ad ogni altro libro di introduzione alla teoria dei giochi. Per gli esercizi, si consiglia il Badia-Marchioro che contiene esercizi analoghi a quelli d'esame. Per le applicazioni e progetti, si può fare riferimento al libro di MacKenzie-DaSilva, ma va bene ogni materiale di ricerca trovato in rete.

IT SERVICE MANAGEMENT

Titolare: Dott. FRANCESCO CLABOT

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

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

Prerequisiti:

Non ci sono prerequisiti: tutte le informazioni necessarie per comprendere gli argomenti trattati verranno fornite durante il corso. L'obiettivo è acquisire familiarità con la terminologia IT e l'esperienza lavorativa relativa all'IT.

Conoscenze e abilità da acquisire:

CHI DOVREBBE PARTECIPARE? Chiunque cerchi di capire come funziona ITSM e chiunque sia interessato ad entrare nel mondo professionale dove i problemi sono allineare l'IT con il business, controllare o ridurre i costi IT, migliorare la qualità del servizio IT e bilanciare le risorse IT nel modo più efficace. Chiunque intenda diventare un professionista IT, project manager IT, manager IT, team IT o membro del progetto, coordinatore, operatore di rete, analista di processi aziendali, architetto IT, consulente, integratore di sistemi, manager e help desk, pianificatori, fornitori di servizi gestiti, outsourcer, sviluppatori di applicazioni e altre posizioni relative all'IT.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali, esercitazioni in aula e discussione dei casi studio proposti. Pagina Moodle con lezioni registrate, approfondimenti, materiale video, testi vari e articoli di stampa.

Contenuti:

COSA SI IMPARA Il corso aiuterà gli studenti a comprendere: - Concetti chiave di gestione dei servizi IT - Come i principi guida ITIL possono aiutare e l'organizzazione ad adottare e adattare la gestione dei servizi - Le 4 dimensioni della gestione dei servizi - Lo scopo e le componenti del sistema di valore del servizio - Le attività della catena del valore del servizio e come le interconnettono - Lo scopo delle pratiche ITIL chiave - Le basi per sostenere l'esame di fondazione ITIL4 STRUTTURA DEL CORSO - Definizioni di gestione dei servizi IT; Servizio, Utilità, Garanzia, Cliente, Utente, Gestione del servizio, Sponsor - Concetti chiave della creazione di valore - Concetti chiave delle relazioni di servizio; offerta di servizi; fornitura di servizi; consumo di servizi; gestione dei rapporti di servizio - La natura, l'uso e l'interazione di 7 principi guida ITIL; Concentrarsi sul valore; Inizia da dove sei; Progredisci in modo iterativo con il feedback; Collaborare e promuovere visibilità; Pensa e lavora in modo olistico; Mantienilo semplice e pratico; Ottimizza e automatizza - Le 4 dimensioni della gestione dei servizi; Organizzazioni e persone; Informazione e tecnologia; Partner e fornitori; Flussi di valore e processi - Il sistema di valori del servizio ITIL - La catena del valore del servizio, i suoi input e output e il suo ruolo nel supportare i flussi di valore - Elementi della catena del valore del servizio; Pianificazione, miglioramento, coinvolgimento, progettazione e transizione, acquisizione/costruzione, fornitura e supporto - Dettaglio di come le seguenti pratiche ITIL supportano la catena del valore del servizio: Continual Improvement (including continual improvement model); Change control; Incident management; Problem Management; Service request management; Service desk; Service level management - Lo scopo delle seguenti pratiche ITIL: Information security management; Relationship management; Supplier management; Availability management; Capacity and performance management; Service configuration management; IT asset management; Business analysis; Service continuity management; Deployment management; Monitoring and event management; Release management

Modalità di esame:

Esame scritto e orale.

Criteri di valutazione:

La prova scritta valuta l'acquisizione da parte 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:

CONTENUTO NON PRESENTE

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: Dott. LUCIANO SERAFINI

Mutuato da: Laurea magistrale in Data Science (Ord. 2023)

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

E' opportuno che gli studenti abbiano una conoscenza di base di logica matematica e di statistica.

