



Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	Algoritmi e Strutture Dati		
Corso di studio	Informatica e Comunicazione Digitale		
Anno Accademico	2023/24		
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS)	9 CFU		
Settore Scientifico Disciplinare	INF/01		
Lingua di erogazione	Italiano		
Anno di corso	Secondo		
Periodo di erogazione	1^ 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/informatica-icd-taranto-270/laurea-triennale-in-informatica-e-comunicazione-digitale-sede-di-taranto-d.m.-270		
Docente/i			
Nome e cognome	Stefano Ferilli		
Indirizzo mail	stefano.ferilli@uniba.it		
Telefono	080 544 2293		
Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n. 525, 5^ piano.		
Sede virtuale	Piattaforma ADA - https://elearning.di.uniba.it/		
Sito web del docente	http://lacam.di.uniba.it/~ferilli/ufficiale/ferilli.html		
Ricevimento (giorni, orari e modalità, es. su appuntamento)	Mercoledì 10-12 e su appuntamento		
Syllabus			
Obiettivi formativi	<ol style="list-style-type: none">1) Conoscenza di una metodologia di progetto formale basata sulla astrazione dei dati.2) Capacità di rappresentare in modo formale diversi tipi di problemi.3) Rudimenti della programmazione orientata agli oggetti attraverso l'uso di dati astratti.4) Principi dell'algoritmica, in funzione del modo di utilizzo dello spazio di ricerca: le caratteristiche dei paradigmi selettivo e generativo e selezione di algoritmi fondamentali.		
Prerequisiti	<ul style="list-style-type: none">• Da Programmazione:<ul style="list-style-type: none">◦ Principi della programmazione strutturata e conoscenza di almeno un linguaggio di Programmazione imperativa;◦ Principi della programmazione modulare;◦ Concetto di tipo e Strutture dati predefinite;◦ Principi dell'astrazione funzionale e meccanismi di passaggio dei parametri.• Da Matematica Discreta:<ul style="list-style-type: none">◦ Insiemistica e capacità di comprendere e scrivere descrizioni formali		
Contenuti di insegnamento			



(Programma)	<p>Introduzione e Nozioni di base</p> <ul style="list-style-type: none"> • L'astrazione in programmazione e nel ciclo di vita del software • Astrarre le strutture di dati: le Algebre di dati • Classificazioni e caratteristiche delle strutture di dati <p>Strutture Dati Astratte notevoli</p> <ul style="list-style-type: none"> • Liste • Pile e Code • Insiemi e Dizionari • Alberi binari • Code con priorità • Alberi binari ed n-ari • Grafi <p>Algoritmica</p> <ul style="list-style-type: none"> • Cenni sulla analisi della complessità computazionale • Le tecniche algoritmiche • Tecniche selettive: enumerativa e backtracking • Tecniche generative: greedy e divide-et-impera <p>Esercitazioni su strutture e algoritmi</p> <ul style="list-style-type: none"> • Programmazione a oggetti e Linguaggio C++ • Applicazione a un gioco d'avventura 		
Testi di riferimento	<ul style="list-style-type: none"> • S. Ferilli "Astrazione Dati - con applicazioni ai Giochi d'Avventura". Boopen, 2016 • M. Cadoli, M. Lenzerini, P. Naggar, A. Schaerf. "Fondamenti della progettazione dei programmi". Città studi Edizioni, 1997. <p>Gli studenti che lo desiderano possono ottenere i testi in prestito dalla Biblioteca. Può convenire verificarne la disponibilità mediante il Sistema Bibliotecario di Ateneo https://opac.uniba.it/easyweb/w8018/index.php? e contattare la biblioteca per concordare il prestito.</p>		
Note ai testi di riferimento	<p>Il testo di Ferilli è adottato integralmente per le parti riguardanti le Strutture Dati Astratte e le Esercitazioni sui giochi d'avventura.</p> <p>Il testo di Cadoli et al. È adottato per le parti "Introduzione e Nozioni di base" e "Algoritmica" e "Linguaggio C++".</p>		
Organizzazione della didattica			
Ore			
Totali	Didattica frontale	Pratica (laboratorio, progetto, esercitazione, altro)	Studio individuale
225 ore	56 ore	30 ore	139 ore
CFU/ETCS			
9 CFU	7 CFU	2 CFU	
Metodi didattici			
<p>Lezioni frontali supportate da slide, esercitazioni in aula, svolgimento di tracce con correzione collettiva e autovalutazione.</p>			



