UNIVERSITÀ DEGLI STUDI DI PADOVA

SCUOLA DI SCIENZE

Bollettino Notiziario

Anno Accademico 2014/2015

Laurea magistrale in Informatica (Ord. 2014)

Programmi dei Corsi

Curriculum: Corsi comuni

AMMINISTRAZIONE DI SISTEMA

(Titolare: Dott. FRANCESCO CLABOT)

Periodo: *I anno, 2 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *40A+8E; 6,00 CFU*

Prerequisiti :
Il corso non prevede particolari prerequisiti.
Conoscenze e abilita' da acquisire :
Il corso si propone di presentare agli studenti lâ€™organizzazione di un dipartimento ICT di una grande azienda. In particolare verranno trattate tematiche legate alle metodologie consolidate per lâ€™impostazione dei processi ICT (ITIL), le motivazioni che sono alla base delle scelte dei prodotti e tecnologie adottate (ROI, SLA, etc.), esempi concreti di architetture informatiche basilari oltre a vari case studies.
Attivita' di apprendimento previste e metodologie di insegnamento :
Lezioni frontali e laboratorio
Contenuti :
- La gestione dei servizi informatici (ITIL): i processi coinvolti nelle due aree della Gestione dei Servizi (Service Support e Service Delivery), la loro applicazione al ciclo operativo completo dei servizi, gli obiettivi fondamentali e perchÃ¨ questi sono stati standardizzati, breve dissertazione su ognuno dei 10 servizi coinvolti, esempi pratici.

- Modelli di servizio: considerazioni su ROI e SLA, approccio ed aspetti pratici.

- Il dipartimento IT: struttura ed organizzazione. Organigramma generale e breve dissertazione sui vari settori. Analisi accurata del "Service Desk" (come evoluzione dell'Help Desk).

- L'infrastruttura informatica: in verticale dal network ai servizi richiamando sempre i concetti esposti nella prima parte del corso. Esempi pratici (no laboratorio) e case study per mettere alla prova le capacitÃ deduttive degli studenti
Modalita' di esame :
Lâ€™esame finale consisterÃ in un test scritto composto da 40 domande a scelta multipla.
Criteri di valutazione :
Le conoscenze dello studente vengono valutate mediante un test a risposta multipla. La votazione finale prenderÃ in considerazione anche la qualitÃ dellâ€™attivitÃ di laboratorio condotta.
Testi di riferimento :
Jan Van Bon, Foundations of IT Service Management-based on ITIL. : Van Haren Publishing, 2007
Eventuali indicazioni sui materiali di studio :
Sul sito web del corso (link da http://www.netadm.it) sono presenti molti documenti scaricabili in formato digitale: case study, articoli divulgativi etc.

ANALISI NUMERICA

(Titolare: Prof. MARCO VIANELLO) - Mutuato da: Laurea magistrale in Astronomia (Ord. 2010)

Periodo: *I anno, 1 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *40A+16E; 6,00 CFU*

Prerequisiti :
Analisi matematica 1 e 2

Algebra lineare e geometria

Conoscenze e abilita' da acquisire :
Apprendere le basi del calcolo numerico in vista delle applicazioni in campo scientifico e tecnologico, con particolare attenzione ai concetti di errore, discretizzazione, approssimazione, convergenza, stabilita', costo computazionale
Attivita' di apprendimento previste e metodologie di insegnamento :
Sistema-floating point e propagazione degli errori:
errore di troncamento e di arrotondamento, rappresentazione floating-point dei reali, precisione di macchina, operazioni aritmetiche con numeri approssimati, condizionamento di funzioni, propagazione degli errori in algoritmi iterativi per esempi, il concetto di stabilita'

Soluzione numerica di equazioni non lineari:
metodo di bisezione, stima dell'errore col residuo pesato; metodo di Newton, convergenza globale, velocita' di convergenza, convergenza locale, stima dell'errore, altri metodi di linearizzazione; iterazioni di punto fisso

Interpolazione e approssimazione di funzioni e dati:
interpolazione polinomiale, interpolazione di Lagrange, errore di interpolazione, il problema della convergenza (controesempio di Runge), interpolazione di Chebyshev, stabilita' dell'interpolazione; interpolazione polinomiale a tratti, interpolazione spline; approssimazione polinomiale ai minimi quadrati

Integrazione e derivazione numerica:
formule algebriche e composte, convergenza e stabilita', esempi; instabilita' dell'operazione di derivazione, calcolo di derivate tramite formule alle differenze; il concetto di estrapolazione

Elementi di algebra lineare numerica:
norme di vettori e matrici, condizionamento di matrici e sistemi; metodi diretti: metodo di eliminazione gaussiana e fattorizzazione LU, calcolo del determinante, calcolo della matrice inversa, fattorizzazione QR, soluzione ai minimi quadrati di sistemi sovradeterminati; metodi iterativi: i metodi di Jacobi e Gauss-Seidel, struttura generale delle iterazioni stazionarie, precondizionamento; metodo delle potenze per il calcolo di autovalori estremali

Introduzione ai metodi alle differenze finite per equazioni differenziali:
i metodi di Eulero esplicito ed implicito, il metodo trapezoidale, convergenza e stabilita', sistemi stiff; equazione di Poisson 1d e 2d; metodo delle linee per l'equazione del calore

Laboratorio: implementazione e applicazione di codici numerici in Matlab
Contenuti :
Sistema floating-point e propagazione degli errori

Soluzione numerica di equazioni non lineari

Interpolazione e approssimazione di dati e funzioni

Integrazione e derivazione numerica

Elementi di algebra lineare numerica

Introduzione ai metodi alle differenze finite per equazioni differenziali
Modalita' di esame :
Prova orale
Testi di riferimento :
A. Quarteroni, F. Saleri, Introduzione al calcolo scientifico. : Springer,
A. Quarteroni, F. Saleri, Scientific computing with Matlab and Octave. : Springer,
G. Rodriguez, Algoritmi numerici. : Pitagora,
Eventuali indicazioni sui materiali di studio :
uno dei testi consigliati e dispense online del docente (www.math.unipd.it/~marcov/studenti.html)

ANALISI STATICA E VERIFICA DEL SOFTWARE

(Titolare: Prof. FRANCESCO RANZATO)

Periodo: *I anno, 1 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *64A; 8,00 CFU*

Prerequisiti :
Conoscenze di base dei linguaggi di programmazione. L'insegnamento non prevede propedeuticita.
Conoscenze e abilita' da acquisire :
Il corso mira ad introdurre metodi e strumenti per la specifica del comportamento, l'analisi statica e la verifica automatica dei programmi e, piÃ¹ 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.
Attivita' di apprendimento previste e metodologie di insegnamento :
L'insegnamento prevede lezioni frontali e la risoluzione in modo indipendente a casa di vari esercizi.
Contenuti :
- Semantica operazionale di programmi: Modellazione del comportamento operazionale dei programmi su una macchina di esecuzione mediante sistemi di regole di derivazione.

- Semantica denotazionale di programmi: Modellazione del comportamento input/output dei programmi mediante la teoria degli ordini parziali e dei punti fissi.

- Analisi statica di programmi mediante interpretazione astratta: L'interpretazione astratta Ã¨ una notoria tecnica basata su una approssimazione della semantica denotazionale dei programmi che permette di specificare le proprietÃ dei programmi deducibili mediante analisi statica e di provarne la correttezza.

- 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 Ã¨ 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 per ottimizzare un programma.

