

FACOLTÀ	Ingegneria
ANNO ACCADEMICO	2012/2013
CORSO DI LAUREA MAGISTRALE	Laurea Magistrale in Ingegneria Informatica Classe LM-32 – Lauree Magistrali in Ingegneria Informatica
INSEGNAMENTO	Linguaggi e Traduttori
TIPO DI ATTIVITÀ	Caratterizzante
AMBITO DISCIPLINARE	Ingegneria Informatica
CODICE INSEGNAMENTO	04761
ARTICOLAZIONE IN MODULI	NO
NUMERO MODULI	--
SETTORI SCIENTIFICO DISCIPLINARI	ING-INF/05
DOCENTE RESPONSABILE	Marco Ortolani Ricercatore Università degli Studi di Palermo
CFU	6
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	66
NUMERO DI ORE RISERVATE ALLE ATTIVITÀ DIDATTICHE ASSISTITE	84
PROPEDEUTICITÀ	Nessuna
ANNO DI CORSO	Primo
SEDE DI SVOLGIMENTO DELLE LEZIONI	Consultare il sito www.ingegneria.unipa.it
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali, esercitazioni in aula
MODALITÀ DI FREQUENZA	Facoltativa
METODI DI VALUTAZIONE	Prova scritta e prova orale
TIPO DI VALUTAZIONE	Voto in trentesimi
PERIODO DELLE LEZIONI	Primo semestre
CALENDARIO DELLE ATTIVITÀ DIDATTICHE	Consultare il sito www.ingegneria.unipa.it
ORARIO DI RICEVIMENTO DEGLI STUDENTI	Giorni e orari di ricevimento

RISULTATI DI APPRENDIMENTO ATTESI

Conoscenza e capacità di comprensione (*knowledge and understanding*):

Lo studente, al termine del corso, avrà acquisito conoscenze e metodologie attinenti alle problematiche relative alle diverse fasi della compilazione con particolare attenzione all'analisi lessicale, sintattica e semantica ma che trovano applicazione anche in altri contesti (traduzioni di linguaggi, parser, scanner). Il corso si prefigge anche di trasmettere la conoscenza dei più importanti strumenti di generazione automatica di parser e scanner.

Conoscenza e capacità di comprensione applicate (*applying knowledge and understanding*):

Lo studente sarà in grado di comprendere il funzionamento degli analizzatori lessicali e sintattici, gli strumenti pratici per la realizzazione di tali analizzatori, il procedimento richiesto per trasformare gli analizzatori in traduttori, alcuni aspetti avanzati della compilazione di linguaggi moderni ed alcune tecniche di analisi automatica di correttezza di programmi. Il corso si propone anche di invitare gli studenti ad una conoscenza più approfondita dei linguaggi di programmazione a loro già noti, tramite lo studio di come tali linguaggi possono essere compilati.

Autonomia di giudizio (*making judgements*):

Lo studente sarà in grado di seguire i trend moderni nell'ambito della progettazione di compilatori e traduttori; sarà in grado di raccogliere i dati necessari alla valutazione delle prestazioni di un particolare compilatore, e di interpretare i risultati della valutazione; infine, sarà in grado di elaborare i requisiti necessari alla progettazione di un nuovo

compilatore, e di valutare l'efficacia di diverse soluzioni alternative.

Abilità comunicative (*communication skills*):

Lo studente acquisirà la capacità di comunicare ed esprimere problematiche inerenti l'oggetto del corso; sarà in grado di sostenere conversazioni su tematiche relative ai moderni linguaggi e tecniche di compilazione, di confrontare diversi metodi di compilazione, e di offrire possibili soluzioni.

Capacità di apprendere (*learning skills*):

Lo studente avrà appreso i nessi tra le tematiche dei compilatori, della progettazione dei linguaggi, e dell'importanza di adeguate tecniche di ottimizzazione, e questo gli consentirà di proseguire gli studi ingegneristici con un elevato grado di autonomia.

OBIETTIVI FORMATIVI

L'obiettivo del corso è quello di fornire gli strumenti fondamentali, sia formali sia pratici, per la definizione dei linguaggi di programmazione e dei loro compilatori. La prima parte del corso è dedicata alla teoria dei linguaggi formali ed alla sua relazione con la teoria degli automi. A tale scopo, si enfatizzano gli aspetti generativi e riconoscitivi dei linguaggi formali.

Nella seconda parte si illustrano gli aspetti più significativi dei linguaggi di programmazione, la loro evoluzione ed i concetti che stanno alla base della compilazione dei linguaggi di alto livello. Infine si presentano strumenti per la generazione automatica di analizzatori lessicali e sintattici, tanto bottom-up, come LEX e YACC, quanto top-down. Parte integrante del corso è la progettazione e la realizzazione di strumenti per l'analisi di programmi scritti in un linguaggio procedurale.

ORE FRONTALI	LEZIONI FRONTALI
8	Elaborazione dei Linguaggi, Struttura di un compilatore, Evoluzione dei Linguaggi di Programmazione, Applicazione della tecnologia dei compilatori, Fondamenti dei linguaggi di programmazione
8	Analisi Lessicale, Il ruolo di un analizzatore lessicale, bufferizzazione dell'input, specifica dei token, riconoscimento dei token, LEX: generatore di analizzatori lessicali, Automi a stati finiti, espressioni regolari ed automi
8	Analisi Sintattica, Grammatiche libere dal contesto, Scrittura di una grammatica, Analisi Top-Down, Analisi Bottom-Up, Analisi LR, Grammatiche ambigue, Generatori di Parser
8	Traduzione guidata dalla Sintassi, Definizioni guidate dalla sintassi (SDD), Ordine di valutazione delle SDD, Applicazioni di traduzioni guidate dalla sintassi e relativi schemi di traduzione, SDD ad L attributi
8	Generazione del Codice Intermedio, Varianti degli alberi sintattici, Codice a tre indirizzi, Tipi e Dichiarazioni, Traduzioni di Espressioni, Controllo dei Tipi, Flusso di Controllo, Correzione all'indietro, Comandi Switch.
8	Ottimizzazione del Codice. Principali ambiti per l'ottimizzazione, Introduzione all'analisi Data Flow, Propagazione delle costanti, Eliminazione delle ridondanze.
	ESERCITAZIONI
36	Realizzazione di un semplice traduttore guidato dalla sintassi
TESTI CONSIGLIATI	Aho, Lam, Sethi, Ullman, "Compilers: Principles, Techniques and Tools" Pearson International Edition