Conoscenze e abilità da acquisire:

Abilità cognitive. Al termine del corso, gli studenti saranno in grado di: - Comprendere i principi delle logiche per la rappresentazione e il ragionamento della conoscenza. - Comprendere i principi dell'apprendimento statistico relazionale. - Comprendere i principi dell'integrazione neuro-simbolica, ovvero come combinare logiche e apprendimento statistico relazionale per costruire sistemi di apprendimento e ragionamento in domini ibridi. Abilità pratiche. Gli studenti saranno in grado di: - Modellare problemi in logiche come la Logica Proposizionale, la Logica del Primo Ordine e la Logica Fuzzy. - Utilizzare algoritmi di rappresentazione della conoscenza e di ragionamento per trovare i modelli di una teoria logica. - Modellare problemi utilizzando approcci di apprendimento statistico relazionale come modelli grafici, reti logiche di Markov, Prolog probabilistico e reti logiche tensoriali. Abilità trasversali. Gli studenti svilupperanno capacità di pensiero analitico, capacità di problem solving e capacità di pensiero sistemico-logico.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso prevede 48 ore di lezione. Le lezioni saranno supportate da esercitazioni pratiche.

Contenuti:

(A) Logiche per la rappresentazione della conoscenza: (A.i) introduzione alle logiche proposizionali, sintassi, semantica, procedura decisionale. Soddisfabilità, soddisfabilità ponderata e migliore soddisfabilità. (A.ii) Logica del primo ordine, sintassi, semantica, risoluzione e unificazione. (A.iii) Logiche fuzzy, sintassi, semantica e ragionamento. (B) Apprendimento statistico relazionale: (B.i) Modelli grafici (B.ii) Reti logiche di Markov (B.iii) Prolog probabilistico, (B.iv) Reti logiche tensoriali

Modalità di esame:

Le conoscenze e le competenze degli studenti saranno valutate attraverso: - Un esame scritto: agli studenti verrà chiesto di risolvere esercizi simili a quelli illustrati nelle lezioni. - Un project work: agli studenti verrà chiesto di implementare e valutare uno degli algoritmi presentati nelle lezioni o di utilizzare uno strumento di inferenza logica/probabilistica per risolvere un problema pratico.

Criteri di valutazione:

Gli studenti saranno valutati in base alla conoscenza critica degli argomenti del corso. I criteri di valutazione consistono in: 50% Esame scritto: capacità di presentare e applicare il materiale studiato. 50% Lavoro di progetto: capacità di applicare il contenuto del corso a problemi pratici.

Testi di riferimento:

Russell, Stuart J.; Norvig, Peter; Russell, Stuart J., Artificial intelligence a modern approach. Harlow: Pearson Education, 2022 Stuart Russell, Unifying logic and probability. : Communications of the ACM, 2015

Eventuali indicazioni sui materiali di studio:

Il materiale di studio consiste in: - diapositive delle lezioni, - appunti delle lezioni.

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 per programmazione service oriented (Jolie) o linguaggi con un controllo statico dell'ownership (come Rust).

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. Proprietà 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 per programmazione orientata ai servizi (Jolie), linguaggi con un controllo statico dell'ownership (come Rust).

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/Concurrency>

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. L'esame prevede un progetto o, in sostituzione, attività durante il corso.

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, Ethem, 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. 2023)

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. - Spazi vettoriali, indipendenza lineare, base in spazi vettoriali - Prodotto scalare e ortogonalità - Tipi di matrici: diagonale, triangolare, simmetrica, definita positiva - Spettro di una matrice: autovalori e autovettori - Fattorizzazione di una matrice: LU, Cholesky, QR - Risoluzione di sistemi lineari Conoscenze di base di analisi matematica. - Norme e metriche in spazi vettoriali - Continuità 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 (principalmente Matlab o Python), 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:

Il corso si baserà principalmente su lezioni frontali in aula ed esercitazioni. Dispense e slides verranno rese disponibili sulla pagina web del corso.

