



Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	Linguaggi di Programmazione	
Corso di studio	Informatica (track N-Z)	
Anno Accademico	2025/26	
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS)	9 CFU	
Settore Scientifico Disciplinare	INFO-01/A	
Lingua di erogazione	Italiano	
Anno di corso	Primo	
Periodo di erogazione	2° semestre, le date esatte sono riportate nel manifesto/regolamento	
Obbligo di frequenza	La frequenza è fortemente raccomandata	
Sito web del corso di studio	https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea	

Docente/i	
Nome e cognome	Marco de Gemmis
Indirizzo mail	marco.degemmis@uniba.it
Telefono	080-5443283
Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n.761, 7° piano.
Sede virtuale	Piattaforma e-learning UNIBA - https://elearning.uniba.it/
Sito web del docente	https://www.uniba.it/it/docenti/degemmis-marco
Ricevimento (giorni, orari e modalità, es. su appuntamento)	Su appuntamento

Syllabus



Obiettivi formativi	L'insegnamento si propone di introdurre: • i fondamenti teorici dei linguaggi di programmazione; • le tecniche sottostanti la progettazione dei linguaggi di programmazione. Studenti/esse acquisiranno la conoscenza della teoria dei linguaggi formali, saranno in grado di comprendere i meccanismi secondo i quali si implementa un linguaggio di programmazione e sapranno associare le operazioni svolte dal compilatore ai fondamenti teorici dei linguaggi formali.
Prerequisiti	Le seguenti conoscenze preliminari facilitano ed accelerano la comprensione degli argomenti dell'insegnamento: • da Programmazione: basi della programmazione imperativa, ricorsione, processo di compilazione; • da Matematica Discreta: teoria degli insiemi, relazioni, principio di induzione, strutture algebriche elementari (in particolare, concetto di monoide libero), dimostrazioni di tipo induttivo e deduttivo, definizione di funzione iniettiva, suriettiva, biettiva. • Raccomandazione: al fine colmare eventuali lacune rispetto ai prerequisiti, si suggerisce di consultare il materiale didattico relativo al percorso di "Linguaggi di Programmazione" disponibile nella piattaforma: https://elearning.uniba.it/
Contenuti di insegnamento (Programma)	La numerazione degli argomenti ne indica l'ordine di presentazione e di studio. Inoltre, l'indice di ogni argomento è utile per ritrovare i relativi contenuti nei testi di riferimento. 1. <u>Introduzione ai linguaggi di programmazione ed ai linguaggi formali:</u> Problemi, Macchine di Turing, 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: 9 Ore esercitazione in aula: 1 2. <u>Grammatiche generative:</u> 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: 9 Ore esercitazione in aula: 1 3. <u>Linguaggi liberi da contesto e linguaggi dipendenti da contesto:</u> Definizioni ed esempi di linguaggi liberi da contesto. Definizioni ed esempi di linguaggi dipendenti da contesto. Grammatiche e linguaggi monotoni. Ore lezione frontale: 3 Ore esercitazione in aula: 1 4. <u>Linguaggi liberi da contesto:</u> Alberi di derivazione. Principio di sostituzione di sottoalberi. Pumping lemma per i linguaggi liberi da contesto. Ore lezione frontale: 9 Ore esercitazione in aula: 9, comprensive di ore per la preparazione alla prova in itinere 5. <u>Grammatiche e macchine:</u> 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: 14 Ore esercitazione in aula: 6 6. <u>Automi:</u> Automi a stati finiti deterministici e non deterministici. Linguaggi a stati finiti. Ore lezione frontale: 3 Ore esercitazione in aula: 2 7. <u>Linguaggi regolari ed espressioni regolari:</u> Definizioni e proprietà delle espressioni regolari. Teorema di Kleene. Pumping Lemma per i linguaggi regolari. Ore lezione frontale: 7 Ore esercitazione in aula: 10, comprensive di ore per la preparazione alla prova scritta finale. 8. <u>Modello del compilatore:</u> Analizzatore lessicale, analizzatore sintattico, analizzatore semantico, generazione e ottimizzazione del codice. Tabella dei simboli. Ore lezione frontale: 2
Testi di riferimento	1. [Semeraro] G. Semeraro. Elementi di teoria dei linguaggi formali, ilmiolibro.it (2017). http://ilmiolibro.kataweb.it/libro/informatica-e-internet/317883/elementi-di-teoria-dei-linguaggi-formali 2. [Gabbrielli] M. Gabbrielli, S. Martini. Linguaggi di Programmazione, Principi e paradigmi. 2/ed., McGraw-Hill (2011). 3. [Hopcroft] J. E. Hopcroft, R. Motwani, J. D. Ullman. Automi, Linguaggi e calcolabilità. Pearson (2018). Nella sezione seguente è specificata in dettaglio la corrispondenza tra gli argomenti del programma ed i capitoli dei testi di riferimento. Studenti/esse che lo desiderano possono ottenere i testi in prestito dalla Biblioteca. Si suggerisce di verificare la disponibilità dei testi mediante il Sistema Bibliotecario di Ateneo e contattare la biblioteca per concordare il prestito.
Note ai testi di riferimento	Si specificano di seguito, per ogni argomento del programma, i capitoli dei testi dai quali studiare. Si specificano di seguito, per ogni argomento del programma, i capitoli dei testi dai quali studiare. 1. <u>Introduzione ai linguaggi di programmazione ed ai linguaggi formali:</u> [Gabbrielli] Capitolo 1; [Semeraro] Capitolo 1. [Hopcroft] Capitolo 8. 2. <u>Grammatiche generative:</u> [Semeraro] Capitoli 1 e 2. 3. <u>Linguaggi liberi da contesto e linguaggi dipendenti da contesto:</u> [Semeraro] Capitolo 3. 4. <u>Linguaggi liberi da contesto:</u> [Semeraro] Capitolo 4. [Hopcroft] Capitoli 5 e 6. 5. <u>Grammatiche e macchine:</u> [Semeraro] Capitolo 5.



