



Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	Laboratorio di Informatica	
Corso di studio	Informatica e Comunicazione Digitale	
Anno accademico	2024/25	
Crediti formativi universitari (CFU) / European credit transfer and accumulation system (ECTS)	6 CFU	
Settore scientifico disciplinare	ING-INF/05	
Lingua di erogazione	Italiano	
Anno di corso	Primo	
Periodo di erogazione	Secondo semestre	
Obbligo di frequenza	La frequenza è fortemente raccomandata	
Sito web del corso di studio	https://www.uniba.it/it/corsi/cdl-informatica-comunicazione-digitale-taranto	

Docente	
Nome e cognome	Gennaro Vessio
Indirizzo e-mail	gennaro.vessio@uniba.it
Sede	Dipartimento di Informatica, Via Orabona 4, 70125 - Bari, stanza n. 673 (6° piano)
Sede virtuale	https://elearning.uniba.it/
Sito web del docente	https://www.gennarovessio.com/
Ricevimento	Su appuntamento

Syllabus	
Obiettivi formativi	L'insegnamento di Laboratorio di Informatica mira a formare i discenti sui principali aspetti pragmatici della programmazione, in contrapposizione a quelli più teorici.
Prerequisiti	Nessuno



Contenuti di insegnamento (Programma)	<ol style="list-style-type: none"> 1. Il linguaggio C 2. Funzioni, array, stringhe 3. Puntatori 4. Strutture e gestione dinamica della memoria 5. Strutture dati dinamiche 6. File 7. Progettazione modulare 8. Stile di programmazione 9. Documentazione del codice 10. Testing e debugging 11. Algoritmi fondamentali
Testi di riferimento	<p>Il corso non adotta un testo di riferimento; tuttavia, i contenuti didattici possono essere integrati dal seguente testo suggerito:</p> <p style="text-align: center;">H. M. Deitel, P. J. Deitel, “Il linguaggio C. Fondamenti e tecniche di programmazione”, Pearson, 2022</p> <p>Gli studenti che lo desiderano possono prendere in prestito il testo dalla biblioteca. Può essere conveniente 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	I contenuti didattici saranno trasferiti principalmente attraverso: dispense del docente; risoluzioni di esercitazioni; esempi di casi di studio del passato.

Organizzazione della didattica			
Ore			
Totali	Didattica frontale	Pratica (esercitazioni, caso di studio)	Studio individuale
150 ore	24 ore	45 ore	81 ore
CFU/ETCS			
6 CFU	3 CFU	3 CFU	

Metodi didattici	Lezioni frontali, esercitazioni in linguaggio C, caso di studio in coppia.
-------------------------	----------------------------------------------------------------------------

Risultati di apprendimento previsti	
Conoscenza e capacità di comprensione	Al termine dell’insegnamento, il discente conoscerà alcuni dei principali aspetti pragmatici della programmazione, quali la programmazione difensiva, un corretto



	stile di scrittura, l'uso appropriato della nomenclatura, testing e debugging, progettazione modulare, ecc.
Conoscenza e capacità di comprensione applicate	Al termine dell'insegnamento, posto di fronte a un problema reale, il discente sarà in grado di implementare una soluzione non solo corretta, ma anche chiara, ben progettata e ben documentata. Inoltre, avrà ampliato le proprie capacità di problem solving.
Competenze trasversali	Al termine dell'insegnamento, il discente avrà sviluppato la capacità di condurre un caso di studio reale dall'analisi dei requisiti, all'implementazione di una soluzione algoritmica e al suo testing e debugging. Il discente apprenderà inoltre a redigere una documentazione di progetto e a comunicarla con efficacia a interlocutori specialisti e non, enfatizzandone i punti chiave.

