UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT	Ingegneria
ACADEMIC YEAR	2023/2024
MASTER'S DEGREE (MSC)	COMPUTER ENGINEERING
SUBJECT	FORMAL LANGUAGES AND COMPILERS
TYPE OF EDUCATIONAL ACTIVITY	B
AMBIT	50369-Ingegneria informatica
CODE	04761
SCIENTIFIC SECTOR(S)	ING-INF/05
HEAD PROFESSOR(S)	LO RE GIUSEPPE Professore Ordinario Univ. di PALERMO
OTHER PROFESSOR(S)	
CREDITS	9
INDIVIDUAL STUDY (Hrs)	144
COURSE ACTIVITY (Hrs)	81
PROPAEDEUTICAL SUBJECTS	
MUTUALIZATION	
YEAR	1
TERM (SEMESTER)	2° semester
ATTENDANCE	Not mandatory
EVALUATION	Out of 30
TEACHER OFFICE HOURS	**LO RE GIUSEPPE** Tuesday 15:00 17:00

DOCENTE: Prof. GIUSEPPE LO RE

PREREQUISITES	Buona conoscenza dei costrutti tipici dei linguaggi di programmazione procedurali e ad oggetti; conoscenza dell'organizzazione della memoria di un calcolatore e del funzionamento del sistema operativo.
LEARNING OUTCOMES	Conoscenza e capacita' di comprensione (knowledge and understanding): Lo studente, al termine del corso, avra' 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 piu' importanti strumenti di generazione automatica di parser e scanner. Conoscenza e capacita' di comprensione applicate (applying knowledge and understanding): Lo studente sara' 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 piu' approfondita dei linguaggi di programmazione a loro gia' noti, tramite lo studio di come tali linguaggi possono essere compilati. Autonomia di giudizio (making judgements): Lo studente sara' in grado di seguire i trend moderni nell'ambito della progettazione di compilatori e traduttori; sara' in grado di raccogliere i dati necessari alla valutazione delle prestazioni di un particolare compilatore, e di interpretare i risultati della valutazione; infine, sara' in grado di elaborare i requisiti necessari alla progettazione di un nuovo compilatore, e di valutare l'efficacia di diverse soluzioni alternative. Abilita' comunicative (communication skills): Lo studente acquisira' la capacita' di comunicare ed esprimere problematiche inerenti all'oggetto del corso; sara' 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. Capacita' di apprendere (learning skills): Lo studente avra' appreso i nessi tra le tematiche dei compilatori, della progettazione dei linguaggi, e dell'importanza di adeguate tecniche di ottimizzazione, e questo gli consentira' di proseguire gli studi ingegneristici con un elevato grado di autonomia.
ASSESSMENT METHODS	La valutazione avverra' mediante somministrazione di una prova scritta, al superamento della quale si potra' accedere all'esame orale. La prova scritta vertera' sulla costruzione di parser tabellari e sulla progettazione di costrutti di un linguaggio assunto come caso di studio. L'esame orale consistera' in una discussione dello svolgimento della prova scritta e nella discussione sugli argomenti del corso e degli elaborati svolti durante le esercitazioni teoriche. Valutazione della prova scritta Durante il corso, in accordo con il calendario accademico, sara' possibile sostenere una o più prove in itinere. Tali prove, a discrezione dell'allievo potra' essere completata con una prova finale da sostenere nel periodo compreso tra la fine delle lezioni ed il primo appello del corso. La media pesata delle prove in itinere e di quella finale costituisce il voto della prova scritta. La prova scritta e' costituita da esercizi e quesiti a risposta aperta volti a verificare le conoscenze dello studente degli argomenti affrontati durante il corso, e di applicare le capacita' e le conoscenze acquisite. Nello svolgimento assume fondamentale importanza il commento teorico dei risultati ottenuti. L'articolazione della soluzione consente di apprezzare tutti i livelli di preparazione. La valutazione e' espressa in trentesimi ed un minimo di 15 e' richiesto per accedere alla prova orale. Valutazione per la prova orale La prova orale consiste in un colloquio, volto ad accertare il possesso delle competenze e delle conoscenze disciplinari previste dal corso; la valutazione viene espressa in trentesimi. Durante il colloquio orale lo studente dovra' essere in grado di discutere le soluzioni proposte durante la prova scritta; inoltre saranno proposte domande di diverso e crescente livello di complessita' al fine di valutare il raggiungimento degli obiettivi formativi e le abilita' comunicative dello studente. Infine, allo scopo di valutare l'autonomia di giudizio, sara' richiesto di analizzare le caratteristiche di specifici scenari applicativi e di proporre le soluzioni piu' adeguate ai problemi individuati. La valutazione finale terra' conto sia del punteggio della prova scritta (50%) che di quello delle prova orale (50%). Eccellente 30-30 e lode. Durante entrambe le prove lo studente dovra' dimostrare padronanza completa degli argomenti del corso. Durante il colloquio orale l'allievo dovra' dimostrare la maturita' di saper collegare i diversi aspetti trattati e la capacita' di saper generalizzare. Dovra' mostrare autonomia nella