	[Hopcroft] Appendice A 6. <u>Automi:</u> [Semeraro] Capitolo 6. [Hopcroft] Capitoli 1 e 2. 7. <u>Linguaggi regolari ed espressioni regolari:</u> [Semeraro] Capitolo 7. [Hopcroft] Capitoli 3 e 4. 8. <u>Modello del compilatore:</u> [Gabrielli] Capitoli 2, 3 e 4 per approfondimenti su analisi lessicale e sintattica. Per ogni argomento è inoltre disponibile una dispensa, corrispondente ai contenuti mostrati in aula durante le lezioni, che non è sostitutiva dei testi di riferimento. Elenco del materiale disponibile su piattaforma di e-learning UNIBA https://elearning.uniba.it/: • dispense in formato pdf degli argomenti del programma; • esercizi con relative soluzioni; • tracce di esempio delle prove d'esame, alcune delle quali con relative soluzioni • tracce di esempio della prova in itinere.		
Organizzazione della didattica			
Ore			
Totali	Didattica frontale	Pratica (esercitazione)	Studio individuale
225 ore	56 ore	30 ore	139 ore
CFU/ETCS			
9 CFU	7 CFU	2 CFU	

Metodi didattici	
• Lezioni frontali condotte con l'ausilio di dispense proiettate in aula e rese disponibili tramite piattaforma di e-learning UNIBA prima delle lezioni; • Svolgimento in aula di due tipologie di esercitazione: (a) esercizi svolti interamente dal docente con indicazione delle soluzioni; (b) esercitazioni guidate in cui gli studenti risolvono insieme al docente quesiti simili a quelli da affrontare nelle prove d'esame. • Entrambe le tipologie di esercitazione sono svolte con l'obiettivo di acquisire dimestichezza con i modelli di computazione (grammatiche, automi, etc.) utili alla classificazione, alla generazione ed al riconoscimento di linguaggi formali.	