Contenuti:

1. Ranking with eigenvectors • Recapitulation of important results from linear algebra • The vector iteration: calculation of dominant eigenpairs • Ranking of web pages: PageRank and Hits. 2. Numerical methods for large scale linear systems • Krylov subspace methods for large eigenvalue problem: Arnoldi and Lanczos • Krylov subspace methods for sparse systems of equations: GMRES and MINRES • Krylov subspace methods for the computation of matrix functions 3. Spectral graph theory • Graphs, the graph Laplacian, graph signals, the Cheeger constant • The graph Fourier transform, graph convolution, filtering and decomposition of graph signals • Dimensionality reduction with Laplacian eigenmaps • Centralities on graphs 4. Clustering algorithms • k-center clustering and hierarchical k-center clustering • The k-means algorithm • Spectral clustering 5. Low rank matrix approximations • Singular value decomposition (SVD): basic properties, numerical computation and best rank-k approximation • Principal component analysis and dimensionality reduction • Latent factor models and recommender systems • Matrix factorizations beyond SVD: Funk SVD and nonnegative matrix factorizations. 6. Multiway Data Analysis • Tensors and tensor decomposition • Higher order singular value decomposition (HOSVD): definition and numerical computation • Applications: Face Recognition Using tensor SVD and Tensor Data Fusion • Kernel methods for data analysis

Modalità di esame:

Esame scritto alla fine del corso. Esercizi a casa durante il periodo delle lezioni forniranno un ulteriore feedback e crediti aggiuntivi per l'esame finale.

Criteri di valutazione:

La prova scritta alla fine del corso mira a verificare la comprensione dei fondamenti teorici e delle competenze pratiche apprese durante il corso. La valutazione nella prova finale si baserà: - sulla comprensione dei modelli matematici e la teoria vista a lezione - sulla capacità di utilizzare i metodi numerici presentati in aula in maniera autonoma per la risoluzione di problemi concreti.

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 Daniela Calvetti; Erkki Somersalo, Mathematics of data science : a computational approach to clustering and classification : pbk. : Society for Industrial and Applied Mathematics, 2021

Eventuali indicazioni sui materiali di studio:

Dispense estese, slides e codici verranno fornite 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 base di ricerca operativa, programmazione lineare, 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. Richiami, 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 vicinati 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 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 Laboratori

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 scritta.

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:

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

Eventuali indicazioni sui materiali di studio:

Le slide del corso sono fornite sul sito web del corso

MOBILE SECURITY

Titolare: Prof.ssa ELEONORA LOSIOUK

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

Periodo: I anno, 1 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. 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:

Prima di ogni lezione, il docente pubblica un video in cui illustra gli argomenti della lezione. Gli studenti devono vedere il video prima di partecipare alla lezione. All'inizio della lezione, il docente rilascia un breve questionario per verificare se gli studenti abbiano compreso i concetti principali descritti nella lezione registrata. Il questionario viene somministrato attraverso la piattaforma Moodle. L'insegnante, quindi, risponde a qualsiasi dubbio o domanda. Il docente individua i gruppi di lavoro scegliendo i componenti di ciascun gruppo (i gruppi saranno diversi per ogni nuovo laboratorio) e rilascia il nuovo laboratorio. Anche se tutti i gruppi sono incoraggiati a svolgere il laboratorio, l'insegnante seleziona il gruppo che dovrebbe risolverlo e illustra la soluzione agli altri gruppi, facendo una presentazione una settimana dopo il rilascio del laboratorio. Durante la lezione successiva, il gruppo selezionato presenta la sua soluzione e risponde alle domande del docente o degli altri studenti. Se l'insegnante è soddisfatto della prestazione del gruppo, ogni membro del gruppo riceve un bonus che verrà sommato al voto ottenuto durante l'esame finale.

Contenuti:

Gli argomenti sono i seguenti: - Architettura interna del sistema operativo Android. - Componenti di un'app mobile (Activity, Service, Content Provider, Broadcast Receiver). - Tecniche di analisi delle app. - Tecniche di reverse engineering per app. - Valutazione della vulnerabilità delle app. - Tecniche di analisi statica e dinamica per app. - Sfruttamento della vulnerabilità delle app.