	soluzione dei quesiti e la capacita' di individuare le informazioni necessarie per la soluzione degli stessi. Molto buono 27-29 Buona padronanza degli argomenti, lo studente e' in grado di applicare le conoscenze per risolvere i problemi proposti. Buono 24-26 buona conoscenza dei principali, discreta padronanza e proprieta' di linguaggio, con capacita' di applicare autonomamente le conoscenze alla soluzione dei problemi proposti. Discreto 21-23 Piu' che sufficiente padronanza degli argomenti principali dell'insegnamento, limitata capacita' di applicare autonomamente le conoscenze acquisite. Sufficiente 18-20 conoscenza di base degli argomenti principali dell'insegnamento e del linguaggio tecnico.
EDUCATIONAL OBJECTIVES	The goal of the course is to provide the fundamental cognitive tools, both from the theoretical and the practical standpoint, for the definition of programming languages and the relative compilers. The first part of the course is devoted to the theory of formal languages, with specific regard to automata theory. To this aim, formal languages are regarded both as string generators and as recognizers. In the second part, the main aspects of programming languages will be discussed, as well as their evolution and the grounding concepts behind compilation of high-leve constructs. Finally, the tools for automatic generation of lexical and syntax analyzers will be presented, both for bottom-up parsing (i.e. Lex/Flex and YACC/Bison), and for top-down parsing. A fundamental aim of the course is the design and implementation of tools for the analysis of procedural languages.
TEACHING METHODS	Lectures, with analysis and discussions of case studies. Theoretical and practical exercises about typical language constructs. Presentation and discussion of projects and implementations.
SUGGESTED BIBLIOGRAPHY	Aho, Lam, Sethi, Ullman, "Compilers: Principles, Techniques and Tools", 2nd edition, Addison Wesley, 2006. ISBN 0-321-49169-6

SYLLABUS

Hrs	Frontal teaching
3	Introduction. Structure of a compiler: analysis and synthesis. Evolution of programming languages and science of compiler design; application of compiler technology. Fundamental elements of a programming language.
8	Lexical analysis.Comparison between lexical analysis and parsing. Tokens, patterns, and lexemes. Handling typical lexical errors. Token specifications. Formal definition of regular expressions. Main properties of reg-exps.
8	Finite state automata; deterministic (DFA) and non-deterministic automata (NFA). Properties of DFAs and NFAs and their tabular representations. Conversion between different representations through NFA, DFA, reg-exps. Optimization of DFA-based lexers.
4	Closure properties of regular languages. Pumping lemma for regular languages.
6	Syntax analysis: the role of the parser. Context-free grammars. Syntax error handling and recovery. Derivations and parsing trees. Handling ambiguity; checking the language generated by a grammar; comparing CFGs
6	Parsing top down e grammatiche LL(1). Parser ricorsivo discendente. Parser predittivo non ricorsivo. Analisi Sintattica Bottom Up. Grammatiche LR(1). Parsing Shift-Reduce. Item LR(0). Chiusura e funzione GOTO. Parser bottom-up SLR. Parser LR canonico. Parser LALR. Efficienza delle implementazioni dei parser bottom-up.
6	Syntax-directed translation. SDDs. Use of synthesized and inherited attributes. SDTs. Implementation of SDTs for different kinds of parsers. On-the-fly code generations. Issues with bottom-up parsing and L-attributes SDDs.
6	Intermediate code generation. Representations of 3-address code. Quadruples and triples.
6	Types and declarations. Translation of primitive types and type checking. Translation of flow-control statements. Translation of declaration and use of composite types (arrays and records). One-pass compilation and backpatching technique.

Hrs	Practice
6	The flex lexer generator. Implementation of reg-exps with flex. Use in bash, egrep.
4	Induction proofs for proving correctness or checking ambiguity. Operations on grammars. Disambiguation, left-factoring, left recursion elimination.
6	Exercises on top-down parser tables; proofs by induction on grammars. Exercises on bottom-up parser tables. The automatic parser generator yacc/bison.

Hrs	Workshops
12	Design and implementation of tools for the analysis of procedural languages.