--	--

Risultati di apprendimento previsti	
Conoscenza e capacità di comprensione	• Comprensione della relazione tra problemi, algoritmi, linguaggi formali e linguaggi di programmazione; • Comprensione della Teoria dei Linguaggi Formali; • Conoscenza delle tecniche di analisi e traduzione dei linguaggi di programmazione; • Conoscenza dei fondamenti teorici alla base delle componenti di analisi e traduzione dei linguaggi di programmazione (scanner, parser).
Conoscenza e capacità di comprensione applicate	• Capacità di classificare un linguaggio formale nella gerarchia di Chomsky; • Capacità di generare, descrivere e riconoscere un linguaggio formale.
Competenze trasversali	Autonomia di giudizio • Capacità di valutare la correttezza della soluzione proposta ai problemi relativi alla teoria dei linguaggi formali Abilità comunicative • Capacità di formalizzare e comunicare la soluzione proposta a problemi relativi alla teoria dei linguaggi formali Capacità di apprendere in modo autonomo • Capacità di associare un problema pratico ai modelli appresi (ad esempio riconoscimento di stringhe o frasi in linguaggio naturale mediante espressioni regolari)

Valutazione	
Modalità di verifica dell'apprendimento	Prova d'esame L'esame consiste in una prova scritta, nella quale si richiede di rispondere alle seguenti tipologie di quesiti: • svolgimento di esercizi sulla teoria dei linguaggi formali; • enunciazione di definizioni sulla teoria dei linguaggi formali; • dimostrazione di teoremi della teoria dei linguaggi formali; • quesiti aperti sul modello di compilatore. Regolamentazione della prova d'esame: • La prova d'esame consisterà nella risoluzione di 3 quesiti;



	● durata della prova: varia tra 60 e 90 minuti, in base alla tipologia di quesiti da risolvere; ● materiale consultabile: tavola relativa alle proprietà di chiusura dei linguaggi, elenco delle proprietà delle espressioni regolari; ● la valutazione è espressa in trentesimi; ● comunicazione dei risultati: sistema esse3. <u>Prova intermedia</u> È una prova scritta, NON obbligatoria, riguardante una specifica parte del programma indicata dal docente in prossimità dell'interruzione delle lezioni stabilita da Regolamento Didattico e Manifesto degli Studi. Si richiede: ● esecuzione di due esercizi sulla teoria dei linguaggi formali. Regolamentazione della prova intermedia: ● durata della prova: 60 minuti; ● materiale consultabile: nessuno; ● la prova è soggetta ad autovalutazione: il docente fornirà le soluzioni e le indicazioni per assegnare una valutazione ad ogni errore commesso (grave/non grave). La valutazione finale, in base agli errori commessi, potrà essere sufficiente/insufficiente; ● Il superamento della prova intermedia NON esonera lo studente dallo svolgimento di una parte della prova d'esame. L'obiettivo della prova è quello di dare un riscontro allo studente sul proprio livello di comprensione degli argomenti dell'insegnamento. Dopo la prova intermedia, si suggerisce di contattare il docente per un ricevimento in caso di dubbi. <u>Simulazione prova d'esame</u> Al termine delle lezioni, si terrà una simulazione della prova d'esame, che consisterà nell'esecuzione di esercizi tratti da prove d'esame precedenti. Gli esercizi saranno svolti in autonomia nel tempo solitamente concesso in sede d'esame in modo da avere un riscontro della propria preparazione <i>prima</i> dell'esame. Al termine della prova, il docente fornirà le relative soluzioni.
Criteri di valutazione	● Conoscenza e capacità di comprensione: ○ Capacità di enunciare con rigore definizioni e teoremi della teoria dei linguaggi formali; ○ Capacità di descrivere con precisione e chiarezza le tecniche ed i fondamenti teorici alla base dei meccanismi di analisi e traduzione dei linguaggi di programmazione. ● Conoscenza e capacità di comprensione applicate: ○ Capacità di applicare i teoremi della teoria dei linguaggi formali alla soluzione di problemi; ○ Capacità di classificare un linguaggio formale nella gerarchia di Chomsky, mediante la costruzione di una dimostrazione formale; ○ Capacità di generare, descrivere e riconoscere un linguaggio formale, mediante la costruzione di grammatiche, la definizione di espressioni regolari, e la progettazione di opportune macchine. ● Autonomia di giudizio: ○ Modalità adottate per la verifica delle soluzioni proposte (ad esempio, verifica di correttezza di una grammatica).