- Verifica di sistemi software mediante model checking: Il model checking e` una tecnica per la verifica automatica di proprietÃ di correttezza di un sistema software, dove la correttezza Ã¨ specificata mediante logiche temporali. Gli inventori del model checking sono stati premiati con il prestigioso Turing Award (noto come il "Premio Nobel" dell'informatica) nel 2007.
Modalita' di esame :
Esame orale, tipicamente suddiviso in tre parti distinte.
Criteri di valutazione :
L'esame orale verte su vari esercizi che lo studente deve svolgere in modo indipendente a casa.

Testi di riferimento :
H. Riis Nielson, F. Nielson, Semantics with Applications: A Formal Introduction. : Wiley, 1992
Eventuali indicazioni sui materiali di studio :
Le slide utilizzate a lezione verranno distribuite.

APPRENDIMENTO AUTOMATICO

(Titolare: Prof. ALESSANDRO SPERDUTI)

Periodo: *I anno, 1 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *40A+8L; 6,00 CFU*

Prerequisiti :
Ã opportuno avere familiaritÃ con le conoscenze matematiche relative al Calcolo delle ProbabilitÃ e all'Analisi di funzioni multivariate. Inoltre Ã¨ consigliabile avere conoscenze di base relative alla Programmazione e all'Intelligenza Artificiale.
L'insegnamento non prevede propedeuticitÃ .
Conoscenze e abilita' da acquisire :
In questo insegnamento si presentano alcuni dei concetti fondamentali che caratterizzano l'Apprendimento Automatico, cioÃ¨ quella classe di tecniche ed algoritmi che a partire da dati empirici permettono di aquisire nuova conoscenza, oppure di correggere e/o raffinare conoscenza giÃ disponibile. Tali tecniche sono particolarmente utili per problemi per cui Ã¨ 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 o sono 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.

Attivita' 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 capacitÃ 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 di Concetti:
ComplessitÃ dello Spazio delle Ipotesi; Misure di ComplessitÃ ; 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 e Selezione di Variabili Categoriche; Model Selection, Holdout, CrossValidation, LeaveOneOut CV; Criteri Esterni e Interni per Valutare un Sistema di Clustering; Sistemi di Raccomandazione: Tipologie, Approcci, Misure di Valutazione.
Modalita' di esame :
Lo studente deve superare un esame scritto e, se ritenuto necessario dal docente, un esame orale.
Criteri di valutazione :
Il testo dell'esame scritto contiene alcune domande che consentono di valutare il livello di apprendimento delle nozioni impartite durante l'insegnamento e la capacitÃ 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 attivitÃ svolte in laboratorio informatico. Tali domande hanno lo scopo di valutare se lo studente ha sviluppato la capacitÃ di applicare le nozioni apprese durante l'insegnamento.
Nel caso in cui la valutazione dello scritto non risulti soddisfacente per lo studente, il docente puÃ² integrare l'esame scritto con un esame orale per meglio verificare la preparazione dello studente.
Testi di riferimento :
Tom Mitchell, Machine Learning. : McGraw Hill, 1998
Ethem Alpaydin, Introduction to Machine Learning. : Cambridge University Press, 2010
Eventuali indicazioni sui materiali di studio :
Vengono rese disponibili, come riferimento, i lucidi utilizzati a lezione.

BIOINFORMATICA

(Titolare: Prof. GIORGIO VALLE)

Periodo: *I anno, 1 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *40A+8E; 6,00 CFU*

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 Ã¨ in lingua inglese, quindi Ã¨ necessario avere una buona conoscenza dell'inglese scritto e parlato.
Conoscenze e abilita' da acquisire :
Il Corso Ã¨ 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 Ã¨ accompagnato da esercitazioni pratiche in cui gli studenti applicheranno metodi bioinformatici per analizzare dati genomici.
In considerazione della complessitÃ della materia e in accordo con i descrittori di Dublino, particolare attenzione sarÃ dedicata affinchÃ© gli studenti acquisiscano la capacitÃ di integrare le conoscenze e gestire la complessitÃ dei problemi trattati, nonchÃ© di formulare giudizi sulla base di informazioni limitate e spesso frammentarie.
Attivita' di apprendimento previste e metodologie di insegnamento :
Il corso sarÃ tenuto con lezioni frontali e con esercitazioni pratiche. SarÃ stimolata la discussione in classe.
Contenuti :
Questo Ã¨ un corso di 6 crediti: cinque di lezioni ed uno di attivitÃ 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 Ã¨ un'approfondita introduzione alla Biologia, presentata come una disciplina scientifica centrata sull'Informazione. I meccanismi che facilitano la trasmissione e l'evoluzione dell'informazione biologica saranno presi come spunto per introdurre alcuni problemi della biologia che richiedono approcci computazionali e strumenti bioinformatici.
La seconda parte del corso descrive i principali algoritmi utilizzati per allineare sequenze biologiche, inclusi quelli sviluppati per il sequenziamento di DNA di ultima generazione. Sono inoltre descritti gli algoritmi utilizzati per l'assemblaggio "de novo" di genomi.
Infine, la terza parte del corso copre alcuni aspetti della bioinformatica relativi alla genomica funzionale, come l'analisi del trascrittoma, le predizione e annotazione genica, la ricerca di pattern e motivi per la predizione delle strutture proteiche. Inolre viene discusso il ruolo della bioinformatica nell'analisi di genomi individuali e nella medicina personalizzata.
Modalita' di esame :
L'esame sarÃ orale, ma un continuo monitoraggio sarÃ attuato durante l'intera durata del corso per verificare la comprensione degli studenti.
Criteri di valutazione :
Nell'esame finale gli studenti dovranno dimostrare una compprensione sistematica del settore e dovranno sapersi destreggiare con i metodi della ricerca associati ad esso. Inoltre gli studenti dovrebbero 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. Materiale didattico con gli approfondimenti di quanto spiegato a lezione sarÃ disponibile sul sito web del docente: http://didattica.cribi.unipd.it/genomica/bioinfoforinfo/.

BIOINFORMATICA 2

(Titolare: Prof. SILVIO TOSATTO)

Periodo: *I anno, 1 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *32A+16E; 6,00 CFU*

Prerequisiti :
Conoscenze base di algoritmi di ottimizzazione e machine learning. Linguaggi di programmazione C++ e/o Java.
Conoscenze e abilita' da acquisire :
Il corso intende comunicare le conoscenze di base sulla struttura e funzione della materia vivente nonchÃ© i principali metodi computazionali per il loro studio. Inoltre intende permettere allo studente lo svolgimento autonomo di un progetto di ricerca in bioinformatica, definendo lo stato dellâ€™arte per un problema aperto e un tentativo di risolverlo con lo sviluppo di software che estenda librerie esistenti e la valutazione critica dei risultati ottenuti.
Attivita' di apprendimento previste e metodologie di insegnamento :
Il corso si compone di lezioni frontali, esercitazioni pratiche al computer, contributo alle dispense, sviluppo di un progetto e presentazione dello stesso con discussione critica. Le esercitazioni servono per familiarizzare lo studente con le librerie software da usare per un progetto bioinformatico relativo ad un problema attuale diverso per ogni gruppo. La presentazione del progetto richiede una presentazione davanti alla classe e successiva discussione in cui far emergere i punti di forza e debolezza del software implementato. Il contributo alle dispense serve per ampliare il materiale del corso prodotto dagli studenti.

Contenuti :
Il corso si compone di due parti:
1) Introduzione alla materia vivente (2 CFU):
1.1) Cenni di chimica organica
1.2) Interazioni deboli ed energetica
1.3) Struttura e funzione di DNA e proteine
1.4) Lipidi, membrane e trasporto cellulare

2) Biochimica computazionale (4 CFU):
2.1) Banche dati biologiche
2.2) Librerie software e concetti per allineamenti di sequenza, profili e ricerca in banche dati
2.3) Relazione sequenza â€“ struttura â€“ funzione nelle proteine e classificazione
2.4) Metodi per la predizione della struttura delle proteine da sequenza. Lâ€™esperimento CASP.

