

<b>FACOLTÀ</b>	Ingegneria
<b>ANNO ACCADEMICO</b>	2013/2014
<b>CORSO DI LAUREA MAGISTRALE</b>	Ingegneria Informatica
<b>INSEGNAMENTO</b>	Linguaggi e Traduttori
<b>TIPO DI ATTIVITÀ</b>	Caratterizzante
<b>AMBITO DISCIPLINARE</b>	Ingegneria Informatica
<b>CODICE INSEGNAMENTO</b>	04761
<b>ARTICOLAZIONE IN MODULI</b>	NO
<b>NUMERO MODULI</b>	--
<b>SETTORI SCIENTIFICO DISCIPLINARI</b>	ING-INF/05
<b>DOCENTE RESPONSABILE</b>	Marco Ortolani Ricercatore Università degli Studi di Palermo <a href="mailto:marco.ortolani@unipa.it">marco.ortolani@unipa.it</a>
<b>CFU</b>	6
<b>NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE</b>	96
<b>NUMERO DI ORE RISERVATE ALLE ATTIVITÀ DIDATTICHE ASSISTITE</b>	54
<b>PROPEDEUTICITÀ</b>	Nessuna
<b>ANNO DI CORSO</b>	Primo
<b>SEDE DI SVOLGIMENTO DELLE LEZIONI</b>	Consultare il sito <a href="http://www.ingegneria.unipa.it">www.ingegneria.unipa.it</a>
<b>ORGANIZZAZIONE DELLA DIDATTICA</b>	Lezioni frontali; Analisi e discussione in aula di casi di studio; Esercitazioni teoriche e pratiche su costrutti tipici dei linguaggi; Presentazioni e discussioni in aula di progetti e implementazioni.
<b>MODALITÀ DI FREQUENZA</b>	Facoltativa
<b>METODI DI VALUTAZIONE</b>	Discussione sugli argomenti del corso; Elaborato su prova pratica di implementazione di costrutti di un linguaggio assunto come caso di studio; Discussione degli elaborati svolti durante le esercitazioni teoriche; Eventuale presentazione e discussione di una tesina concordata su un tema di ricerca.
<b>TIPO DI VALUTAZIONE</b>	Voto in trentesimi
<b>PERIODO DELLE LEZIONI</b>	Consultare il sito <a href="http://www.ingegneria.unipa.it">www.ingegneria.unipa.it</a>
<b>CALENDARIO DELLE ATTIVITÀ DIDATTICHE</b>	Consultare il sito <a href="http://www.ingegneria.unipa.it">www.ingegneria.unipa.it</a>
<b>ORARIO DI RICEVIMENTO DEGLI STUDENTI</b>	Lunedì 15.00-17.00 o su appuntamento.

<b>RISULTATI DI APPRENDIMENTO ATTESI</b>
<b>Conoscenza e capacità di comprensione (knowledge and understanding):</b> 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 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 all'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.

<b>ORE</b>	<b>LEZIONI FRONTALI</b>
6	Evoluzione dei linguaggi di programmazione. Struttura di un compilatore. Applicazione della tecnologia dei compilatori.
6	Il ruolo di un analizzatore lessicale, automi a stati finiti, espressioni regolari
6	Analisi Sintattica. Grammatiche libere dal contesto. Scrittura di una grammatica. Analisi top-down e bottom-up. Analizzatori LL e LR. Grammatiche ambigue.
6	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.

6	Generazione del Codice Intermedio. Codice a tre indirizzi. Casi di studio: tipi e dichiarazioni, traduzioni di Espressioni, controllo dei tipi, controllo di flusso, correzione all'indietro.
6	Ottimizzazione del Codice. Principali ambiti per l'ottimizzazione, introduzione all'analisi data flow, propagazione delle costanti, eliminazione delle ridondanze.
<b>36</b>	
	<b>ESERCITAZIONI TEORICO-PRATICHE</b>
4	Introduzione alle espressioni regolari, con esempi in contesti di uso comune (egrep, bash). Uso del generatore di analizzatori lessicali lex.
4	Esercizi sulla costruzione di tabelle di parsing top-down e bottom-up; dimostrazioni per induzione sulle grammatiche. Yacc
4	Esercizi sulla costruzione di parser. Utilizzo del generatore automatico di parser yacc/bison.
4	Casi di studio sulla traduzione di istruzioni di controllo di flusso e di dichiarazione di tipi composti.
4	Esempio completo di costruzione di un compilatore per un linguaggio ad uso didattico.
<b>20</b>	
<b>TESTI CONSIGLIATI</b>	Aho, Lam, Sethi, Ullman, "Compilers: Principles, Techniques and Tools", 2nd edition, Addison Wesley, 2006. D. Grune, H. Bal, C. Jacobs, K. Langendoen. Modern Compiler Design. Wiley, 2000.