Modalità di esame:

L'esame finale consisterà in una serie di domande a risposta multipla su tutti gli argomenti del corso. Il bonus accumulato con la partecipazione durante il semestre verrà sommato al voto ottenuto all'esame. Poiché la partecipazione non è obbligatoria, uno studente può ottenere il voto massimo (es. 30L) anche senza frequentare il corso.

Criteri di valutazione:

Conoscenza dei concetti presentati durante il corso.

Testi di riferimento:

Elenkov, Nikolay, Android security internals : an in-depth guide to Android's security architecture. : No Starch Press, 2015

NATURAL LANGUAGE PROCESSING

Titolare: Prof. GIOVANNI DA SAN MARTINO

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00

Prerequisiti:

Sono richieste conoscenze di base di Analisi Matematica, Probabilità e Statistica, Algebra Lineare. È consigliata familiarità con la programmazione, nello specifico il linguaggio Python. È altresì consigliata familiarità con i concetti di base di Apprendimento Automatico.

Conoscenze e abilità da acquisire:

ABILITÀ COGNITIVE Gli studenti saranno in grado di: - comprendere ed utilizzare una terminologia appropriata - scegliere le tecniche più adatte per risolvere problemi di apprendimento automatico riguardanti il linguaggio naturale - analizzare in modo competente i risultati di propri esperimenti ed articoli in letteratura **ABILITÀ PRATICHE** Gli studenti saranno in grado di: - utilizzare librerie di apprendimento automatico e natural language processing per risolvere problemi di complessità crescente - utilizzare in modo critico gli strumenti collegando le varie azioni ai concetti teorici corrispondenti **COMPETENZE TRASVERSALI** Gli studenti svilupperanno capacità di pensiero critico, capacità di inquadrare un problema e di problem solving.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso offrirà: - Lezioni frontali, - Compiti individuali, - esercizi in laboratorio

Contenuti:

Gli argomenti del corso sono i seguenti: - Introduzione agli algoritmi di apprendimento automatico: una prospettiva applicativa; - Elaborazione di base di testo: sentence splitting, tokenisation, Part of Speech tagging, Named Entity Recognition, - Coreference Resolution - Syntactic Parsing, Semantic Role Labeling - Rappresentazione vettoriale di dati testuali: n-grams, word2vec, rappresentazione di frasi. - Generazione di testo: language models, sequence-to-sequence models - Applicazioni: machine translation, sentiment analysis, (community) question answering, news analysis.

Modalità di esame:

Le conoscenze degli studenti e le loro abilità saranno valutate attraverso: - Compiti - gli studenti risolveranno durante il corso dei brevi esercizi sugli argomenti visti a lezione - Un progetto – gli studenti applicheranno le loro conoscenze per risolvere un problema che riguarda il linguaggio naturale - Un esame orale – gli studenti dovranno rispondere a domande sul progetto e/o sugli argomenti visti a lezione

Criteri di valutazione:

Compiti: 25%, Progetto con report scritto: 40%, esame orale: 35%.

Testi di riferimento:

Eisenstein, Jacob, Introduction to natural language processing. Cambridge, Massachusetts London: The MIT Press, 2019 by Daniel Jurafsky; James H. Martin, Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition. : Pearson/Prentice Hall, 2009

Eventuali indicazioni sui materiali di studio:

Verranno resi disponibili i lucidi utilizzati a lezione ed ulteriore materiale per approfondire i concetti visti a lezione.

PROCESS MINING

Titolare: Prof. MASSIMILIANO DE LEONI

Mutuato da: Laurea magistrale in Data Science (Ord. 2023)

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" o nei corsi di laurea in Informatica o affini.