	● Abilità comunicative: ○ Chiarezza nella descrizione delle soluzioni proposte ai quesiti. ● Capacità di apprendere: ○ Capacità di astrazione, di ragionamento per analogia e dimostrazione di creatività nella risoluzione dei quesiti.
Criteri di misurazione dell'apprendimento e di attribuzione del voto finale	<u>Prova d'esame</u> La valutazione della prova è espressa in trentesimi. Ad ogni quesito da svolgere sarà attribuito un punteggio, assicurando che la somma dei punteggi sia pari a 30 (ad esempio, 3 quesiti ad ognuno dei quali sono attribuiti 10 punti). La determinazione del voto attribuito alla risposta data ad ogni quesito tiene conto dei seguenti criteri: 1) la correttezza della soluzione, della dimostrazione formale, o della definizione fornita; 2) la completezza della soluzione, della dimostrazione formale o della definizione fornita; 3) la logica seguita dallo studente nel proporre la soluzione o la dimostrazione formale; 4) l'utilizzo di un adeguato formalismo per la descrizione della soluzione, della dimostrazione o della definizione. Per ottenere una valutazione sufficiente, la risposta fornita al quesito deve soddisfare almeno il criterio 1). I voti superiori alla sufficienza sono attribuiti se le soluzioni proposte soddisfano anche gli altri criteri. Il voto finale è dato dalla somma dei punteggi ottenuti sui singoli quesiti. La prova è superata con voto maggiore o uguale a 18. Attribuzione della lode: la lode è attribuita quando la logica seguita dallo studente del proporre la soluzione evidenzia particolari capacità di astrazione, ragionamento per analogia, creatività. <u>Prova intermedia</u> I criteri adottati per la valutazione della prova intermedia sono gli stessi 1)-4) della prova finale. Il voto potrà essere sufficiente/insufficiente. L'esito della prova intermedia NON concorre alla valutazione finale.
Altro	Si suggerisce agli studenti di affidarsi esclusivamente alle informazioni/comunicazioni fornite sui siti ufficiali del Dipartimento di Informatica, ovvero sui gruppi social solo se costituiti e amministrati esclusivamente dai docenti dei relativi insegnamenti: ● https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea ● https://www.uniba.it/it/ricerca/dipartimenti/informatica I programmi degli insegnamenti sono disponibili al seguente URL: ● Piano di studi e schede insegnamenti Le informazioni che tutti gli studenti dovrebbero conoscere sono scritte nei Regolamenti Didattici e Manifesti degli studi disponibili sul sito: ● https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea <i>Si suggerisce agli studenti di diffidare dalle informazioni e dei materiali circolanti su siti o gruppi social non ufficiali, poiché spesso sono risultati non affidabili, non</i>



corretti o incompleti. Per ogni dubbio, fissare un incontro con il docente secondo le modalità previste per il ricevimento.

Link all'insegnamento nella piattaforma di e-learning:

<https://elearning.uniba.it/course/view.php?id=12881>

Suggerimenti per una corretta preparazione, in particolare per studenti/esse non frequentanti:

- 1) Studiare gli argomenti teorici nell'ordine suggerito prima di cimentarsi negli esercizi. L'ordine con il quale sono presentati gli argomenti è fondamentale per la comprensione degli stessi. Tentare di risolvere gli esercizi apprendendo solo dalle tracce di esempio e relative soluzioni non consente di acquisire le basi culturali per poter superare l'esame.
Comprendere perché si arriva ad una certa soluzione è molto più importante che svolgere molti esercizi senza comprenderne del tutto le basi teoriche.
- 2) Svolgere gli esercizi, dei quali sono fornite le soluzioni, disponibili sulla piattaforma di e-learning, in modo da confrontare le proprie soluzioni con quelle fornite dal docente.
- 3) Svolgere gli esercizi tratti da prove d'esame per valutare la propria preparazione.
- 4) Contattare il docente per eventuali chiarimenti e/o verificare le proprie soluzioni agli esercizi svolti in autonomia prima di iscriversi alla prova d'esame.
- 5) In caso di mancato superamento della prova, contattare il docente per la revisione del proprio elaborato e la comprensione degli errori commessi.



