

<b>Principali informazioni sull'insegnamento</b>	
Titolo insegnamento	Linguaggi di Programmazione
Corso di studio	Informatica
Crediti formativi	9
Denominazione inglese	Programming Languages
Obbligo di frequenza	No, ma la frequenza è fortemente consigliata
Lingua di erogazione	Italiano

<b>Docente responsabile</b>	Nome Cognome	Indirizzo Mail
	Pasquale Lops	pasquale.lops@uniba.it
Luogo ed Orario di Ricevimento	Ufficio 760 Dip. Informatica, Piano VII	Martedì: 10:00-12:00

<b>Dettaglio credi formativi</b>	Ambito disciplinare	SSD	Crediti
	Informatico	INF/01	9

<b>Modalità di erogazione</b>	
Periodo di erogazione	Secondo semestre
Anno di corso	Primo
Modalità di erogazione	Lezioni frontali Esercitazioni in aula

<b>Organizzazione della didattica</b>	
Ore totali	225
Ore di corso	86
Ore di studio individuale	139

<b>Calendario</b>	
Inizio attività didattiche	24/02/2020
Fine attività didattiche	29/05/2020

<b>Syllabus</b>	
Prerequisiti	Propedeuticità formali: nessuna Propedeuticità culturali: programmazione (indispensabile), matematica discreta (importante)
Risultati di apprendimento previsti (declinare rispetto ai Descrittori di Dublino) (si raccomanda che siano coerenti con i risultati di apprendimento del CdS, compreso i risultati di apprendimento trasversali)	<ul style="list-style-type: none"> <li>• <i>Conoscenza e capacità di comprensione</i> Lo studente sarà in grado di comprendere: <ul style="list-style-type: none"> <li>- l'associazione tra problemi, linguaggi formali e linguaggi di programmazione;</li> <li>- la relazione tra linguaggi formali e linguaggi di programmazione;</li> <li>- le tecniche di analisi e traduzione dei linguaggi di programmazione;</li> <li>- i fondamenti teorici alla base delle componenti di analisi</li> </ul> </li> </ul>

	<p>e traduzione dei linguaggi di programmazione (scanner, parser).</p> <ul style="list-style-type: none"> <li>• <i>Conoscenza e capacità di comprensione applicate</i> Lo studente sarà in grado di: <ul style="list-style-type: none"> <li>- Classificare un linguaggio formale nella gerarchia di Chomsky;</li> <li>- Generare, descrivere e riconoscere un linguaggio formale.</li> </ul> </li> <li>• <i>Autonomia di giudizio</i> Lo studente sarà in grado di: <ul style="list-style-type: none"> <li>- valutare la correttezza della sintassi di un linguaggio di programmazione;</li> <li>- comprendere la natura dei messaggi di errore prodotti dalle componenti di analisi e traduzione dei linguaggi di programmazione (scanner, parser).</li> </ul> </li> <li>• <i>Abilità comunicative</i> Lo studente sarà in grado di: <ul style="list-style-type: none"> <li>- descrivere la semantica denotazionale di un linguaggio formale;</li> <li>- generare un linguaggio formale attraverso opportune regole (sintassi);</li> <li>- costruire macchine per il riconoscimento di un linguaggio formale.</li> </ul> </li> <li>• <i>Capacità di apprendere</i> <ul style="list-style-type: none"> <li>- Lo studente apprenderà la capacità di descrivere formalmente sintassi e semantica di linguaggi di programmazione.</li> </ul> </li> </ul>
<p>Contenuti di insegnamento</p>	<ol style="list-style-type: none"> <li>1. Introduzione ai linguaggi di programmazione ed ai linguaggi formali: Problemi, cenni a calcolabilità e linguaggi di programmazione. Interpretazione e compilazione. Gerarchia di linguaggi di programmazione e di macchine astratte. Aree di ricerca dell'informatica teorica, panoramica su e relazioni tra sintassi e semantica, alberi di derivazione. Regole di produzione, esempi di linguaggi formali. Ore lezione frontale: 11</li> <li>2. Grammatiche generative: Linguaggi formali e monoidi liberi generati da un insieme. Generazione e riconoscimento di linguaggi formali. Esempi di grammatiche generative. Correttezza di una grammatica. Ore lezione frontale: 11 Ore esercitazione: 1</li> <li>3. Linguaggi liberi da contesto e linguaggi dipendenti da contesto: Definizioni ed esempi di linguaggi liberi da contesto. Definizioni ed esempi di linguaggi dipendenti da contesto. Grammatiche e linguaggi monotoni. Ore lezione frontale: 3</li> </ol>

	<p>Ore esercitazione in aula: 1</p> <p>4. Linguaggi liberi da contesto: Alberi di derivazione. Principio di sostituzione di sottoalberi. Pumping lemma per i linguaggi liberi da contesto. Ore lezione frontale: 8 Ore esercitazione in aula: 10 (comprehensive di ore per la preparazione e revisione della prova in itinere)</p> <p>5. Grammatiche e macchine: Classificazione delle grammatiche secondo Chomsky. Teorema della Gerarchia di Chomsky. Operazioni sui linguaggi e proprietà di chiusura delle classi di linguaggi rispetto alle operazioni. Ore lezione frontale: 11 Ore esercitazione in aula: 6</p> <p>6. Automi: Automi a stati finiti deterministici e non deterministici. Linguaggi a stati finiti. Ore lezione frontale: 3 Ore esercitazione in aula: 3</p> <p>7. Linguaggi regolari ed espressioni regolari: Definizioni e proprietà delle espressioni regolari. Teorema di Kleene. Pumping Lemma per i linguaggi regolari. Ore lezione frontale: 5 Ore esercitazione in aula: 9 (comprehensive di ore per la preparazione alla prova scritta finale)</p> <p>8. Modello del compilatore: Analizzatore lessicale, analizzatore sintattico, analizzatore semantico, generazione e ottimizzazione del codice. Tabella dei simboli. Ore lezione frontale: 2</p> <p>9. Linguaggi di programmazione: Sintassi e semantica di un linguaggio di programmazione. Cenni all'evoluzione dei linguaggi di programmazione. Cenni ai paradigmi di programmazione. Ore lezione frontale: 2</p>
--	---