2.5) Metodi per la predizione di funzione delle proteine. Lâ€™esperimento CAFA.
2.6) Cenni di biologia delle reti e dei sistemi.
2.7) Correlazione genotipo â€“ fenotipo. Lâ€™esperimento CAGI.
Modalita' di esame :
L'esame si compone di quattro parti separate, che devono essere superate tutte: (i valori tra parentesi indicano i pesi per il voto complessivo)
1) Contributo alle dispense del corso (ca. 15%)
2) Progetto software (ca. 40%)
3) Presentazione del progetto con valutazione critica (ca. 20%)
4) Esame finale scritto con domande di calcolo, aperte brevi e lunghe (ca. 25%).
Criteri di valutazione :
Viene valutata:
1) la comprensione di concetti e gli algoritmi presentati a lezione
2) la capacitÃ di applicare le nozioni fornite a lezione su problemi reali
3) la capacitÃ critica di saper utilizzare i metodi nei modi piÃ¹ opportuni, scegliendo tra le alternative possibili
4) la capacitÃ di sviluppare software riutilizzabile estendendo librerie esistenti
5) la capacitÃ espositiva e di discussione critica
Testi di riferimento :
K.C. Mathews, K.E. Van Holde, K.G. Ahern, Biochimica (3Â° edizione). : Casa Editrice Ambrosiana, 2004
S. Pascarella, A. Paiardini, Bioinformatica. : Zanichelli, 2011
Eventuali indicazioni sui materiali di studio :
Sul sito E-learning vengono resi disponibili molti materiali per il corso. Questi comprendono i lucidi del corso (appena disponibili) e le registrazioni audio (podcast), le dispense e la letteratura usata per i progetti. Le dispense scaricabili in formato PDF contengono oltre 300 pagine per facilitare lo studio.

CALCOLO PARALLELO

(Titolare: Prof. GIANFRANCO BILARDI) - Mutuato da:

Periodo: *I anno, 2 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *72A; 9,00 CFU*

Prerequisiti :
- Progetto ed analisi di algoritmi sequenziali
- Programmazione di algoritmi sequenziali
- Architetture degli elaboratori con processore seriale
- Teoria della computatione

Conoscenze e abilita' da acquisire :
- Il corso si propone di fornire un quadro teorico per la progettazione e l'utilizzo efficiente dei sistemi di calcolo parallelo, con applicazioni ai multiprocessori, alle schede grafiche (GPU), ai sistemi riconfigurabili (FPGA), e ai sistemi di supercalcolo.
- Le tematiche principali affrontate sono: progettazione edanalisi di algoritmi paralleli, analisi di architetture parallele, programmazione parallela (con attivita' di laboratorio), ottimizzazione congiunta di algoritmo e architettura nella realizzazione mediante circuiti integrati.

Attivita' di apprendimento previste e metodologie di insegnamento :
- L'insegnamento si basa su lezioni frontali e su attivita' di laboratorio di progammazione parallelo.
- Oltre a presentare i contenuti del corso, le lezioni dedicano spazio significativo al processo di sviluppo delle idee e del loro utilizzo innovativo, in contesti differenti da quello nel quale sono state apprese.
- I testi di esame delle annate precedenti vengono messi a disposizione degli allievi, per fornire un campione di problemi che mettono alla prova sia la conoscenza dei contenuti specifici del corso sia le capacita' generali di "problem solving".

Contenuti :
- L'evoluzione verso il parallelismo dei sistemi di calcolo ed il ruolo di algoritmi, linguaggi di programmazione, architetture degli elaboratori e tecnologie dei circuiti integrati.
- Parallelismo implicito, sfruttato da compilatori e processori. Parallelismo nell'organizzazione dei microprocessori: architetture superscalari e "very long instruction word". Accenni alle tecniche di "branch prediction", "register renaming" e "dynamic scheduling".
- Elementi di algoritmica parallela. Quantificazione del parallelismo di un algoritmo. Lavoro e cammino critico di un algoritmo. Legge di Brent. Progettazione ed analisi delle prestazioni di algoritmi paralleli. Algoritmi per vari problemi computazionali tra cui il calcolo di funzioni associative, il calcolo dell'evoluzione di sistemi dinamici finiti, le operazioni base dell'algebra lineare, la fusione e l'ordinamento di sequenze, la trasformata di Fourier, l'istradamento dei messaggi in una rete.
- Linguaggi di programmazione per il parallelismo. Introduzione a MPI.
- Struttura e funzionamento delle macchine parallele. Reti di processori. Topologie di interconnessione: "array" lineare, anello, "mesh", toro, "array" multidimensionali, ipercubo, "shuffle-exchange", "cube-connected cycles", albero, "fat-tree". Metriche di diametro e banda di una rete.
- Istradamento dei messaggi: tecniche di routing per varie topologie. Teoria dell'"embedding" e della simulazione tra macchine parallele. Metriche di carico, dilatazione e congestione di un "embedding". Sistemi di memoria gerarchica e distribuita.
- La complessita' delle realizzazione mediante circuiti VLSI ("Very Large Scale Integration"). Concetto di layout. Area e volume delle reti e loro relazione con le metriche di banda. Complessita' area-tempo dei problemi computationali. Reti universali.

Modalita' di esame :
E' prevista una prova scritta ed un eventuale approfondimento orale. Il voto d'esame tiene anche conto dei delle attivita' di laboratorio di progammazione parallela.

Criteri di valutazione :
La valutazione della preparazione dello studente si basa sul livello di padronanza dei concetti e dei metodi presentati nelle lezioni. Particolare enfasi viene posta sulla capacita' di risolvere problemi che richiedono un certo livello di creativita' per essere ricondotti ai concetti e metodi acquisiti nel corso.

Testi di riferimento :
D. Culler and J.P. Singh, Parallel Computer Architecture: A Hardware/Software Approach. San Francisco, California, USA: Morgan Kaufmann Publishers, 1998
Joseph Ja'Ja', An Introduction to Parallel Algorithms. : Addison Wesley, 1992
F. Thomson Leighton, Introduction to Parallel Algorithms and Architectures: Arrays - Trees - Hypercubes. San Francisco, California, USA: Morgan Kaufmann Publishers, 1992
Eventuali indicazioni sui materiali di studio :
Sebbene siano disponibili numerosi buoni testi su singoli aspetti del calcolo parallelo, la sintesi ed integrazione tra tali aspetti fornita dalle lezioni non e' immediatamente reperibile in letteratura. Si consiglia pertanto di seguire le lezioni e prendere note accurate. Alcuni testi saranno comunque indicati in seguito.

COMPUTABILITà E ALGORITMI

(Titolare: Prof. PAOLO BALDAN)

Periodo: *I anno, 2 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *64A+16E; 10,00 CFU*

Prerequisiti :
Il corso richiede familiaritÃ con alcuni concetti matematici di base, quali relazioni, funzioni, insiemi, cardinalitÃ , ordini parziali, principi di induzione.
Non ci sono corsi propedeutici.
Conoscenze e abilita' da acquisire :
Obiettivo del corso Ã¨ quello di avvicinare lo studente ai temi classici della teoria della calcolabilitÃ e di completare e approfondire le conoscenze algoritmiche fondamentali acquisite nella laurea di primo livello. Per la prima parte, 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'indecidibilitÃ e della ricorsione. Per la seconda parte si approfondiscono alcune tecniche algoritmiche per l'elaborazione di strutture fondamentali quali grafi, stringhe e oggetti geometrici, si studiano algoritmi multithread e randomizzati. A livello piÃ¹ generale, il corso mira ad implementare le capacitÃ di formalizzazione, ragionamento e problem solving dello studente.
Attivita' di apprendimento previste e metodologie di insegnamento :
Il corso prevede lezioni frontali ed esercizi.
Contenuti :
Il corso si articola in due parti, la prima focalizzata sulla teoria della computabilitÃ , e la seconda che approfondisce tematiche di natura prettamente algoritmica.