Risultati di apprendimento previsti	
Conoscenza e capacità di comprensione	<ul style="list-style-type: none">○ Categorizzazione dei problemi mediante spazio di ricerca. Conoscenza dei paradigmi algoritmici selettivo e generativo, delle relative tecniche solutive e dei problemi più noti. Consapevolezza delle possibilità e dei limiti delle metodologie informatiche basate sulla programmazione orientata ad oggetti, indipendentemente dallo specifico ambiente di programmazione.○ Aspetti essenziali della disciplina che rimangono inalterati a fronte del cambiamento tecnologico, in modo da avere un sistema di riferimento culturale che trascende il tempo e le circostanze.
Conoscenza e capacità di comprensione applicate	<ul style="list-style-type: none">○ Progettazione efficace di algoritmi mediante la metodologia dei dati astratti. Capacità di progettare e valutare implementazioni alternative in base all'efficienza computazionale. Concetti di base della teoria della complessità e capacità di applicarli in semplici algoritmi di base.○ Capacità di comprendere i limiti della programmazione imperativa e di cogliere le opportunità offerte dalla programmazione orientata ad oggetti.
Competenze trasversali	<p>Autonomia di giudizio</p> <ul style="list-style-type: none">○ Capacità di valutare la qualità di una soluzione algoritmica in termini di efficienza e possibilità di riutilizzo. Capacità di valutare criticamente quanto appreso, formulando un proprio punto di vista ed essendo in grado di sostenerlo. <p>Abilità comunicative</p> <ul style="list-style-type: none">○ Adeguata appropriatezza espressiva e capacità di comunicazione di problematiche inerenti gli studi algoritmici anche ad interlocutori non esperti. <p>Capacità di apprendere in modo autonomo</p> <ul style="list-style-type: none">○ Capacità di adattare le conoscenze acquisite anche a nuovi contesti, nonché di aggiornarsi attraverso la consultazione delle fonti specialistiche del settore algoritmico. Capacità di reperire e consultare materiale bibliografico, di sintetizzarne i contenuti, anche in inglese, e di utilizzarlo in fase preparazione dell'esame.
Valutazione	
Modalità di verifica dell'apprendimento	<p>- tipo di valutazione utilizzata: voto in trentesimi</p> <p>- modalità di comunicazione dei risultati delle prove: comunicazione diretta in caso di non approvazione del caso di studio</p> <p>L'esame si articola in due prove:</p> <ul style="list-style-type: none">• prova di pratica (idoneità con giudizio indicativo)• prova orale (voto in trentesimi) <p>Per accedere alla prova orale è necessario aver svolto con valutazione positiva la</p>