Main information on the course

Course name	Programming Languages	
Degree	Informatics (track N-Z)	
Academic year	2025/26	
European Credit Transfer and Accumulation System (ECTS), in Italian Crediti Formativi Universitari (CFU)	9 CFU (each CFU corresponds to 25 hours (h) of student's time); CFU are of type T1, T2 or T3 T1 = 8 h lecture + 17 h individual study T2 = 15 h practice + 10 h individual study T3 = 25 h individual study	
Settore Scientifico Disciplinare		
Course language	Italian	
Course year	First	
Course period	Second Semester - exact dates can be found in the didactic regulations	
Course attendance requirement	None, but it is highly recommended to attend classes	
Website of the Degree	https://www.uniba.it/it/corsi/informatica/corso-di-laurea-in-informatica	

Teacher(s)

Name and Surname	Marco de Gemmis
email	marco.degemmis@uniba.it
phone	080-5443283
office	Department of Computer Science, Campus, Via E. Orabona 4, 70126, Bari. office n.761, 7th floor.
e-learning platform	UNIBA e-learning platform - https://elearning.uniba.it/
Teacher's homepage	https://www.uniba.it/it/docenti/degemmis-marco
Office hours	students are required to send an email to the instructor to ask for an appointment.

Syllabus

Course goals	The course aims to introduce: - the theoretical foundations of programming languages; - the techniques underlying the design of programming languages. The student will acquire the knowledge of the theory of formal languages, will be able to understand the mechanisms according to which a programming language is implemented and will be able to associate the operations carried out by the compiler to the theoretical foundations of formal languages.
Prerequisites/requirements	The following preliminary knowledge facilitates and accelerates the understanding of the topics of the course: - from "Programming": basics of imperative programming, recursion, compilation process. - from "Discrete Mathematics": set theory, relations, principle of induction, elementary algebraic structures (in particular, concept of monoid and free monoid generated by a set), inductive and deductive proofs, definition of injective, surjective, bijective function. - Recommendation: in order to fill any gaps with respect to prerequisites, it is suggested to consult the teaching materials related to the "Programming Languages" pre-course available in the platform: https://elearning.uniba.it/



Course program	The numbering of topics indicates their order of presentation and study. In addition, the index of each topic is useful for finding the related content in reference texts. <u>1. Introduction to programming languages and formal languages:</u> Problems, Turing machines, computability and programming languages. Interpreters and compilers. Hierarchy of programming languages and abstract machines. Research areas of theoretical computer science, overview of and relationships between syntax and semantics, derivation trees. Production rules, examples of formal languages. Lecture: 9 Hands-on: 1 <u>2. Generative grammars:</u> Formal languages and free monoids generated by a set. Generation and recognition of formal languages. Generative grammars. Examples of generative grammars. Derivations. Correctness of a grammar. Lecture: 9 Hands-on: 1 <u>3. Context-free languages and context-dependent languages:</u> Definitions and examples of context-free languages. Definitions and examples of context-dependent languages. Monotonous grammars and languages. Lecture: 3 Hands-on: 1 <u>4. Context-free languages:</u> Derivation trees. Subtree replacement principle. Pumping lemma for context free languages. Lecture: 9 Hands-on: 9, including hours for preparation for the mid-term test. <u>5. Grammars and machines:</u> Classification of grammars according to Chomsky. Chomsky's Hierarchy Theorem. The empty string lemma. Operations on languages and closure properties of language classes with respect to the operations. Lecture: 14 Hands-on: 6 <u>6. Automata:</u> Deterministic and non-deterministic finite state automata. Finite state languages. Lecture: 3 Hands-on: 2 <u>7. Regular languages and regular expressions:</u> Definitions and properties of regular expressions. Kleene's theorem. Pumping Lemma for regular languages. Lecture: 7 Hands-on: 10, including hours for preparation for the mid-term test. <u>8. Compiler model:</u> Lexical analyzer, syntactic analyzer, semantic analyzer, code generation and optimization. Symbol table. Lecture: 2
Books of reference	