Per quanto riguarda la teoria della computabilitÃ saranno sviluppati i seguenti temi:

- Algoritmi ed il concetto di procedimento effettivo. Macchine a registri (URM). Funzioni parziali ricorsive. Equivalenze tra modelli di calcolo. UniversalitÃ 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. IndecidibilitÃ 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-Sheperdson. Primo teorema di ricorsione. Secondo teorema di ricorsione.

L'approfondimento delle tecniche algoritmiche si concentrerÃ su:

- Algoritmi su grafi. Visita in ampiezza e visita in profonditÃ . Ordinamento topologico. Componenti fortemente connesse.

- Algoritmi su stringhe. Algoritmi basati su confronti (Knuth,Morris e Pratt, di Boyer, Moore e Yao,Corasich). Algoritmi seminumerici (ShiftAnd e Fingerprint di Rabin,Karp). Alberi dei suffissi e algoritmo di Ukonnen per la loro costruzione in tempo lineare.

- Algoritmi Multithread.

- Algoritmi di Geometria Computazionale. Rappresentazione degli oggetti geometrici e algoritmi di base. Test di non intersezione tra segmenti. Involucro convesso: algoritmi di Graham e di Jarvis. Localizzazione di un punto in un piano suddiviso in regioni poligonali.

- Algoritmi randomizzati. Algoritmo di rendering. Algoritmo di routing.
Modalita' di esame :
L'esame si articola in una prova scritta, principalmente focalizzata sullo svolgimento di esercizi di teoria della computabilitÃ , e in una discussione orale sulle tecniche algoritmiche.

Criteri di valutazione :
La prova scritta contiene esercizi atti a verificare la capacitÃ dello studente di utilizzare nozioni e tecniche dimostrative apprese durante il corso, per la soluzione di problemi nuovi. La prova orale verifica la conoscenza ed il livello di approfondimento dei temi trattati a lezione, con la descrizione di nozioni e la riproduzione di dimostrazioni note.
Testi di riferimento :
Nigel Cutland, Computability. An Introduction to Recursive Function Theory. : Cambridge University Press, 1980
T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduzione agli Algoritmi e Strutture Dati (3a edizione). : McGraw-Hill Italia, 2010
Eventuali indicazioni sui materiali di studio :
Pagina web: http://www.math.unipd.it/~baldan/Computabilita

CRITTOGRAFIA

(Titolare: Prof. ALESSANDRO LANGUASCO)

Periodo: *I anno, 2 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *40A+8E; 6,00 CFU*

Prerequisiti :
Gli argomenti dei corsi di Algebra, Analisi I e Algoritmi (in particolare per le stime di complessita' computazionale).
Conoscenze e abilita' da acquisire :
Lo scopo del corso e' quello di offrire una panoramica delle basi teoriche necessarie per permettere uno studio critico dei protocolli crittografici usati oggigiorno in molte applicazioni (autenticazione, commercio digitale). Nella prima parte verranno esposti gli strumenti matematici di base (essenzialmente dalla teoria elementare ed analitica dei numeri) neceasari 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.
Attivita' di apprendimento previste e metodologie di insegnamento :
Lezione frontale.
Contenuti :
First Part: Basic theoretical facts: Modular arithmetic. Prime numbers. Little Fermat theorem. Chinese remainder theorem. Finite fields: order of an element and primitive roots. Pseudoprimality tests. Agrawal-Kayal-Saxena's test. RSA method: first description, attacks. Rabin's method and its connection with the integer factorization. Discrete logarithm methods. How to compute the discrete log in a finite field. Elementary factorization methods. Some remarks on Pomerance's quadratic sieve.
Second Part: Protocols and algorithms. Fundamental crypto algorithms. Symmetric methods (historical ones, DES, AES) . Asymmetric methods. Attacks. Digital signature. Pseudorandom generators (remarks). Key exchange, Key exchange in three steps, secret splitting, secret sharing, secret broadcasting, timestamping. Signatures with RSA and discrete log.
Modalita' di esame :
Scritto
Criteri di valutazione :
Durante la prova scritta lo studente dovra' rispondere ad alcune domande relative al programma svolto dimostrando di aver compreso gli argomenti del corso. Il massimo dei voti (30/30) verra' assegnato in presenza di un compito privo di errori. Il docente si riserva di fare alcune domande orali nel caso in cui sia necessario investigare ulteriormente la preparazione del candidato.
Testi di riferimento :
A. Languasco e A. Zaccagnini, Introduzione alla Crittografia. Milano: Hoepli, 2004
Eventuali indicazioni sui materiali di studio :
Utilizzeremo i seguenti testi:
1) A.Languasco, A.Zaccagnini - Introduzione alla Crittografia - Hoepli Editore, 2004. (italian).
2) N.Koblitz - A Course in Number Theory and Cryptography, Springer, 1994.
3) R.Crandall, C.Pomerance, - Prime numbers: A computational perspective - Springer, 2005.
4) B. Schneier - Applied Cryptography - Wiley, 1994

DATA MINING

(Titolare: Prof. BRUNO SCARPA)

Periodo: *I anno, 2 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *34A+16L; 6,00 CFU*

Prerequisiti :
Conoscenze di Informatica di base, Basi di Dati
Conoscenze e abilita' da acquisire :
Il corso intende fornire una panoramica sui concetti e sulle metodologie e strumenti avanzati di analisi di grandi quantitÃ di dati, spesso usate come supporto al processo di decisione aziendale.
Attivita' di apprendimento previste e metodologie di insegnamento :
Lezioni frontali, laboratori con analisi di dati reali
Contenuti :
- L'analisi dei dati come strumento di supporto per le decisioni e la Business Intelligence. Motivazioni e contesto per il data mining.
- I modelli statistici: modelli lineari e GLM, la stima e l'adattamento ai dati
- Nozioni generali per il data mining: contrasto tra aderenza ai dati e complessitÃ del modello ovvero contrasto tra distorsione e varianza, tecniche generali per la selezione del modello (AIC, BIC, convalida incrociata, oltre ai test statistici classici), suddivisione dei dati in un insieme di lavoro e uno di verifica.
- Metodi di regressione: regressione non parametrica, modelli additivi, alberi, mars, projection pursuit, reti neurali (richiami).
- Metodi di classificazione: mediante la regressione lineare, regressione logistica e multilogit, modelli additivi, alberi, polymars, reti neurali, combinazione di classificatori (bagging, boosting, foreste casuali).

- Metodi di analisi interna: nozioni sui metodi di raggruppamento, analisi delle associazioni tra variabili e algoritmo Apriori. Reti sociali (cenni).
Modalita' di esame :
Scritta/Pratica (con eventuale progetto)
Criteri di valutazione :
Le prove d'esame misureranno quanto ciascuno studente (a) saprÃ e quanto (b) saprÃ applicare degli strumenti proposti durante il corso.
Testi di riferimento :
Azzalini A., Scarpa B., Analisi dei dati e data mining. : Springer, 2004
Azzalini A., Scarpa B., Data analysis and data mining. : Oxford University Press, 2012
Eventuali indicazioni sui materiali di studio :
Libro di testo e materiale didattico fornito dal docente.