	<p>prova pratica.</p> <p>La prova pratica può essere svolta sul computer portatile personale dello studente e consiste nello sviluppo di un programma funzionante che risolva un problema dato facendo uso delle strutture dati astratte e delle tecniche algoritmiche studiate nel corso.</p> <p>La prova di laboratorio può essere sostituita, su richiesta dello studente, da un caso di studio. La traccia del caso di studio sarà assegnata dal docente e, salvo diversi accordi, riguarderà la modifica o l'aggiunta di funzionalità in un gioco d'avventura fornito dal docente, facendo uso delle strutture dati astratte e delle tecniche algoritmiche studiate nel corso.</p> <p>Il caso di studio si intende rivolto a singoli studenti. Gli studenti interessati devono fare richiesta della traccia un mese circa prima della data d'esame scelta, e consegnare tutto l'elaborato entro l'ottavo giorno precedente la data dell'appello sotto forma di link per scaricare un file ZIP contenente tutto il materiale di progetto (progetto CodeBLOCKS compilabile ed eseguibile, realizzazioni intercambiabili delle strutture dati, documentazione in formato ODT e PDF).</p> <p>La prova orale finale è costituita, di norma, da tre domande che possono riguardare sia argomenti di natura teorica che soluzioni di problemi analoghi a quelli trattati durante il corso.</p> <p>I progetti sono validi per l'appello per cui sono stati chiesti.</p>
Criteri di valutazione	<ul style="list-style-type: none">● Conoscenza e capacità di comprensione:<ul style="list-style-type: none">○ Chiarezza e completezza delle risposte alle interrogazioni orali○ Correttezza e perspicacia delle risposte alle domande orali● Conoscenza e capacità di comprensione applicate:<ul style="list-style-type: none">○ Prova di laboratorio o Caso di studio○ Chiarezza e completezza della documentazione della prova pratica● Autonomia di giudizio:<ul style="list-style-type: none">○ Decisioni di progetto prese nella prova pratica○ Risoluzione di problemi durante la prova orale● Abilità comunicative:<ul style="list-style-type: none">○ Chiarezza e completezza delle risposte alle domande orali○ Chiarezza e completezza della documentazione del caso di studio● Capacità di apprendere:<ul style="list-style-type: none">○ Capacità di identificare le nozioni di base rilevanti per affrontare un nuovo argomento○ Capacità di eseguire una corretta ricerca delle fonti
Criteri di misurazione dell'apprendimento e di attribuzione del voto finale	<p>Saranno valutati i risultati di apprendimento previsti in base a:</p> <ul style="list-style-type: none">● Qualità, completezza ed ampiezza della prova pratica (compresa la documentazione).● Chiarezza, correttezza e completezza delle risposte alle domande orali.● Capacità di discernimento e di fornire consigli su questioni specifiche sulla base della conoscenza acquisita.



	<p>La prova pratica è valutata in termini di idoneità a sostenere la prova orale, con un giudizio per fasce (insufficiente, sufficiente, buono, ottimo). La prova orale è valutata in trentesimi.</p>
Altro	<p>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:</p> <ul style="list-style-type: none">● https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea● https://www.uniba.it/it/ricerca/dipartimenti/informatica● https://elearning.di.uniba.it/ <p>I programmi degli insegnamenti sono disponibili qui:</p> <ul style="list-style-type: none">● https://programmi.di.uniba.it/ <p>Le informazioni che tutti gli studenti dovrebbero conoscere sono scritte nei Regolamenti didattici e manifesti degli studi disponibili nel sito:</p> <ul style="list-style-type: none">● https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea <p>Si suggerisce agli studenti di diffidare delle informazioni e dei materiali circolanti su siti o gruppi social non ufficiali, poiché spesso sono risultati non affidabili, non corretti o incompleti. Per ogni dubbio, chiedere un incontro al docente secondo le modalità previste per il ricevimento.</p> <hr/> <p><i>Teams: 130cked</i></p>