Conoscenze e abilità da acquisire:

Oggigiorno, tutte le medie e grandi imprese seguono procedure e processi standardizzati per fornire beni e servizi, al fine di garantire livelli di servizio e di efficienza più alti. La modellazione formale di questi processi viene utilizzata come input per sistemi informativi (per es. sistemi SAP o ERP) che automaticamente si autoconfigurano per funzionare in accordo con tali modelli di processo. Lo scopo del corso è quello di introdurre e studiare concetti, metodi e tecniche che riguardano la modellazione, l'analisi e il miglioramento di processi di business. Processi più efficienti portano a sistemi più efficienti, che a loro volta permettono alle organizzazioni di fornire migliori servizi e beni di più alta qualità. Il corso può concettualmente essere suddiviso in tre parti: 1) In una prima parte, il corso introduce i principi e i linguaggi alla base della modellazione dei processi. 2) In una seconda parte, la più estesa, il corso discute le tecniche di analisi dei processi aziendali basate sul Process Mining. Process Mining si pone l'obiettivo di utilizzare lo storico dei dati transazionali delle aziende (i log degli eventi) per analizzare e migliorare i processi aziendali. Il Process Mining utilizza le informazioni nei log degli eventi per scoprire come i processi vengono REALMENTE eseguiti, svelando le violazioni di protocolli e regolamenti, le inefficienze e tutte le criticità che mettono a rischio le esecuzioni stesse dei processi. Mentre le tecniche tradizionali di Business Intelligence si basano su interviste e questionari con gli attori dei processi, Process Mining è parte integrante di Data Science: la soggettività degli attori viene sostituita dall'oggettività delle conclusioni estratte dai dati effettivi di processo. 3) L'obiettivo ultimo del Process Mining è predisporre azioni concrete che possano portare a migliorare i processi, sia dal punto di vista di aderenza alle normative vigenti, sia dal punto di vista dell'efficienza. Nella terza parte del corso, gli "insight" ottenuti tramite Process Mining vengono utilizzati per costruire modelli di simulazioni dei processi aziendali che siano "gemelli digitali" dei processi reali. Tali "gemelli digitali" rispecchiano il comportamento dei processi reali, in tutti gli aspetti (caratteristiche e vincoli temporali, decisioni, dati manipolati, risorse, ecc.). Il vantaggio di avere modelli di simulazione (i gemelli digitali) risiede nel fatto che sia è possibile simulare variazioni di tali processi e misurare gli impatti sulle performance dei processi, senza dover veramente cambiare il processo reale, riducendo i costi e rischi di eseguire processi alternativi ancora più costosi e meno efficienti degli originali.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso consiste di lezioni, esercitazioni, sessioni di laboratorio e seminari. Le sessioni di laboratorio daranno l'opportunità di "sporcarsi le mani" e verranno prevalentemente utilizzati dataset reali per meglio comprendere la complessità di eseguire un progetto di Process Mining e di lavorare con dati di processo reali. I seminari verranno organizzati da aziende nazionali e internazionali del settore di Process Mining

Contenuti:

Il corso affronterà i seguenti argomenti: 1. MODELLAZIONE DEI PROCESSI 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 reti di Petri: Principi e Verifica. - Uso di Woped come software per modellare e verificare la correttezza dei modelli di processo 2. 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 Region 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 Social - Uso di software open-source and commerciali per l'analisi dei processi aziendali tramite Process Mining 3. SIMULAZIONE DEI PROCESSI AZIENDALI - Principi e Metodologie per la simulazione: Definizione di intervalli di Warm-up e Cool-down, gestione della stocasticità della simulazione tramite simulazioni multiple ed indipendenti, analisi dei risultati. - Definizione degli elementi di uno scenario di simulazione di processi aziendali: arrival rate, branching probabilities, resource pools, distribuzioni di probabilità delle durate delle attività, - Uso di BPMN per la specifica di modelli di simulazione di processi aziendali - Uso di Process Mining per la generazione di modelli di simulazione - Metodologie per la definizione scenari "What-if" - Uso di BIMP come software per la simulazione di processi aziendali.

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. Berlin: Springer, 2018 Aalst, Wil M. P. : van der, Modeling business processes a Petri net-oriented approach. Cambridge: MIT Press, 2011 van der Aalst, Wil, Process Mining Data Science in Action. Berlin, Heidelberg: Springer Berlin Heidelberg Imprint: Springer, 2016

Eventuali indicazioni sui materiali di studio:

Tre testi verranno utilizzati per le tre parti, come indicato nel box "Testi di riferimento" sottostante Questi sono complementati da (1) un eserciziaro e da (2) documenti relativi ad esercitazioni autoguidate per la parte di laboratorio. Sia (1) che (2) saranno messi a disposizione su Moodle.

