

FACOLTÀ	Scienze MM.FF.NN.
ANNO ACCADEMICO	2015/2016
CORSO DI LAUREA	Informatica
INSEGNAMENTO	Compilatori
TIPO DI ATTIVITÀ	Attività caratterizzanti
AMBITO DISCIPLINARE	Discipline Informatiche
CODICE INSEGNAMENTO	14049
ARTICOLAZIONE IN MODULI	NO
SETTORI SCIENTIFICO DISCIPLINARI	INF/01
DOCENTE RESPONSABILE	Marinella Sciortino Professore Associato Università di Palermo
CFU	6
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	102
NUMERO DI ORE RISERVATE ALLE ATTIVITÀ DIDATTICHE ASSISTITE	48
PROPEDEUTICITÀ	Tutte le materie del I anno, Algoritmi e Strutture Dati, Informatica Teorica, Sistemi Operativi
ANNO DI CORSO	Terzo
SEDE DI SVOLGIMENTO DELLE LEZIONI	Dipartimento di Matematica e Informatica di Palermo
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali, Attività in laboratorio
MODALITÀ DI FREQUENZA	Facoltativa
METODI DI VALUTAZIONE	Prova orale con discussione di un elaborato
TIPO DI VALUTAZIONE	Voto in trentesimi
PERIODO DELLE LEZIONI	Primo semestre
CALENDARIO DELLE ATTIVITÀ DIDATTICHE	Come da calendario disponibile presso www.cs.unipa.it
ORARIO DI RICEVIMENTO DEGLI STUDENTI	Da programmare

RISULTATI DI APPRENDIMENTO ATTESI

Conoscenza e capacità di comprensione

Il corso intende fornire agli studenti le nozioni necessarie per comprendere ed affrontare le diverse problematiche relative alle 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 di importanti strumenti di generazione automatica di parser e scanner (BISON e FLEX)

Capacità di applicare conoscenza e comprensione

Il corso si pone come obiettivo principale la comprensione, da parte dello studente, dei modelli computazionali e delle tecniche che gestiscono il funzionamento degli analizzatori lessicali e sintattici in un moderno compilatore, al fine di applicare tali metodologie da un lato per la generazione automatica o manuale di tali analizzatori, e dall'altro per la trasformazione e l'analisi dei testi guidata dalla sintassi.

Autonomia di giudizio

Gli studenti saranno guidati ad apprendere in maniera critica e responsabile gli argomenti che

verranno loro proposti in aula e in laboratorio. Ciascuno studente avrà inoltre occasione di arricchire la propria autonomia di giudizio attraverso la realizzazione di un progetto o di un elaborato che sarà parte integrante della prova di valutazione.

Abilità comunicative

Attraverso le attività di laboratorio previste, il corso tenderà a sviluppare negli studenti l'interazione e la capacità di saper lavorare in gruppo, di confrontarsi sulle problematiche al fine di individuare le soluzioni in base alle conoscenze acquisite durante il corso. L'acquisizione delle abilità comunicative sarà realizzata tramite la partecipazione attiva dello studente alle attività di laboratorio nonché l'esposizione dei risultati del lavoro individuale o di gruppo su argomenti o problematiche proposti dal docente.

Capacità d'apprendimento

Il materiale proposto in aula e le esercitazioni in laboratorio svilupperanno le capacità di apprendimento degli studenti che saranno in grado di "interrogare" in modo integrato le proprie conoscenze-competenze a fronte delle problematiche affrontate. Gli studenti saranno stimolati inoltre ad una conoscenza più approfondita e critica dei linguaggi di programmazione a loro già noti, tramite lo studio di come avviene il processo di compilazione di tali linguaggi.