Main information on the course

Course name	Algorithms and Data Structures		
Degree	Informatica e Comunicazione Digitale		
Academic year	2023/24		
European Credit Transfer and Accumulation System (ECTS), in Italian Crediti Formativi Universitari (CFU)	9 CFU		
Settore Scientifico Disciplinare	INF/01		
Course language	Italian		
Year of study	Second		
Semester	1st semester, the exact dates are indicated annually in the degree course regulations		
Obbligo di frequenza	It is highly recommended to attend classes		
Sito web del corso di studio	https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/computer-science/computer-science		
Teacher(s)			
Name and Surname	Stefano Ferilli		
email	stefano.ferilli@uniba.it		
phone	080 544 2293		
office	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n. 525, 5th floor.		
e-learning platform	Piattaforma ADA - https://elearning.di.uniba.it/		
Teacher's homepage	http://lacam.di.uniba.it/people/ferilli.html		
Office hours	appointment with email		
Syllabus			
Course goals	<ul style="list-style-type: none"> • Knowledge of a formal design methodology based on data abstraction • Ability to formally represent different kinds of problems • Basics of Object-Oriented programming through the use of abstract data • Principles of algorithmics, based on the way the search space is used: features of selective and generative paradigms, selection of fundamental algorithms 		
Prerequisites/requirements	<p>From Programming:</p> <ul style="list-style-type: none"> • Principles of structured programming and knowledge of at least one imperative programming language • Principles of modular programming • Concept of type and built-in data structures • Principles of functional abstraction and argument-passing mechanisms <p>From Discrete Mathematics:</p> <ul style="list-style-type: none"> • Set theory and ability to understand and write formal descriptions 		
Course program	<p>Introduction and Basic Notions</p> <ul style="list-style-type: none"> • Abstraction in programming and software lifecycle • Abstracting data structures: data algebras • Classifications and characteristics of data structures <p>Noteworthy Data Structures</p> <ul style="list-style-type: none"> • List 		



	<ul style="list-style-type: none"> • Stack & Queue • Set & Dictionary • Priority Queue • Binary & N-ary Tree • Graph <p>Algorithmics</p> <ul style="list-style-type: none"> • Basics of computational complexity analysis • Algorithmic techniques • Selective techniques: enumerative & backtracking • Generative techniques: greedy & divide-et-impera <p>Exercises on structures and algorithms</p> <ul style="list-style-type: none"> • Object-oriented programming and C++ Language • Application to an adventure game 		
Books of reference	<p>Adopted:</p> <ul style="list-style-type: none"> • S. Ferilli "Astrazione Dati - con applicazioni ai Giochi d'Avventura". Boopen, 2016 • M. Cadoli, M. Lenzerini, P. Naggar, A. Schaerf. "Fondamenti della progettazione dei programmi". Città studi Edizioni, 1997. <p>Students who wish to, can obtain texts on loan from the Library. It could be convenient to check their availability through the University Library System https://opac.uniba.it/easyweb/w8018/index.php? and contact the library to arrange the loan.</p>		
Notes to the books	<p>The text by Ferilli is adopted in its entirety for the parts on Abstract Data Structures and Adventure Games Exercises.</p> <p>The text by Cadoli et al. Is adopted for the parts on "Introduction and Basics" and "Algorithmics" and "C++ Language."</p>		
Organization of the didactic activities			
Hours			
Total	Lectures	Practice sessions	Individual study
225 hours	56 hours	30 hours	139 hours
CFU/ETCS			
9 CFU	7 CFU	2 CFU	
Teaching methods			
Lectures supported by slides, classroom exercises, exercise assignments with collective correction and self-assessment.			
Expected learning outcomes			
Knowledge and understanding	<ul style="list-style-type: none"> • Categorization of problems based on search space. Knowledge of selective and generative algorithmic paradigms, of related algorithmic techniques and most reknown algorithms. Awareness of capabilities and limits of information technologies based on object-oriented programming, independently of a specific programming environment. • Fundamental aspects of the discipline that stay unchanged against technological change, so as to have a cultural reference system that goes beyond time and situations. 		
Applying knowledge and understanding	<ul style="list-style-type: none"> • Effective design of algorithms using the data abstraction methodology. Ability to design and evaluate alternative implementations based on 		