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. Lo stile di insegnamento facilita e promuove

il ragionamento e l'approfondimento personale per colmare le eventuali lacune.

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 particolare viene 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:

Jane W.S. Liu, Real-Time Systems. : Prentice Hall, 2000 Alan Burns and Andy Wellings, Analysable Real-Time Systems - Programmed in Ada. : CreateSpace Independent Publishing Platform, 2016

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 . >.

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, concentrandosi 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, 2017

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.

SECURITY AND RISK: MANAGEMENT AND CERTIFICATIONS

Titolare: da definire

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

Periodo: 1 anno, 2 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, concentrandosi 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 di quanto studiato.

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:

Andrew S Tanenbaum, Maarten van Steen, Distributed Systems - Principles and paradigms. : Pearson Education International, 2017 Alan Burns and Andy Wellings, Concurrent and Real-Time Programming in Ada. : Cambridge University Press, 2007

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.

SELECTED TOPICS IN ALGORITHMS

Titolare: Prof. MICHELE SCQUIZZATO

Periodo: I anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A; 6,00

Prerequisiti:

Sebbene non ci siano prerequisiti formali, si consiglia di seguire questo corso dopo Advanced Algorithms.

Conoscenze e abilità da acquisire:

Questo è un corso orientato alla ricerca che presenta una selezione di argomenti sugli algoritmi, estendendo e integrando quelli solitamente trattati in un tradizionale corso di algoritmi di livello magistrale.

Attività di apprendimento previste e metodologie di insegnamento:

Lezioni frontali e discussione di articoli scientifici

Contenuti:

Un elenco di potenziali argomenti: algoritmi avanzati su grafi, strutture dati avanzate, algoritmi online, algoritmi di streaming, algoritmi paralleli e distribuiti, algoritmi cache-efficient, algoritmi di apprendimento automatico, algoritmi con previsioni, algoritmi dinamici, algoritmi sketching, complessità fine-grained, teoria algoritmica dei giochi, algoritmi sublineari

Modalità di esame:

Progetto

Criteri di valutazione:

La valutazione degli studenti si basa su un progetto che richiede di approfondire un argomento scelto dagli studenti stessi. Il progetto può assumere diverse forme: studiare alcuni algoritmi dalla letteratura recente, indagare su un problema aperto, scrivere un survey su di un tema di ricerca o implementare uno o più algoritmi per qualche problema del mondo reale. Tutti i progetti si concludono con una presentazione.

Testi di riferimento:

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio:

Dispense e articoli scientifici

SEMINARS AND OTHER ACTIVITIES

Titolare: Prof. DAVIDE BRESOLIN

Periodo: Il anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: ; 3,00

SHORT INTERNSHIP

Titolare: Prof. DAVIDE BRESOLIN

Periodo: Il anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: ; 3,00

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:

Esame finale su uno specifico caso di studio.

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

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 informatica, algoritmi di ottimizzazione e machine learning. Linguaggio di programmazione: Python.

Conoscenze e abilità da acquisire:

L'obiettivo di questo corso è fornire agli studenti una comprensione fondamentale della struttura e della funzione degli organismi viventi a livello molecolare, nonché delle principali metodologie computazionali utilizzate per lo studio degli stessi. Inoltre, il corso mira ad equipaggiare gli studenti con le competenze necessarie per sviluppare in modo autonomo un progetto di ricerca nel campo della bioinformatica strutturale. Gli studenti impareranno a identificare i problemi irrisolti nel campo, valutare lo stato attuale della ricerca, implementare modelli e metodi per affrontare tali problemi e analizzare criticamente i risultati ottenuti. Al termine del corso, gli studenti avranno acquisito le conoscenze e gli strumenti necessari per fare contributi significativi allo studio del materiale vivente, sviluppando al contempo le loro competenze di ricerca e di pensiero critico.