	1. [Semeraro] G. Semeraro. Elementi di teoria dei linguaggi formali, ilmiolibro.it (2017), http://ilmiolibro.kataweb.it/libro/informatica-e-internet/317883/elementi-di-teoria-dei-linguaggi-formali 2. [Gabrielli] M. Gabrielli, S. Martini. Linguaggi di Programmazione, Principi e paradigmi. 2/ed., McGraw-Hill (2011). 3. [Hopcroft] J. E. Hopcroft, R. Motwani, J. D. Ullman. Automi, Linguaggi e calcolabilità. Pearson (2018). The following section details the correspondence between the program topics and the chapters of the reference books. Students can obtain the books on loan from the Library. It is suggested that students check whether the texts are available through the University Library System https://opac.uniba.it/easyweb/w8018/index.php? and contact the library to arrange borrowing.
Notes to the books	The chapters of the texts from which to study are specified below for each topic in the program. 1. <u>Introduction to programming languages and formal languages:</u> [Gabrielli] Chapter 1; [Semeraro] Chapter 1. [Hopcroft] Chapter 8. 2. <u>Generative grammars:</u> [Semeraro] Chapters 1 and 2. 3. <u>Context-free languages and context-dependent languages:</u> [Semeraro] Chapter 3. 4. <u>Context-free languages:</u> [Semeraro] Chapter 4. [Hopcroft] Chapters 5-6. 5. <u>Grammars and machines:</u> [Semeraro] Chapter 5. 6. <u>Automata:</u> [Semeraro] Chapter 6. [Hopcroft] Chapters 1-2. 7. <u>Regular languages and regular expressions:</u> [Semeraro] Chapter 7. [Hopcroft] Chapters 3-4. 8. <u>Compiler model:</u> [Gabrielli] Chapters 2, 3 and 4 for further information on lexical and syntactic analysis. A handout is also available for each topic, corresponding to the content projected in the classroom during lectures, which is not a substitute for the reference books. List of materials available on e-learning platform https://elearning.uniba.it/: • pdf handouts of the program topics; • exercises with solutions; • samples of the exam tests, some of them with solutions samples of the mid-term tests.
Organization of the didactic activities	
Hours	



Total	Lectures	Practice sessions	Project work	Individual study
225	56	30	0	139
CFU/ETCS				
9	7	2	0	

Teaching methods	
	- Classes given with the help of slides made available through the e-learning platform before classes; - Two types of exercises carried out in practice sessions: - exercises carried out by the professor; - guided exercises in which students solve, with the professor, problems to be addressed in the exam test. Both types of exercises are carried out with the aim of acquiring familiarity with computation models (grammars, automata, etc.) useful for the classification, generation and recognition of formal languages.

Expected learning outcomes	
Knowledge and understanding	- Understanding of the relationship between problems, algorithms, formal languages and programming languages; - Understanding of the Theory of Formal Languages; - Knowledge of analysis and translation techniques of programming languages; - Knowledge of the theoretical foundations underlying analysis and translation components of programming languages (scanner, parser).
Applying knowledge and understanding	- Ability to classify a formal language in the Chomsky hierarchy; - Ability to generate, describe and recognize a formal language.
Other skills	Autonomy of judgment - Ability to evaluate the correctness of the proposed solution to problems related to the theory of formal languages. Communication skills - Ability to formalize and communicate the proposed solution to problems related to the theory of formal languages. Ability to learn independently - Ability to associate a practical problem with learned models (e.g. recognition of strings or sentences in natural language using regular expressions).

Assessment	
Assessment methods	Exam The exam consists of a written test, in which it is required to answer the following types of questions: - carrying out exercises on the theory of formal languages; - enunciation of definitions on the theory of formal languages;