FONDAMENTI LOGICI DEI LINGUAGGI FUNZIONALI

(Titolare: Prof. SILVIO VALENTINI)

Periodo: *I anno, 1 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *40A+8E; 6,00 CFU*

Prerequisiti :
Conoscenze di base di logica matematica e del linguaggio insiemistico
Conoscenze e abilita' da acquisire :
Lo scopo di questo corso Ã¨ quello di fornire una introduzione teorica ai linguaggi di programmazione funzionali tipati e non tipati.
Attivita' di apprendimento previste e metodologie di insegnamento :
Lezioni frontali in aula
Contenuti :
Dopo aver richiamato la nozione di funzione calcolabile si introdurrÃ il lambda calcolo puro e si dimostrerÃ che esso Ã¨ uno strumento universale di calcolo. Si analizzerÃ quindi il lambda calcolo tipato semplice ed i suoi legami con il frammento implicativo del calcolo proposizionale intuizionista. Si introdurrano poi il calcolo con tipi dipendenti, che rappresenta il contenuto computazionale della logica del primo ordine, per continuare con calcoli con tipi di secondo ordine, potenti quanto l'aritmetica di Heyting al secondo ordine, e finire quindi con calcoli estremamente potenti che considerano insieme entrambi i sistemi di tipi ed eventualmente anche i tipi induttivamente generati, i tipi ricorsivi ed i tipi intersezione. Per tutti tali lambda calcoli si intendono dimostrare i principali teoremi matematici, vale a dire il teorema di normalizzazione e di confluenza, e fornire esempi di applicazione in informatica teorica.
Modalita' di esame :
L' accertamento di profitto avverrÃ con una prova orale dopo il completamento di esercitazioni personali da parte dello studente.
Criteri di valutazione :
L'esame intende valutare le conoscenze acquisite dallo studente sui temi del corso e le sue capacitÃ di svolgere del lavoro autonomo su di essi.
Testi di riferimento :
J.Y.Girard, Y.Lafont, P.Taylor, Proofs and Types. : Cambridge University Press,
H.Barendreght, The Lambda Calculus, its Syntax and Semantics. : North-Holland,
H.Barendreght,, Lambda Calculi with Types. : Oxford University Press,
Eventuali indicazioni sui materiali di studio :
Appunti forniti dal docente

INFORMATION RETRIEVAL

(Titolare: Prof. MASSIMO MELUCCI) - *Mutuato da: Laurea magistrale in Scienze Statistiche*

Periodo: *I anno, 2 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *48A; 6,00 CFU*

Prerequisiti :
Fondamenti di informatica, calcolo delle probabilitÃ e statistica.
Conoscenze e abilita' da acquisire :
L'insegnamento si occupa di Information Retrieval (IR) e dei metodi e modelli per i motori di ricerca, nonchÃ© di argomenti piÃ¹ avanzati come ad esempio Machine Learning e le sue applicazioni in IR. Le lezioni, i compiti assegnati e il laboratorio hanno lo scopo di dare gli strumenti metodologici per il progetto e la realizzazione di funzionalitÃ di information retrieval utili per applicazioni reali.
Attivita' di apprendimento previste e metodologie di insegnamento :
Lezione frontale ed attivitÃ di laboratorio.
Contenuti :
Gli argomenti principali necessari per la comprensione di un sistema di IR sono i seguenti:
Metodi di indicizzazione e reperimento
Modelli di reperimento
Motori di ricerca
Machine Learning e applicazioni in IR
Valutazione

Modalita' di esame :
Colloqui e presentazioni orali di progetti di gruppi di studenti.
Criteri di valutazione :
Si terrÃ conto di eventuali relazioni di progetto oltre alla conoscenza e competenza della materia.
Testi di riferimento :

M. Melucci, Information Retrieval: metodi e modelli per i motori di ricerca. : Franco Angeli, 2013
W.B. Croft, D. Metzler, T. Strohman, Search Engines: Information Retrieval in Practice. : Addison Wesley, 2009
C. Manning, P. Raghavan, H. Schutze, An introduction to information retrieval. : Cambridge University Press, 2008
R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval. : Addison Wesley, 2010
Eventuali indicazioni sui materiali di studio :
Si veda il libro di testo.

INTELLIGENZA ARTIFICIALE

(Titolare: Prof.ssa SILVANA BADALONI) - Mutuato da:

Periodo: *I anno, 1 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *60A; 8,00 CFU*

Prerequisiti :
Conoscenze di base di informatica.
Conoscenze e abilita' da acquisire :
Il corso ha come obiettivo l'acquisizione da parte degli studenti della conoscenza dei concetti di base, delle metodologie e delle tecniche applicative dellâ€™Intelligenza Artificiale.

Attivita' di apprendimento previste e metodologie di insegnamento :
Il corso ha una struttura modulare costituita da:
- lezioni in aula
- lezioni guidate in laboratorio informatico: esperienze nelle aule informatiche
- seminari invitati
- approfondimento di tematiche di ricerca da parte degli studenti nel lavoro delle tesine

Contenuti :
Introduzione all'Intelligenza Artificiale.
La nozione di Agente Intelligente.
Algoritmi per risolvere i problemi:
- Strategie di ricerca non informata: breadth-first search, depth-search, iterative deepening search
*- Ricerca informata: algoritmo greedy best-first search, algoritmo A**
Rappresentazione della conoscenza e ragionamento:
- Logica proposizionale
- Calcolo dei predicati
- Principio di risoluzione e introduzione alla programmazione logica
- Introduzione al Prolog
Problemi di soddisfacimento di vincoli:
- Rappresentazione di un problema come CSP
- Algoritmi di backtracking, forward checking, arc and path-consistency
Temporal Reasoning:
- Algebra degli intervalli e dei punti
Pianificazione:
- Ricerca nello spazio degli stati
- Partial-order planning POP
- Planning graphs
Ragionamento in presenza di incertezza:
- Teoria dei Fuzzy Sets, Logica Fuzzy e uso di vincoli fuzzy
- Ragionamento probabilistico e uso di tecniche bayesiane (cenni)
Apprendimento automatico con esperienze di laboratorio su reti neurali.
Algoritmi Meta-euristici.
Modalita' di esame :
L'esame consiste in una prova scritta (test a risposte multiple), nello sviluppo e nella presentazione del lavoro di una tesina, svolta in gruppo, come progetto di approfondimento di un argomento inerente al programma del corso, ed in un eventuale colloquio orale.
Criteri di valutazione :
Il voto finale Ã¨ una media ponderata dei punteggi conseguiti nella prova scritta (65%) e nella presentazione del lavoro di tesina (35%). Concorre alla valutazione la relazione riguardante le esperienze di laboratorio informatico. In caso di colloquio orale il voto puÃ² essere rimodulato.
Testi di riferimento :
S.Russell, P.Norvig, Intelligenza Artificiale. Un approccio moderno. Milano-Torino: Pearson Prentice Hall, 2005
S.Russell, P.Norvig, Intelligenza Artificiale. Un approccio moderno. Milano-Torino: Pearson Prentice Hall, 2010
Eventuali indicazioni sui materiali di studio :
Tutto il materiale didattico, tra cui le slides delle lezioni, gli articoli di rassegna e altra documentazione, viene pubblicato nel sito del Corso.