<b>Programma</b>	
Testi di riferimento	<p>1) G. Semeraro, Elementi di teoria dei linguaggi formali, ilmiolibro.it (2017), <a href="http://ilmiolibro.kataweb.it/libro/informatica-e-internet/317883/elementi-di-teoria-dei-linguaggi-formali/">http://ilmiolibro.kataweb.it/libro/informatica-e-internet/317883/elementi-di-teoria-dei-linguaggi-formali/</a></p> <p>2) M. Gabrielli, S. Martini, Linguaggi di Programmazione, Principi e paradigmi. 2/ed., McGraw-Hill (2011).</p>
Note ai testi di riferimento	<p>Le trasparenze mostrate a lezione, dispense integrative e tracce di esempio delle prove d'esame sono rese disponibili nella piattaforma di e-learning del Dipartimento di Informatica: <a href="http://informatica2.di.uniba.it">http://informatica2.di.uniba.it</a></p>
Metodi didattici	Lezioni frontali: 56 ore

<p>Metodi di valutazione (indicare almeno la tipologia scritto, orale, altro)</p>	<p>Esercitazioni in aula: 30 ore</p> <ol style="list-style-type: none"> <li>1) Appelli d'esame (al termine dell'insegnamento) - L'esame consta di una prova scritta articolata come segue: esecuzione di esercizi sulla teoria dei linguaggi formali, enunciazione di definizioni, dimostrazione di teoremi della teoria dei linguaggi formali, quesiti su linguaggi di programmazione e compilatori. La durata della prova varia in base alla tipologia di quesiti da risolvere, non superando comunque i 90 minuti. Durante la prova è ammessa esclusivamente la consultazione di: tavola relativa alle proprietà di chiusura dei linguaggi; elenco delle proprietà delle espressioni regolari.</li> <li>2) Prova intermedia (durante lo svolgimento dell'insegnamento) - E' prevista una prova scritta intermedia, articolata su una specifica parte del programma indicata dal docente prima dell'interruzione delle lezioni prevista da calendario accademico. Il superamento della prova intermedia esonera lo studente dallo svolgimento di una parte della prova d'esame, purché essa sia sostenuta nel primo appello di giugno.</li> </ol>
<p>Criteri di valutazione (per ogni risultato di apprendimento atteso su indicato, descrivere cosa ci si aspetta lo studente conosca o sia in grado di fare e a quale livello al fine di dimostrare che un risultato di apprendimento è stato raggiunto e a quale livello)</p>	<p>Nella prova scritta lo studente dovrà dimostrare di aver acquisito la capacità di:</p> <ul style="list-style-type: none"> <li>- classificare un linguaggio formale nella gerarchia di Chomsky, mediante la costruzione di una dimostrazione formale;</li> <li>- generare, descrivere e riconoscere un linguaggio formale, mediante la costruzione di grammatiche generative, la definizione di espressioni regolari, e la progettazione di opportune macchine (automi).</li> <li>- descrivere le tecniche di analisi e traduzione dei linguaggi di programmazione.</li> </ul> <p>La valutazione della prova è espressa in trentesimi. La prova intermedia è superata con un minimo di 16/30, conseguito svolgendo correttamente almeno uno dei quesiti posti. La prova d'appello è superata con un minimo di 18/30. Per gli studenti non esonerati, il voto finale d'esame è quello riportato nella prova scritta d'appello. Per gli studenti esonerati il voto finale d'esame è la media delle valutazioni riportate nella prova intermedia ed in quella del primo appello di giugno.</p> <p>La determinazione del voto tiene conto dei seguenti aspetti:</p> <ol style="list-style-type: none"> <li>1) la correttezza della soluzione o della dimostrazione formale proposta;</li> <li>2) la completezza della soluzione o della dimostrazione formale proposta;</li> <li>3) la logica seguita dallo studente nel proporre la soluzione o dimostrazione formale;</li> <li>4) l'utilizzo di un adeguato formalismo per la descrizione della soluzione o della dimostrazione proposta.</li> </ol> <p>Per superare la prova d'esame o la prova intermedia, lo studente deve essere in grado di proporre una soluzione che soddisfi almeno l'aspetto 1). I voti superiori al minimo</p>

	vengono attribuiti agli studenti la cui soluzione soddisfi anche gli altri aspetti.
Altro	Si suggerisce la partecipazione alle lezioni di esercitazione, che prevedono lo svolgimento in aula di esercizi tratti da precedenti prove d'esame. Per gli studenti non frequentanti, si suggerisce di svolgere gli esercizi di preparazione alla prova d'esame messi a disposizione dal docente nella piattaforma di e-learning, e di utilizzare lo strumento del ricevimento per verificare la correttezza delle soluzioni proposte.