Attività di apprendimento previste e metodologie di insegnamento:

Il corso è strutturato attorno a una combinazione di lezioni, esercitazioni pratiche al computer e lo sviluppo di un progetto di ricerca, che sarà presentato e discusso criticamente. Le esercitazioni pratiche al computer sono progettate per familiarizzare gli studenti con le librerie software utilizzate nei progetti di bioinformatica. Attraverso queste esercitazioni, gli studenti acquisiranno esperienza pratica nell'utilizzo di questi strumenti e svilupperanno le loro competenze nella risoluzione di problemi reali nel campo. Il progetto di ricerca consentirà agli studenti di applicare i concetti e i metodi appresi nel corso a un problema specifico nel campo della bioinformatica. Gli studenti lavoreranno sui loro progetti in gruppo, con il supporto e il feedback del professore, e presenteranno i loro risultati alla classe. La presentazione del progetto richiederà una discussione critica, in cui gli studenti saranno chiamati a identificare i punti di forza e di debolezza dell'implementazione del loro software e a valutare i risultati della loro ricerca. Al termine del corso, gli studenti avranno acquisito una conoscenza approfondita degli strumenti e dei metodi utilizzati nella ricerca di bioinformatica, nonché la capacità di applicare questi concetti a problemi reali e di valutare criticamente i risultati ottenuti.

Contenuti:

Il corso è strutturato attorno ai seguenti argomenti: - Introduzione alla chimica, chimica organica e biochimica - Struttura e funzione delle proteine - Termodinamica del ripiegamento proteico - Metodi sperimentali per la determinazione della struttura - Basi di dati biologiche - Visualizzazione della struttura con PyMOL - Analisi e manipolazione di strutture proteiche con la libreria BioPython - Relazioni sequenza-struttura nelle proteine e classificazione strutturale - Proteine non globulari, disordine e proteine ripetute - Sovrapposizione e allineamento strutturale - Metodi per la predizione di feature strutturali - Metodi per la predizione della struttura, incluso l'esperimento CASP - Introduzione alla dinamica molecolare

Modalità di esame:

L'esame è suddiviso in tre parti separate, ognuna delle quali deve essere superata per completare con successo il corso. I pesi relativi di ciascuna parte sono indicati tra parentesi di seguito: - Test scritti durante il corso, che consistono in domande teoriche e pratiche - Progetto software - Presentazione del progetto e valutazione critica Per superare l'esame, gli studenti devono completare con successo tutte e tre le parti e ottenere un voto sufficiente in ciascuna di esse. I test scritti durante il corso valuteranno la comprensione dei concetti teorici affrontati nel corso, mentre il progetto software valuterà la capacità di applicare questi concetti a un problema pratico. La presentazione del progetto valuterà invece la capacità di comunicare i risultati della ricerca e di valutare il lavoro dei colleghi.

Criteri di valutazione:

La valutazione finale del corso sarà basata sui seguenti criteri: - Comprensione dei concetti e degli algoritmi presentati in classe - Capacità di applicare i concetti descritti a problemi reali - Abilità di pensiero critico, inclusa la capacità di scegliere i metodi e le alternative più appropriate - Capacità di sviluppare software riutilizzabile implementando nuovi modelli e metodi - Abilità di presentazione e discussione efficaci, inclusa la capacità di valutare criticamente e comunicare i risultati della ricerca.

Testi di riferimento:

Arthur M. Lesk, Introduction to Bioinformatics. : Oxford University Press, 2013 Bruce Alberts, Alexander Johnson, Julian Lewis, David Morgan, Martin Raff, Keith Roberts, and Peter, Molecular Biology of the Cell. : Garland Science, 2014 Donald Voet, Judith G. Voet, and Charlotte W. Pratt, Fundamentals of Biochemistry: Life at the Molecular Level. : Wiley, 2016 Theodore L. Brown, H. Eugene LeMay Jr., and Bruce E. Bursten, Chemistry: The Central Science. : Pearson, 2017 Philip E. Bourne and Helge Weissig, Structural Bioinformatics. : John Wiley & Sons, 2003 Jeremy M. Berg, John L. Tymoczko, and Lubert Stryer, Biochemistry. : W. H. Freeman, 2015

Eventuali indicazioni sui materiali di studio:

Sul sito E-learning vengono resi disponibili molti materiali per il corso. Questi comprendono le slide del corso (appena disponibili) e la letteratura usata per i progetti. Il software utilizzato durante le sessioni pratiche viene condiviso tramite Google Colab Notebooks.