LINGUAGGI DI PROGRAMMAZIONE

(Titolare: Prof. GILBERTO FILE')

Periodo: *I anno, 1 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *54A+24E; 10,00 CFU*

Prerequisiti :
Conoscenze approfondite dei linguaggi Java e C++.
Conoscenze e abilita' da acquisire :
Conoscere un linguaggio funzionale (ML, Haskell). Apprezzare le differenze tra linguaggi funzionali ed imperativi. Apprezzare l'importanza dei tipi. Capire la gestione dei dati durante l'esecuzione di un programma (funzionale ed imperativo) e le sue implicazioni rispetto alla compilazione del linguaggio. Conoscere i temi principali che hanno segnato l'evoluzione dei linguaggi di programmazione dal 1950 a Java. La capacitÃ di costruire un compilatore ed un interprete.
Attivita' di apprendimento previste e metodologie di insegnamento :
Il corso consiste fondamentalmente di lezioni tradizionali in aula. Per l'apprendimento Ã¨ rilevante che ogni settimana una lezione di 2 ore sia organizzata come segue: nella prima ora gli studenti cercano di risolvere alcuni esercizi proposti dal docente sul materiale svolto nella settimana precedente. Nella seconda ora gli esercizi sono corretti alla lavagna con una forte interazione tra studenti e docente. Infine il progetto viene presentato agli studenti durante lo svolgimento del corso (in 5 parti) attraverso una documento ed alcune lezioni dedicate all'argomento. Inoltre il corso usa un sistema di elearning, basato sulla piattaforma Moodle, che consente un'interazione libera docente-studenti e anche studenti-studenti.
Contenuti :
I principali argomenti del corso sono i seguenti:
1) Un linguaggio funzionale (ML o Haskell): sintassi, esercizi, ricorsione, inferenza dei tipi, esecuzione eager e lazy;
2)Vari tipi di polimorfismo: parametrico, sovraccaricamento e di sottotipo;
3)Gestione run-time dei dati: blocchi, funzioni, ricorsione, scoping statico e dinamico, eccezioni;
4) Breve storia dei linguaggi orientati agli oggetti: Simula, Smalltalk, C++ e Java;
5) Pro e contro di C++;
6) Java a confronto con C++;
7)Il progetto consiste nella realizzazione di un compilatore per un semplice linguaggio funzionale: analisi lessicale, sintattica, generazione di codice intermedio, compilazione e interpretazione della traduzione finale.
Modalita' di esame :
L'esame consiste di una prova scritta ed una orale. Nella prova scritta ci sono domande pratiche e domande teoriche. L'orale Ã¨ una discussione sul progetto.
Criteri di valutazione :
La valutazione Ã¨ una misura dell'assimilazione del materiale del corso da parte dello studente. Gli esercizi scritti pratici mostrano la capacitÃ dello studente di applicare le nozioni apprese a problemi sempre diversi. Le domande teoriche mostrano la profonditÃ e l'ampiezza dell'apprendimento dello studente. Per ultimo, l'esame orale mostra la comprensione da parte dello studente del progetto che mette in gioco diversi concetti rilevanti insegnati nel corso.
Testi di riferimento :
John Mitchell, Concepts in Programming Languages. : Cambridge University Press, 2003
Eventuali indicazioni sui materiali di studio :
Le slide usate a lezione sono tutte a disposizione degli studenti sul sito elearning del corso. Alcuni articoli, menzionati durante il corso, vengono resi disponibili sull'elearning. Lo stesso vale per gli esercizi svolti ogni settimana nella lezione speciale descritta in precedenza e per il documento del progetto. Anche esami passati vengono resi disponibili sul sito di elearning.
Oltre a questo il corso segue un testo di riferimento.

LINGUAGGI DI PROGRAMMAZIONE AVANZATI

(Titolare: Dott.ssa SILVIA CRAFA)

Periodo: *I anno, 1 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *48A; 6,00 CFU*

Prerequisiti :
Conoscenze di programmazione e di programmazione ad oggetti.
Conoscenze e abilita' da acquisire :
Il corso presenta alcune tecniche avanzate dei moderni linguaggi di programmazione. Lo studente svilupperÃ la capacitÃ di comprendere, ragionare e valutare alcune delle nuove tecniche di programmazione.
Attivita' di apprendimento previste e metodologie di insegnamento :
Lezioni frontali con esercizi ed approfondimenti di argomenti di ricerca tramite articoli scientifici.
Contenuti :
Il corso presenta alcune tecniche avanzate dei moderni linguaggi di programmazione, tra cui: l'uso dei sistemi di tipi per ragionare sui programmmi, concetti avanzati di programmazione orientata agli oggetti (typing strutturale, type checking dinamico, mixins), linguaggi multi-paradigma, il design-by-contracts, programmazione concorrente basata sul modello ad attori. Tra i linguaggi su cui saranno affrontari questi argomenti ci sono Scala, C#, Spec#, Python, Ruby, Erlang, Go.
Modalita' di esame :
Sono previste una prova scritta e una seconda prova che consiste nella discussione orale di un tema di approfondimento o in alternativa nella realizzazione di un progetto software.
Criteri di valutazione :
La prova scritta valuta l'acquisizione dello studente degli aspetti fondazionali affrontati durante il corso. La seconda prova valuta la capacita' dello studente di analizzare e valutare aspetti avanzati dei linguaggi di programmazione.

Testi di riferimento :
B.C. Pierce, Types and Programming Languages. : The MIT Press, 2002
M. Odersky, L. Spoon, B. Venners, Programming in Scala. : Artima, 2008

LINGUAGGI E MODELLI PER IL GLOBAL COMPUTING

(Titolare: Prof. PAOLO BALDAN)

Periodo: *I anno, 2 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *48A; 6,00 CFU*

Prerequisiti :
Il corso richiede familiaritÃ con alcuni concetti matematici di base, quali relazioni, funzioni, insiemi, cardinalitÃ , ordini parziali, principi di induzione, sistemi di deduzione basati su regole di inferenza. Sono utili alcune conoscenze di semantica dei linguaggi di programmazione.
Il corso non ha propedeuticitÃ .
Conoscenze e abilita' 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), linguaggi di orchestrazione (ORC) e linguaggi per programmazione service oriented (Jolie).
Attivita' 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 mobilitaÌ€: 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. ProprietaÌ€ di safety e liveness.

- Verifica di proprietÃ con strumenti automatici. Il Concurrency Workbench ed il Mobility Workbench.

- Sistemi con topologia dinamica e mobilitÃ : pi-calcolo. Specifica di proprietÃ spaziali e cenni di applicazioni alla sicurezza dei protocolli.

- Dai linguaggi di specifica ai linguaggi di programmazione: linguaggi avanzati per la concorrenza (Google Go, Erlang), linguaggi di orchestrazione (ORC) e linguaggi per programmazione orientata ai servizi (Jolie).
Modalita' di esame :
Esercizi in classe, soluzione e discussione orale di esercizi avanzati, presentazione di un tema scelto dallo studente. Tra le opzioni ci saraÌ€ anche la realizzazione di un piccolo progetto che usi uno strumento di verifica.
Criteri di valutazione :
Lo studente Ã¨ valutato rispetto alla sua capacitÃ di risolvere semplici esercizi, verificando cosÃ¬ l'acquisizione di nozioni e tecniche discusse durante il corso. Alcuni esercizi avanzati sono finalizzati a verificare la capacitÃ di mettere a frutto quanto appreso per la soluzione di problemi nuovi. La presentazione verifica l'abilitÃ 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 :
R. Milner, Communication and Concurrency. : Prentice Hall, 1989
L. Aceto, A. Ingolfsdottir, K.G. Larsen, J. Srba, Reactive systems. : Cambridge University Press, 2007
Eventuali indicazioni sui materiali di studio :
Il libro di testo Ã¨ complementato con articoli di ricerca e altre risorse disponibili online.
Pagina web: http://www.math.unipd.it/~baldan/Global