Valutazione	
Modalità di verifica dell'apprendimento	L'esame prevede un caso di studio – pensato per essere svolto in coppia – su un problema reale complesso scelto da un insieme di possibili tracce. Il caso di studio, discusso in sede di orale, e corredato da una documentazione di progetto e dal codice e i dati a esso associati, dovrà dimostrare la capacità di produrre codice corretto, leggibile, ben documentato, e correttamente progettato e strutturato. Poiché l'insegnamento è di natura fortemente pratica, non prevede esoneri; tuttavia, i casi di studio potranno iniziare a essere svolti già a metà del programma per consentire agli studenti di sostenere l'esame già al primo appello. Il caso di studio svolto viene mantenuto per tutto l'anno accademico.
Criteri di valutazione	Ci si aspetta il discente abbia appreso a: scrivere programmi che garantiscano un'adeguata correttezza e aderenza ai requisiti di progetto; applicare metodologie di progettazione modulare per ottimizzare la struttura del codice sorgente; scrivere correttamente una documentazione di progetto; scrivere codice sorgente che risponda ai dettami del corretto stile di programmazione, garantendo leggibilità e semplicità; applicare correttamente metodologie di testing del codice sorgente e strumenti di debugging; esporre efficacemente l'elaborato.
Criteri di misurazione dell'apprendimento e di attribuzione del voto finale	Il voto finale sarà espresso in trentesimi e l'esame si riterrà superato con un voto conseguito maggiore o uguale a 18. Il maggior contributo al voto finale verrà dalla consegna di un programma funzionante, di una documentazione chiara ed esaustiva, e da codice ben scritto e strutturato. Parte della valutazione riguarderà l'uso di quanto non espressamente richiesto dalla traccia, come i puntatori o la gestione dinamica della memoria. Premialità saranno infine riconosciute a gruppi che avranno arricchito il caso di studio con estensioni non trattate durante il corso.
Altro	<p>Si raccomanda vivamente ai discenti di impegnarsi a svolgere le esercitazioni (da soli o in gruppo) senza attenderne la risoluzione, al fine di sviluppare capacità di analisi e giudizio critico.</p> <p>Si suggerisce altresì agli studenti di affidarsi esclusivamente alle informazioni/comunicazioni fornite sui siti ufficiali del Dipartimento di Informatica, o sui gruppi social costituiti e amministrati esclusivamente dai docenti degli insegnamenti di interesse:</p> <ul style="list-style-type: none">• https://www.uniba.it/it/ricerca/dipartimenti/informatica• https://elearning.uniba.it/ <p>Le informazioni che tutti gli studenti dovrebbero conoscere sono riportate nei regolamenti didattici e nei manifesti degli studi disponibili sul sito:</p>



<https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea>

Si suggerisce agli studenti di diffidare delle informazioni che circolano su siti o gruppi social non ufficiali, poiché sono risultate spesso inaffidabili, errate o incomplete.

Gli studenti saranno invitati a unirsi a un gruppo Telegram che sarà creato all'inizio delle lezioni.



Main course information

Name	Computer Science Laboratory	
Degree	Computer Science and Digital Communication	
Academic year	2024/25	
European credit transfer and accumulation system (ECTS) / Crediti formativi universitari (CFU)	6 CFUs	
Scientific disciplinary area	ING-INF/05	
Language	Italian	
Course year	First	
Delivery period	Second semester	
Required attendance	Attendance is strongly recommended	
Website	https://www.uniba.it/it/corsi/cdl-informatica-comunicazione-digitale-taranto	

Lecturer

Name and surname	Gennaro Vessio
E-mail address	gennaro.vessio@uniba.it
Office	Dipartimento di Informatica, Via Orabona 4, 70125 - Bari, stanza n. 673 (6° piano)
Virtual platform	https://elearning.uniba.it/
Lecturer's website	https://www.gennarovessio.com/
Reception	By appointment only

Syllabus

Learning objectives	The teaching of Computer Science Laboratory aims to train students in the main pragmatic aspects of programming, as opposed to the more theoretical ones.
Prerequisites	None



