Syllabus

Descrizione corso

Titolo insegnamento	Formal Languages and Compilers
Codice insegnamento	76214
Titolo aggiuntivo	
Settore Scientifico-Disciplinare	INFO-01/A
Lingua	Italiano
Corso di Studio	Corso di laurea in Informatica
Altri Corsi di Studio (mutuati)	
Docenti	prof. Alessandro Artale, Alessandro.Artale@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/3026
Assistente	
Semestre	Secondo semestre
Anno/i di corso	2
CFU	6
Ore didattica frontale	40
Ore di laboratorio	20
Ore di studio individuale	90
Ore di ricevimento previste	
Sintesi contenuti	L'obiettivo principale è quello di introdurre le nozioni fondamentali sui linguaggi formali (classificazione di Chomsky delle lingue, lingue regolari, automi, grammatiche libere dal contesto) e di comprendere i meccanismi che regolano l'analisi e la sintesi dei linguaggi di programmazione. Gli studenti apprenderanno le tecniche più importanti per la rappresentazione e la generazione di linguaggi (in particolare, linguaggi regolari e context-free). Queste tecniche saranno applicate alla costruzione di un compilatore per un linguaggio di programmazione. Durante il corso lo studente imparerà a costruire le diverse parti di un compilatore, con

	particolare attenzione agli analizzatori lessicali (con l'uso di Lex durante il laboratorio), ai parser (con l'uso di YACC durante il laboratorio) e alle basi della generazione del codice.
Argomenti dell'insegnamento	- Teoria dei linguaggi formali - Linguaggi regolari: automi, espressioni regolari, grammatiche regolari - Linguaggi liberi dal contesto (macchine a pila) - Analisi lessicale e sintattica: Specificazione del lexer, parsing top-down e bottom-up - Regole di analisi semantica per: controllo dei tipi, tabella dei simboli e flusso di controllo - Generazione di codice intermedio
Parole chiave	Teoria Formale dei Linguaggi Automi a Stati Finiti Grammatiche Context-Free e Parser Analisi Semantica e Generazione del codice
Prerequisiti	Conoscenza del linguaggio di programmazione C e capacità di gestire strutture dinamiche. Esperienza di lavoro in ambiente Linux (preferibile)
Insegnamenti propedeutici	
Modalità di insegnamento	Il corso prevede lezioni frontali, sessioni di laboratorio con esercizi di programmazione e progetti di gruppo.
Obbligo di frequenza	La frequenza non è obbligatoria ma consigliata. Gli studenti non frequentanti devono contattare il docente all'inizio del corso per concordare le modalità dello studio indipendente. Le modalità d'esame per gli studenti non frequentanti sono le stesse degli studenti frequentanti.
Obiettivi formativi specifici e risultati di apprendimento attesi	Conoscenza e comprensione - D1.7 Possedere una solida conoscenza dei fondamenti teorici dell'informatica. - D1.10 Conoscere i concetti di linguaggi formali e le tecniche di compilazione di vari linguaggi di programmazione di alto livello. Applicare conoscenza e comprensione - D2.8 Essere in grado di sviluppare e costruire traduttori e compilatori. Capacità di formulare giudizi

	- D3.1 Essere in grado di raccogliere e interpretare dati utili e di giudicare i sistemi informativi e la loro applicabilità. - D3.2 Essere in grado di lavorare autonomamente in base al proprio livello di conoscenza e comprensione. Abilità comunicative - D4.1 Essere in grado di utilizzare una delle tre lingue, inglese, italiano e tedesco, e di utilizzare in modo appropriato termini tecnici e di comunicazione. - D4.5 Essere in grado di lavorare in team per la realizzazione di sistemi informatici. Capacità di apprendimento - D5.1 Avere sviluppato capacità di apprendimento per proseguire gli studi con un elevato grado di autonomia.
Obiettivi formativi specifici e risultati di apprendimento attesi (ulteriori info.)	
Modalità di esame	La valutazione consiste in un progetto di gruppo e in un esame scritto. Il progetto è volto a valutare l'applicazione delle conoscenze acquisite, la capacità di formulare giudizi informati e le abilità comunicative, attraverso lo sviluppo collaborativo di un compilatore per un piccolo linguaggio di programmazione. Il completamento con successo del progetto è un prerequisito per l'ammissione all'esame scritto. L'esame scritto è composto da due parti: la prima si basa su argomenti di linguaggio formale e la seconda su argomenti di regole lessicali, parser e semantiche. La prima parte sarà offerta anche come esame di metà corso. L'esame scritto comprende domande di verifica, compiti di trasferimento delle conoscenze ed esercizi pratici e mira a valutare la conoscenza e la comprensione, la capacità di applicare le conoscenze e le capacità di apprendimento dello studente.
Criteri di valutazione	Il voto finale è composto da un esame scritto del 70% (35% sugli argomenti di linguaggio formale e 35% sugli argomenti di regole

	lessicali, parser e semantiche) e da un progetto sullo sviluppo di compilatori del 30%. L'esame scritto sarà valutato in base alla correttezza e alla chiarezza delle risposte. Il progetto sarà valutato in base alla qualità della soluzione, compresa la complessità e l'originalità del linguaggio di programmazione progettato, le strutture dati utilizzate per implementare la tabella dei simboli e la profondità dell'analisi semantica effettuata. La valutazione del progetto sarà valida per tre sessioni d'esame regolari consecutive.
Bibliografia obbligatoria	• Alfred V. Aho, Monica S. Lam, Ravi Sethi e Jeffrey D. Ullman. Compilatori: Principles, Techniques, and Tools. Addison-Wesley Longman Publishing Co., Inc., USA, 2a edizione, 2006. ISBN 0321486811. • John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. Introduzione alla teoria degli automi, ai linguaggi e alla computazione. Pearson/Addison Wesley, Boston, terza edizione, 2007. ISBN 0321455363 9780321455369 0321462254 9780321462251 0321455371 9780321455376 0321476174 9780321476173. URL: http://infolab.stanford.edu/~ullman/ialc.html.
Bibliografia facoltativa	• Kenneth C. Louden. Costruzione di compilatori: Principles and Practice. PWS Publishing Co., USA, 1997. ISBN 0534939724. • Steven S. Muchnick. Progettazione e implementazione di compilatori avanzati. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998. ISBN 1558603204. • David Watt, Deryck Brown e Robert W. Sebesta. Processori di linguaggio di programmazione in Java: Compilatori e Interpreti E Concetti di Linguaggi di Programmazione. Prentice Hall Press, USA, 2007. ISBN 1408200414.
Altre informazioni	- C (https://gcc.gnu.org) - YACC (https://pubs.opengroup.org/onlinepubs/9799919799/utilities/yacc.html) - LEX (https://pubs.opengroup.org/onlinepubs/9799919799/utilities/lex.html) - Linux (ad esempio, https://ubuntu.com)
Obiettivi di Sviluppo	Istruzione di qualità

Sostenibile (SDGs)	