LOGICA 2

(Titolare: Prof.ssa MARIA EMILIA MAIETTI) - Mutuato da: Laurea magistrale in Matematica (Ord. 2011)

Periodo: *I anno, 1 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *32A+16E; 6,00 CFU*

Prerequisiti :
E' caldamente suggerito, ma non strettamente necessario, aver seguito un corso di introduzione alla logica matematica.
Conoscenze e abilita' da acquisire :
Potenzialita' e limiti teorici del concetto di dimostrazione formale.
Differenze tra ragionamento classico e costruttivo.
Introduzione alla logica categoriale e alla matematica costruttiva e loro applicazioni computazionali.
Attivita' di apprendimento previste e metodologie di insegnamento :
Si intende sollecitare la partecipazione attiva di ogni studente, allo scopo di mettere in moto la sua visione critica, oltre che l'apprendimento nozionistico. Quindi le lezioni tradizionali saranno accompagnate da discussioni in aula, da esercizi da svolgere personalmente e da approfondimenti a scelta su temi concordati con il docente su articoli relativi ai temi del corso.
Contenuti :
Deduzione naturale per logica classica predicativa con variabili tipate sul lambda calcolo tipato semplice.
Deduzione naturale per logica intuizionista predicativa con variabili tipate sul lambda calcolo tipato semplice.
Aritmetica di Peano.
Aritmetica di Heyting.

Differenze tra aritmetica di Peano e di Heyting in termini di Tesi formale di Church e assioma di scelta.
Richiami dei teoremi di incompletezza di Goedel e confronti
tra prova per l'aritmetica classica e costruttiva.
Semantica della realizzabilita' (calcolabilita') per l'aritmetica di Heyting.
Introduzione alla teoria delle categorie: categoria, funtore, trasformazione naturale, aggiunzione, categorie indiciate.
Connettivi e quantificazioni logiche come aggiunzioni.
Dottrine categoriali elementari e iperdottrine di Lawvere.
Teorema di validita' e completezza con metodi categoriali.
Temi di approfondimento: topos elementari, topos effettivo (della calcolabilita'), Fondazione Minimalista per la matematica costruttiva.
Modalita' di esame :
A scelta tra una di queste tre opzioni:
1. orale su tutto il materiale del corso;
2. scritto su tutto il materiale del corso;
3. relazione orale su tema approfondito in accordo con il docente
e presentazione delle soluzioni di esercizi assegnati a lezione.
Criteri di valutazione :
Capacita' dello studente di utilizzare i concetti appresi durante il corso in modo personale. Capacita' di svolgere alcuni semplici esercizi, come applicazione dei concetti appresi e delle loro principali proprieta'.
Testi di riferimento :
A. S. Troelstra and D. van Dalen, Constructivism in Mathematics. An Introduction. : North-Holland, 1988
A. M. Pitts, Categorical Logic in Handbook of Logic in Computer Science, vol. 5. Algebraic and Logical Structures.. : Oxford University Press, 2000
S. Mac Lane, Categories for the Working Mathematician. : Springer, 1998
Eventuali indicazioni sui materiali di studio :
Dispense del docente, esercizi assegnati in aula e articoli
per approfondimenti proposti dal docente.

SICUREZZA

(Titolare: Prof. MAURO CONTI)

Periodo: *I anno, 2 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *40A; 6,00 CFU*

Prerequisiti :
Conoscenze di base di sistemi distribuiti, crittografia e sicurezza delle reti.
Conoscenze e abilita' da acquisire :
Acquisire conoscenze di sicurezza di sistema in ambiente Linux e Windows, sicurezza di rete wireless e wired, web-application security, sistema di gestione della sicurezza.
Al termine del corso gli studenti saranno in grado di: progettare lâ€™architettura di sistemi ed applicazioni sicure, e aggiornare autonomamente le proprie competenze nel settore.
Attivita' di apprendimento previste e metodologie di insegnamento :
Lezioni frontali, discussione di articoli scientifici.
Contenuti :
1) COMPUTER SECURITY TECHNOLOGY AND PRINCIPLES: Cryptographic Tools, User Authentication, Access Control, Database Security, Malicious Software, Denial-of-Service Attacks, Intrusion Detection, Firewalls and Intrusion Prevention Systems.
2) SOFTWARE SECURITY AND TRUSTED SYSTEMS: Buffer Overflow, Software Security, Operating System Security, Trusted Computing and Multilevel Security.
3) MANAGEMENT ISSUES: IT Security Management and Risk Assessment, IT Security Controls, Plans, and Procedures, Physical and Infrastructure Security, Human Resources Security, Security Auditing, Legal and Ethical Aspects.
4) PART FOUR CRYPTOGRAPHIC ALGORITHMS: Symmetric Encryption and Message Confidentiality, Public-Key Cryptography and Message Authentication.
5) NETWORK SECURITY: Internet Security Protocols and Standards, Internet Authentication Applications, Wireless Network Security.
Modalita' di esame :
Scritta.
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, 2011
M. Bishop, Introduction to Computer Security. : Addison-Wesley Professional, 2004
Eventuali indicazioni sui materiali di studio :
Libro (testo principale Computer Security: Principles and Practice 2/E) e articoli scientifici.

Il corso sarÃ tenuto in Inglese.
Il sito web del corso offrirÃ tutte le informazioni e materiale ulteriore:
http://www.math.unipd.it/~conti/teaching.html

SISTEMI CONCORRENTI E DISTRIBUITI

(Titolare: Prof. TULLIO VARDANEGA)

Periodo: *I anno, 1 semestre*
Indirizzo formativo: *Corsi comuni*

Tipologie didattiche: *52A+12E; 8,00 CFU*

Prerequisiti :
Lâ€™insegnamento assume familiaritÃ con lâ€™architettura degli elaboratori tradizionali, con la struttura e le attivitÃ dei loro sistemi operativi, particolarmente per quanto attiene a concorrenza, sincronizzazione e gestione dellâ€™I/O, e dei fondamenti delle reti. Lâ€™insegnamento non prevede propedeuticitÃ .
Conoscenze e abilita' da acquisire :
Il corso si propone di:
- illustrare problematiche e modelli di base e avanzati di concorrenza (intesa come parallelismo potenziale) realizzata a software, studiando le soluzioni proposte da Java e Ada, in quanto linguaggi riccamente dotati di supporto diretto alla concorrenza, come strumenti di sperimentazione e di confronto;
- 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 lascamente interconnessi.
Attivita' di apprendimento previste e metodologie di insegnamento :
Il corso si compone di due segmenti complementari. 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.
Nel secondo segmento si affronta invece l'evoluzione architetturale tecnologica dei sistemi distribuiti, culminando nell'analisi di CORBA come paradigma di interconnessione di sistemi eterogenei secondo il modello cliente-servente. In questa parte del corso si illustrano anche i fondamenti di approcci particolarmente avanzati come virtualizzazione e cloud computing. Nell'ambito di entrambi i segmenti 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.
Contenuti :
Problematiche di concorrenza
- Introduzione storica e metodologica
- Nozione di processo e modalitÃ 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
- Soluzioni concrete: Java RMI, Ada DSA, CORBA
- La frontiera del cloud computing
Modalita' di esame :
L'esame di profitto consiste nella redazione e nella discussione di una relazione scritta che illustri le problematiche affrontate nello svolgimento del progetto didattico assegnato dal docente, e le soluzioni adottate per risolverle. La presentazione della relazione viene accompagnata da una dimostrazione pratica del prodotto software realizzato in risposta ai requisiti del progetto.
Criteri di valutazione :
Lo sviluppo del progetto didattico 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 stesura della relazione tecnica mette alla prova la capacitÃ di sintesi e di astrazione dello studente. La presentazione e discussione del progetto di fronte al docente consente di completare la valutazione il grado di apprendimento complessivo dello studente rispetto ai principali temi della materia.
Testi di riferimento :
Alan Burns and Andy Wellings, Concurrent and Real-Time Programming in Ada. : Cambridge University Press, 2007
Andrew S Tanenbaum, Maarten van Steen, Distributed Systems - Principles and paradigms. : Pearson Education International, 2006
Eventuali indicazioni sui materiali di studio :
Il docente pubblica regolarmente tutte le diapositive utilizzate a lezione e anche materiale supplementare utile per l'approfondimento dei temi trattati in aula.