	• proof of theorems on the theory of formal languages; • open questions about the compiler model. Regulation of the exam: • the exam consists of three questions; • duration of the test: it varies between 60 and 90 minutes, depending on the type of problems to be solved; • material that can be consulted during the test: table relating to the closure properties of languages, list of properties of regular expressions; • the evaluation is expressed in thirtieths; • publication of results: ESSE3 platform. <u>Mid-term partial exam</u> It is a written test, NOT mandatory, concerning a specific part of the syllabus indicated by the instructor around the time of the break in lectures established by the Didactic Regulations and the Course Catalogue. The test requires: • completion of two exercises on the theory of formal languages. Regulations for the midterm test: • Duration: 60 minutes; • Allowed materials: none; • The test is self-assessed: the instructor will provide the solutions and guidelines for assigning a grade to each error made (serious / non-serious). The final evaluation, based on the errors committed, may be pass or fail; • Passing the midterm test does NOT exempt the student from taking part of the final exam. The purpose of the test is to provide feedback to the student on their level of understanding of the course topics. After the midterm test, it is recommended to contact the instructor during office hours in case of any doubts. <u>Exam test simulation</u> At the end of the lessons there will be a simulation of the exam, which will consist in the execution of exercises taken from previous exam tests. The exercises will be carried out independently by the students in the time usually allowed during the exam in order to have a feedback of their preparation before the exam. The professor will provide the solutions at the conclusion of the test.
Evaluation criteria	• <i>Knowledge and understanding:</i> ○ Ability to rigorously enunciate definitions and theorems of the theory of formal languages; ○ Ability to describe with accuracy and clarity the techniques and theoretical foundations underlying the mechanisms of analysis and translation of programming languages. • <i>Applied knowledge and understanding:</i> ○ Ability to apply the theorems of the theory of formal languages to problem solving; ○ Capacità di classificare un linguaggio formale nella gerarchia di Chomsky, mediante la costruzione di una dimostrazione formale ○ Ability to classify a formal language in the Chomsky hierarchy, by constructing a formal proof; ○ Ability to generate, describe and recognize a formal language, through the construction of grammars, the definition of regular expressions, and the design of appropriate machines. • <i>Autonomy of judgment:</i> ○ Methods adopted for the verification of the proposed solutions (for example, verification of correctness of a grammar). • <i>Communication skills:</i> ○ Clarity of the proposed solutions to the questions. • <i>Learning skills:</i>



	○ Ability to abstract, reasoning by analogy and demonstration of creativity in solving questions.
Measurements and final grade	<u>Exam test</u> The evaluation of the exam test is expressed over a 30-point scale. Each question to be completed will be assigned a score, ensuring that the sum of the scores is equal to 30 (for example, 3 questions, each of which is assigned 10 points). The determination of the grade attributed to the answer given to each question considers the following criteria: 1) the correctness of the solution, formal proof, or definition provided; 2) the completeness of the solution, formal proof, or definition provided; 3) the logic followed by the student in proposing the solution or formal demonstration; 4) the use of an adequate formalism for the description of the solution, demonstration or definition. To obtain a sufficient grade (18/30), the answer provided to the question must satisfy at least criterion 1). Higher grades are awarded if the proposed solutions also satisfy the other criteria. The final grade is given by the sum of the scores obtained on the individual questions. The test is passed with a grade greater than or equal to 18. Attribution of honors: honors are awarded when the logic followed by the student in proposing the solution or formal proof highlights particular capacities of abstraction, reasoning by analogy, creativity. <u>Mid-term partial exam</u> The criteria adopted for the evaluation of the mid-term partial exam are the same as 1)-4) of the final exam. The grade may be sufficient/insufficient. The result of the mid-term partial exam does NOT contribute to the final grade.
Further information	It is suggested that students rely exclusively on the information/communication provided on the official websites of the Department of Computer Science, or on social groups only if they are established and administered exclusively by the faculty members of the relevant subjects: ● https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea ● https://www.uniba.it/it/ricerca/dipartimenti/informatica The programs of the courses are available at the following URL: ● Elenco dei Programmi del Corso di Studi in Informatica Information that all students need to know is available in the Didactic Regulations and Study Prospectus available on the website: ● https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea <i>Students are suggested to be wary of information and materials circulating on unofficial sites or social groups, as they are often found to be unreliable, incorrect or incomplete. For any doubts, arrange a meeting with the teachers in accordance with the reception arrangements.</i> Link to the course on the e-learning platform: https://elearning.uniba.it/course/view.php?id=12881



Tips for proper preparation, especially for non-attending students:

- 1) Study the theoretical topics in the suggested order before engaging in the exercises. The order in which the topics are presented is fundamental for understanding them. Trying to solve the exercises by learning only from the example exam tests and related solutions does not allow the student to acquire the cultural bases to be able to pass the exam.

Understanding the reasons that led to a certain solution is much more important than doing many exercises without fully understanding the theoretical bases.

- 2) Carry out the exercises, whose solutions are provided by the instructor on the e-learning platform, in order to compare your own solutions with those provided by the teacher.
- 3) Carry out the exercises taken from exam tests to evaluate your preparation.
- 4) Contact the instructor for any clarifications and / or verify your solutions to the exercises carried out independently before registering for the exam.

In case of failure to pass a test, contact the instructor for the revision of your test and the understanding of the errors made.