	<p>computational efficiency. Basic concepts of complexity theory and ability to apply them in simple basic algorithms.</p> <ul style="list-style-type: none">• Ability to understand the limits of imperative programming and to take the opportunities offered by object-oriented programming.
Other skills	<ul style="list-style-type: none">• Making informed judgments and choices<ul style="list-style-type: none">◦ Ability to evaluate the quality of an algorithmic solution in terms of efficiency and capability for reuse. Ability to critically evaluate the learned concepts, defining own viewpoint and defending it in discussions.• Communicating knowledge and understanding<ul style="list-style-type: none">◦ Use of suitable technical terminology and ability to communicate on problems concerning algorithmic studies, also to non-expert people.• Capacities to continue learning<ul style="list-style-type: none">◦ Ability to adapt the acquired knowledge to new contexts, and to keep up-to-date by consulting specialistic sources in the algorithmic field. Ability to find and consult bibliographic materials, to synthesize their contents, also in English, and to use them when preparing the exam.
Assessment	
Assessment methods	<p>- type of evaluation used: grade in 30ths</p> <p>- mode of communication of test results: direct communication in case of non-approval of the case study</p> <p>The examination consists of two tests:</p> <ol style="list-style-type: none">1. practical test (suitability with indicative grade)2. oral test (grade in 30ths) <p>To qualify for the oral examination, it is necessary to have successfully completed the practical test.</p> <p>The practical test may be carried out on the student's personal laptop computer and consists of the development of a working program that solves a given problem making use of the abstract data structures and algorithmic techniques studied in the course.</p> <p>The laboratory test may be replaced by a case study at the student's request. The topic of the case study will be assigned by the instructor and, unless otherwise agreed, will involve modifying or adding features in an adventure game provided by the instructor, making use of the abstract data structures and algorithmic techniques studied in the course.</p> <p>The case study is intended for individual students. Interested students should apply for the assignment a month or so before the chosen exam date, and hand in the entire developed project no later than the eighth day before the exam date in the form of a link to download a ZIP file containing all project materials (compilable and executable CodeBLOCKS project, interchangeable realizations of the data structures, documentation in ODT and PDF formats).</p> <p>The final oral exam usually consists of three questions that may cover either theoretical topics or solutions to problems similar to those covered during the course.</p> <p>The projects are valid for the exam date for which they were requested.</p>
Evaluation criteria	<ul style="list-style-type: none">• Knowledge and understanding<ul style="list-style-type: none">◦ Clarity and completeness in answers to oral questions◦ Correctness and insight in answers to oral questions• Applying knowledge and understanding<ul style="list-style-type: none">◦ Laboratory test or Case Study.



	<ul style="list-style-type: none">◦ Clarity and completeness of the documentation of the practical test• Autonomy of judgment<ul style="list-style-type: none">◦ Design decisions taken in the case study◦ Problem solving during oral test• Communication skills<ul style="list-style-type: none">◦ Ability to use appropriately technical terminology◦ Clarity, compactness and balance of answers to oral questions• Capacities to continue learning<ul style="list-style-type: none">◦ Ability to identify the proper background useful to approach a new topic◦ Ability to carry out suitable literature research
Measurements and final grade	<p>Intended learning outcomes will be assessed based on:</p> <ul style="list-style-type: none">- Quality, completeness and breadth of the case study (including documentation).- Clarity, correctness and completeness of responses to oral questions.- Ability to provide insights or advice on specific issues based on the acquired knowledge. <p>The case study is evaluated in terms of its suitability for accessing the oral test, with a category-based rating (insufficient, sufficient, good, excellent). The oral test is graded in 30ths.</p>
Further information	<p>Students are advised to rely exclusively on the information/communications provided on the official websites of the Computer Science Department, or on social groups only if set up and administered exclusively by the teachers of the related courses:</p> <ul style="list-style-type: none">• https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea• https://www.uniba.it/it/ricerca/dipartimenti/informatica• https://elearning.di.uniba.it/ <p>Course schedules are available here:</p> <ul style="list-style-type: none">• https://programmi.di.uniba.it/ <p>The information that all students should know is written in the Teaching regulations and study posters available on the site:</p> <ul style="list-style-type: none">• https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea <p>Students are advised to be wary of information circulating on unofficial sites or social groups, as they are often found to be unreliable, incorrect or incomplete. For any doubt, a meeting with the teacher can be set up with the same procedure as for working hours.</p> <hr/> <p><i>Teams: 130cked</i></p>