SISTEMI INFORMATIVI TERRITORIALI

(Titolare: da definire) - *Mutuato da:*

Periodo: *I anno, 2 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *48A; 6,00 CFU*

Prerequisiti :
nessuno
Conoscenze e abilita' da acquisire :
Gli studenti acquisiranno conoscenze, metodi e strumenti che consentono di trattare lâ€™informazione geografica e di progettare e realizzare sistemi per la sua gestione e fruizione.
Attivita' di apprendimento previste e metodologie di insegnamento :
Insegnamento frontale
esercitazioni pratiche
visita tecnica a societÃ specializzata in GIS/cartografia
Contenuti :
Concetti di base su informazione geografica e sistemi informativi geografici
La modellazione e la rappresentazione dellâ€™informazione geografica
Strutture dei dati spaziali
Architetture dei sistemi informativi geografici
Standard e norme applicabili
sviluppo di applicazioni gis desktop e web

Modalita' di esame :
test scritto
Criteri di valutazione :
valutazione di:
-corretta applicazione dei metodi per la modellazione e per la definizione dei componenti delle archietture applicative.
-capacitÃ di utilizzare gli standard e di rispettare le normative
Testi di riferimento :
CONTENUTO NON PRESENTE
Eventuali indicazioni sui materiali di studio :
VerrÃ reso disponibile materiale di supporto

SISTEMI MULTIMEDIALI

(Titolare: Dott.ssa OMBRETTA GAGGI)

Periodo: *I anno, 1 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *44A; 6,00 CFU*

Prerequisiti :
Sistemi Operativi, Reti e Sicurezza
Conoscenze e abilita' da acquisire :
Il corso illustra le principali tecnologie per la codifica, memorizzazione e diffusione di informazioni multimediali, e analizza le applicazioni distribuite con particolare riferimento all'ambiente Internet.
Attivita' di apprendimento previste e metodologie di insegnamento :
Lezioni frontali
Contenuti :
- 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 profonditÃ 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.

- La trasmissione dei dati continui. La suite di protocolli RTSP, RTCP e RTP.

- Concetto di qualitÃ di sevizio (QoS) nella trasmissione di dati Multimediali: il protocolli RSVP, IntServ e DiffServ.

- Architetture per la distribuzione di dati multimediali. Architetture client-server e P2P. Streaming e Jitter. Interazione tra flussi di rete elastici e real time, gestione del buffer.

- I sistemi operativi per media continui. Gestione delle risorse. QualitÃ di servizio. Scheduling real-time. Algoritmi di scheduling per media continui. Cenni alla programmazione su SmartPhone.
Modalita' di esame :
Esame orale o progetto
Criteri di valutazione :
L'esame verifica l'effettivo apprendimento dei concetti esposti
durante l'insegnamento. Questo puÃ² avvenire in forma di discussione orale, oppure applicando quanto appreso nella progettazione e realizzazione di una applicazione per smartphone
Testi di riferimento :
Ze-Nian Li, Mark S Drew, Fundamentals of Multimedia. : Prentice Hall, 2004
Eventuali indicazioni sui materiali di studio :
Le slide del corso sono fornite sul sito web del corso

SISTEMI REAL-TIME

(Titolare: Prof. TULLIO VARDANEGA)

Periodo: *I anno, 2 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *36A+12E; 6,00 CFU*

Prerequisiti :
L'insegnamento assume familiaritÃ con l'architettura degli elaboratori tradizionali, con la struttura e le attivitÃ dei loro sistemi operativi, particolarmente per quanto attiene a concorrenza, sincronizzazione e gestione dellâ€™I/O. Lâ€™insegnamento non prevede propedeuticitÃ .

Conoscenze e abilita' 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 piÃ¹ li differenziano dagli altri sistemi di calcolo. Attenzione sarÃ posta su alcuni paradigmi di progettazione e programmazione di tali sistemi, che ne facilitano l'analisi e la verifica.
Attivita' 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 piÃ¹ recenti sviluppi della teoria real-time intorno a tematiche di particolare interesse.
Contenuti :
- Introduzione: cenni storici e visione architetturale
- Cenni sulla affidabilitÃ 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
Modalita' di esame :
L'esame si svolge in una di due modalitÃ a scelta dello studente. Una modalitÃ 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 modalitÃ 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 modalitÃ , 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
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.

TECNOLOGIE OPEN-SOURCE

(Titolare: Dott. FRANCESCO TAPPARO)

Periodo: *I anno, 1 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *48A; 6,00 CFU*

Prerequisiti :
Nessuno
Conoscenze e abilita' da acquisire :
Conoscenza della storia del movimento open source e di tecnologie collaborative libere.
Attivita' di apprendimento previste e metodologie di insegnamento :
Lezioni frontali.
Contenuti :
Il corso si compone di due parti; nella prima si darÃ un'introduzione ai concetti ed alla storia del software libero ed open source, mentre nella seconda si introdurranno alcune tecnologie collaborative libere. I temi trattati saranno:
- la cultura hacker del MIT
- la nascita del progetto GNU
- il movimento open source
- Creative Common
- RDF e ccrel
- alcune tecnologie collaborative libere
Modalita' di esame :
Orale
Criteri di valutazione :
Conoscenza degli argomenti impartiti a lezione; dimistichezza teorica e pratica con le tecnologie insegnate.
Testi di riferimento :
CONTENUTO NON PRESENTE
Eventuali indicazioni sui materiali di studio :
Slide e materiale indicato nelle slide quando necessario.

TECNOLOGIE WEB 2

(Titolare: Prof. MASSIMO MARCHIORI)

Periodo: *I anno, 1 semestre*
Indirizzo formativo: *Corsi comuni*
Tipologie didattiche: *48A; 6,00 CFU*

Prerequisiti :
E' opportuno avere familiaritÃ con gli elementi di base del web, cosÃ¬ come forniti nel corso di "Tecnologie Web", in particolare HTML, CSS, XML, XSLT.

Conoscenze e abilita' da acquisire :
L'obiettivo principale del corso Ã¨ quello di dare una panoramica introduttiva di alcune tra le principali tecnologie web di livello avanzato, in modo da avere una visione ad alto livello del web attuale e del suo futuro.
Attivita' 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
UsabilitÃ ed interazione con gli utenti, analisi multi-livello, come costruire un sito web di successo.

+ E-commerce
Il caso studio dei siti di e-commerce, specializzazione dell'interazione col cliente.

+ Web Advertisement
La pubblicitÃ 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.

Modalita' di esame :
Lo studente deve superare uno scritto, e consegnare un progetto. Sopra una certa soglia minima di punteggio lo studente puÃ² opzionalmente richiedere un ulteriore esame orale.

Criteri di valutazione :
Il criterio di valutazione principale Ã¨ 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 Ã¨ fornito tramite il sito web del corso (http://corsi.math.unipd.it/tecweb2/), attraverso risorse online.