OBIETTIVI FORMATIVI DEL CORSO

L'insegnamento ha due obiettivi. Il primo è di tipo culturale. Si vuole cioè introdurre una teoria, quella alla base della realizzazione dei compilatori per i linguaggi di programmazione, che rappresenta probabilmente il più importante contributo scientifico allo sviluppo e alla diffusione delle tecnologie informatiche. Il secondo obiettivo è più applicativo. Infatti, anche qualora lo studente non venga mai coinvolto (nella propria carriera lavorativa) nel progetto di un nuovo compilatore, è comunque altamente probabile che egli/ella si trovi spesso ad utilizzare metodologie e tecniche che di tale teoria fanno parte: dagli automi e le grammatiche come formalismi per definire il comportamento di un sistema, alle tecniche per realizzare traduttori eventualmente più semplici di un compilatore vero e proprio (ad esempio, l'interprete di un file di configurazione). Verrà studiata la struttura generale di un compilatore, ponendo poi particolare enfasi agli aspetti di analisi lessicale e di parsing. Verranno studiati parser di tipo top-down (a discesa ricorsiva, predittivi e LL(1)) e di tipo bottom-up (LR e LALR). Infine, saranno esaminati esempi di traduttori per semplici linguaggi.

ORE FRONTALI	LEZIONI FRONTALI
4	Introduzione: Linguaggi di Programmazione e Processori di Linguaggi di Programmazione. Compilatori e Linguaggi. Linguaggi Macchina, Linguaggi Assembly ed evoluzione dei linguaggi di programmazione. Compilatori e Interpreti. Macchina Virtuale. Struttura di un compilatore: Preprocessore. Linker e Loader. Fasi della compilazione. Front End e Back End. Passate di un compilatore.
6	Analisi lessicale: Operazioni preliminari. Token e Lessemi. Errori lessicali. Token e espressioni regolari. Definizioni regolari. Eliminazioni di ambiguità. Automi a stati finiti. Implementazioni di DFA. Simulazioni di NFA. Generazione automatica di scanner. Uso di Flex.
14	Analisi sintattica: Grammatiche context-free. Alberi di derivazione. Grammatiche ambigue. Automi a pila deterministici e non deterministici. Complessità di calcolo di un automa a pila. Algoritmo di Earley. Errori sintattici e metodi di gestione degli errori. Parser discendenti o top-down: Parser a discesa ricorsiva. Parser LL(1). Eliminazione della ricorsione sinistra. Fattorizzazione sinistra. Insiemi First e Follow.

	Parser ascendenti o bottom-up: Parser shift-reduce. Parser LR(0). Parser SLR. Parser LR(1). Parser LALR(1). Proprietà dei linguaggi e delle grammatiche LR(k). Confronto tra le grammatiche LL(k) e LR(k). Generatori automatici di parser. Uso di Bison.
6	Analisi semantica: Semantica statica e dinamica. Grammatiche con attributi. Semantica guidata dalla sintassi. Albero sintattico decorato. Calcolo degli attributi. Grafo delle dipendenze. Grammatiche con S- attributi. Grammatiche con L-attributi. Ordinamento topologico del grafo delle dipendenze. Tabella dei simboli. Vari tipi di implementazioni tramite array, liste concatenate e ABR. Implementazione tramite hash table with chaining. Attributi di visibilità e metodi di realizzazione. Type checking. Equivalenza di tipi. Type coercion.
2	Generazione del codice: Codice intermedio. Codice a tre indirizzi. Strutture dati per l'implementazione del 3AC. Codice per macchina virtuale. P-code. Ottimizzazione del codice. Esempi di ottimizzazioni indipendenti dalla macchina. Generatori di codice oggetto. Esempi di ottimizzazioni dipendenti dalla macchina.
ATTIVITA' in LABORATORIO	
6	Esercizi risolti con l'ausilio di Flex
10	Esercizi risolti con l'ausilio di Bison. Uso della tabella dei simboli
TESTI CONSIGLIATI	A. Aho, M. Lam, R. Sethi, J. Ullman Compilers, Principles, Techniques & Tools Addison Wesley A. Aho, M. Lam, R. Sethi, J. Ullman COMPILATORI. Principi, Tecniche e strumenti Addison Wesley Giorgio Bruno Linguaggi Formali e Compilatori Utet Libreria Stefano Crespi Reghizzi Linguaggi Formali e Compilazione Pitagora Editrice