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.

Conoscenze e abilità da acquisire:

Lo scopo di questo corso è quello di fornire --- una introduzione teorica alla TEORIA DEI TIPI DIPENDENTI; --- una introduzione all'uso di un PROOF-ASSISTANT alla dimostrazione; --- conoscere le applicazioni della teoria dei tipi all' INFORMATICA tra cui la +++ CORRETTEZZA dei PROGRAMMI FUNZIONALI; +++ loro VERIFICA FORMALE in un proof-assistant; --- conoscere le applicazioni della teoria dei tipi alla MATEMATICA tra cui lo +++ sviluppo di DIMOSTRAZIONI COSTRUTTIVE; +++ VERIFICA di CORRETTEZZA delle DIMOSTRAZIONI tramite proof-assistant; +++ ESTRAZIONE del contenuto COMPUTAZIONALE delle dimostrazioni di MATEMATICA COSTRUTTIVA.

Attività di apprendimento previste e metodologie di insegnamento:

Nel corso verranno erogate --- lezioni di TEORIA; --- esercitazioni su RISOLUZIONI di esercizi; --- lezioni su un PROOF-ASSISTANT; --- attività di LABORATORIO per l'uso del proof-assistant.

Contenuti:

Nel corso verranno introdotti --- i principali concetti di TEORIA dei TIPI; --- cenni di applicazioni in ambito +++ informatico; +++ matematico; +++ scientifico in genere. Lo studente verrà introdotto a comprendere i seguenti aspetti della sua multiforme natura: 1) analisi della natura COMPUTAZIONALE della teoria dei tipi vista come --- LAMBDA-CALCOLO tipato a' la Church; --- paradigma di un linguaggio di PROGRAMMAZIONE FUNZIONALE; --- sistema logico per TIPARE i PROGRAMMI con la loro specifica; --- calcolo di verifica di CORRETTEZZA dei PROGRAMMI in modo INTERATTIVO per mezzo di un PROOF-ASSISTANT. 2) analisi della natura INSIEMTISTICA della teoria dei tipi che la rende adatta --- a formalizzare le DIMOSTRAZIONI in MATEMATICA; --- ad estrarre il CONTENUTO COMPUTAZIONALE di DIMOSTRAZIONI in MATEMATICA COSTRUTTIVA. 3) analisi della natura PREDICATIVA delle costruzioni di TIPO DIPENDENTE a' la Martin-Löf tramite --- DEFINIZIONI INDUTTIVE; --- PARADOSSI generati da costruzioni NON predicative. 4) Interpretazione della LOGICA in teoria dei tipi tramite controllo di --- validità della logica e aritmetica INTUZIONISTA; --- casi in cui vale la logica CLASSICA. 5) Panorama su attuali teoria dei tipi analizzando: --- proprietà INTENSIONALI; --- proprietà ESTENSIONALI; --- DECIDIBILITÀ del TYPE CHECKING. 6) Meta' del corso includerà --- apprendimento laboratoriale all'USO di un PROOF-ASSISTANT come lo svedese AGDA basato sulla TEORIA DEI TIPI DIPENDENTI di Martin-Loef appreso nella parte teorica.

Modalità di esame:

L'accertamento di profitto avverrà tramite --- RISOLUZIONE PERSONALE di ESERCIZI assegnati e PROGETTO su proof-assistant; --- PROVA ORALE con docente.

Criteri di valutazione:

L'esame intende --- valutare le CONOSCENZE acquisite dallo studente sui temi del corso; --- valutare le capacità dello studente 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:

Saranno fornite --- NOTE SCRITTE dal docente sulla parte di teoria; --- VIDEO su parti teoriche e risoluzione di esercizi; --- VIDEO su apprendimento del proof-assistant.

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, così come di analisi matematica, probabilità e statistica, algebra lineare. È inoltre consigliabile una familiarità con concetti di base di apprendimento automatico.

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-modalità 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. : Prentice Hall,

Eventuali indicazioni sui materiali di studio:

Vengono rese disponibili le trasparenze utilizzate in aula.