Program	<ol style="list-style-type: none"> 1. The C language 2. Functions, arrays, strings 3. Pointers 4. Structures and dynamic memory management 5. Dynamic data structures 6. Files 7. Modular design 8. Programming style 9. Code documentation 10. Testing e debugging 11. Fundamental algorithms
Reference books	<p>The course does not adopt a reference text; however, the teaching content can be supplemented by the following suggested text:</p> <p style="text-align: center;">H. M. Deitel, P. J. Deitel, “Il linguaggio C. Fondamenti e tecniche di programmazione”, Pearson, 2022</p> <p>Students who wish to do so may borrow the text from the library. It may be convenient to check its availability through the University Library System (https://opac.uniba.it/easyweb/w8018/index.php) and contact the library to arrange borrowing.</p>
Notes to reference books	Teaching content will be transferred mainly through: lecturer’s handouts; exercise resolutions; examples of past case studies.

Teaching organization			
Hours			
Total	Frontal teaching	Practice (exercises, case study)	Individual study
150 hours	24 hours	45 hours	81 hours
CFU/ETCS			
6 CFUs	3 CFUs	3 CFUs	

Teaching methods	Lectures, C language exercises, paired case study.
-------------------------	----------------------------------------------------

Expected learning outcomes	
Knowledge and understanding skills	Upon completion of the teaching, the student will know some of the main pragmatic aspects of programming, such as defensive programming, proper writing style, appropriate use of nomenclature, testing and debugging, modular design, etc.



Applied knowledge and understanding skills	At the end of the teaching, confronted with a real problem, the student will be able to implement a solution that is not only correct, but also clear, well designed and well documented. In addition, he/she will have expanded his/her problem solving skills.
Other skills	By the end of the teaching, the student will have developed the ability to conduct a real-world case study from requirements analysis to the implementation of an algorithmic solution and its testing and debugging. The student will also learn how to write project documentation and communicate it effectively to specialist and non-specialist stakeholders, emphasizing key points.

Assessment	
Verification modality	The exam involves a case study – designed to be done in pairs – on a complex real-world problem chosen from a set of possible tracks. The case study, discussed at the oral, and accompanied by design documentation and the code and data associated with it, must demonstrate the ability to produce code that is correct, readable, well-documented, and properly designed and structured. Because the teaching is highly practical in nature, there are no waivers; however, case studies may begin to be taken as early as the middle of the program to enable students to take the exam as early as the first call. The case study done is maintained throughout the academic year.
Assessment criteria	The student is expected to have learned to: write programs that ensure adequate correctness and adherence to project requirements; apply modular design methodologies to optimize the structure of source code; correctly write project documentation; write source code that meets the dictates of proper programming style, ensuring readability and simplicity; correctly apply source code testing methodologies and debugging tools; and effectively expose the project.
Criteria and final grade	The final grade will be expressed in thirtieths, and the exam will be considered passed with an achieved grade greater than or equal to 18. The greatest contribution to the final grade will come from the delivery of a working program, clear and comprehensive documentation, and well-written and structured code. Part of the evaluation will involve the use of topics not explicitly required by the track, such as pointers or dynamic memory management. Finally, rewards will be awarded to groups that have enriched the case study with extensions not covered in the course.
Other	<p>It is strongly recommended that students engage in the exercises (alone or in groups) without waiting for their resolution, in order to develop analytical skills and critical judgment.</p> <p>It is also suggested that students rely exclusively on the information/communication provided on the official websites of the Department of Computer Science, or on the social groups established and administered exclusively by the faculty members of the teachings of interest:</p> <ul style="list-style-type: none"> • https://www.uniba.it/it/ricerca/dipartimenti/informatica • https://elearning.uniba.it/ <p>Information that all students should be aware of can be found in the teaching regulations and study manifestos available on the website: https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea</p> <p>Students are suggested to be wary of information circulating on unofficial social sites or groups, as it has often been found to be unreliable, incorrect or incomplete.</p>



UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

DIPARTIMENTO
DI
INFORMATICA

Students will be asked to join a Telegram group that will be created at the beginning of class.