



Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	Laboratorio di informatica (track cognomi M-Z)	
Corso di studio	Informatica	
Anno Accademico	2024/25	
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS)	6 CFU	
Settore Scientifico Disciplinare	INF/01	
Lingua di erogazione	Italiano	
Anno di corso	Primo	
Periodo di erogazione	2° semestre, le date esatte sono riportate nel manifesto/regolamento	
Obbligo di frequenza	No, ma 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	

Docenti	
Nome e cognome	Pierpaolo Basile
Indirizzo mail	pierpaolo.basile@uniba.it
Telefono	0805442301
Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n.758, 7° piano.
Sede virtuale	Piattaforma e-learning UNIBA - https://elearning.uniba.it/
Sito web del docente	https://www.uniba.it/it/docenti/pierpaolo-basile/pierpaolo-basile
Ricevimento (giorni, orari e modalità, es. su appuntamento)	Martedì 11:30-13:30, è preferibile prenotarsi via mail.
Nome e cognome	Marco de Gemmis
Indirizzo mail	marco.degemmis@uniba.it
Telefono	0805443283
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 trasmettere le conoscenze necessarie per progettare, realizzare e testare programmi mediante il paradigma della programmazione imperativa, attraverso una progettazione modulare. Studenti/esse estenderanno le capacità di base già acquisite dall'insegnamento di "Programmazione" per formalizzare, usando il linguaggio di programmazione C, soluzioni a problemi mediamente complessi.
Prerequisiti	Le seguenti conoscenze preliminari facilitano ed accelerano la comprensione degli argomenti dell'insegnamento: • da Programmazione: basi della programmazione imperativa e della programmazione strutturata, ricorsione, processo di compilazione, dati strutturati, problem-solving, tecniche di rappresentazione di una soluzione, • da Architetture degli Elaboratori e SO: tipologia di dati e occupazione della memoria, algebra di Boole e porte logiche, gestione dei processi.
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>Richiamo ai principali concetti di programmazione e dei costrutti in C:</u> • Dal <i>problem solving</i> al programma • Costrutti di base della programmazione strutturata • Cenni sulla memoria e l'occupazione dei dati • Dati strutturati e Strutture Dati: definizione e gestione • Prime esercitazioni: pratica con IDE, soluzioni di problemi Ore lezione frontale: 7 Ore esercitazione in aula: 6 2. <u>Programmazione in C:</u> • Elaborazione avanzata di stringhe, funzioni per la gestione delle stringhe • Puntatori concetto astratto e utilizzo concreto nel linguaggio C • Array di puntatori e aritmetica dei puntatori • Record gestione e manipolazione • File di testo e file binari: gestione e manipolazione • Preprocessore del linguaggio C • Esercitazioni: soluzione di problemi di media complessità Ore lezione frontale: 0 Ore esercitazione in aula: 20 3. <u>Stile di Programmazione:</u>



	• La scelta dei nomi delle variabili e delle funzioni • Indentazione e formato delle espressioni e istruzioni • Uso dei Commenti • Convenzioni di programmazione • Esercitazioni: modificare e migliorare i programmi realizzati Ore lezione frontale: 1 Ore esercitazione in aula: 2 4. <u>Programmazione modulare:</u> • Modularizzazione e strutturazione dei programmi • Astrazione dati e funzionale • Information Hiding • Classi di memoria • Scope degli identificatori • Header file • Esercitazioni: creazione di programmi modulari Ore lezione frontale: 6 Ore esercitazione in aula: 4 5. <u>Testing e Debugging:</u> • Tipi di test • Lo unit test: white box e black box • Strumenti e tecniche di debugging • Test per classi di equivalenza • Test per valori estremi Ore lezione frontale: 2 Ore esercitazione in aula: 2 6. <u>Algoritmi Fondamentali:</u> • Algoritmi di ricerca (esaustiva, con sentinella, binaria) • Algoritmi di ordinamento (bubble sort, selection sort, shell sort, insertion sort, merge sort, quick sort) • Cenni di complessità computazionale Ore lezione frontale: 8 Ore esercitazione in aula: 5 7. <u>Caso di Studio:</u> • Esercitazioni pratiche per la preparazione del caso di studio Ore lezione frontale: 0 Ore esercitazione in aula: 6
Testo di riferimento	[Deitel] P. Deitel e H. Deitel. Il linguaggio C – Fondamenti e tecniche di programmazione 9ª edizione - Pearson 2022 - ISBN: 9788891901651 (ultima edizione disponibile, sono utilizzabili anche le edizioni precedenti dalla 4a) Si 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.			
Note ai testi di riferimento	Si specificano di seguito, per ogni argomento del programma, i capitoli del testo [Deitel] dai quali studiare. Argomento 1: capitoli 1-4 del testo. Argomenti 2-3: capitoli 6-8, 13 del testo. Argomento 4: capitolo 5 del testo. Argomento 5: capitolo 13 del testo. Argomento 6: appendice D del testo Per ogni argomento è inoltre disponibile su piattaforma di e-learning UNIBA https://elearning.uniba.it/ una dispensa, corrispondente ai contenuti mostrati in aula durante le lezioni, che non è comunque sostitutiva dei contenuti del testo di riferimento. Inoltre, sulla piattaforma sono rese disponibili tutte le indicazioni per realizzare il caso di studio oggetto della prova d’esame.			
Organizzazione della didattica				
Ore				
Totali	Didattica frontale	Laboratorio/esercitazione	Progetto	Studio individuale
150 ore	24 ore	45 ore	0 ore	81 ore
CFU/ETCS				
6 CFU	3 CFU	3 CFU	0 CFU	

Metodi didattici	
	- Lezioni frontali condotte con l'ausilio di dispense proiettate in aula e rese disponibili tramite piattaforma di e-learning UNIBA; - Esercitazioni guidate svolte in aula mediante utilizzo del proprio dispositivo; - Esercitazioni da svolgere autonomamente.

Risultati di apprendimento previsti	
--	--



Conoscenza e capacità di comprensione	● Acquisizione conoscenze approfondite del linguaggio di programmazione C. ● Acquisire capacità di progettazione, sviluppo e test di programmi di complessità più elevata di quelli affrontati nel primo approccio alla programmazione imperativa. ● Migliorare le capacità di problem solving.
Conoscenza e capacità di comprensione applicate	● Capacità di tradurre algoritmi in programmi (in C) corretti ed opportunamente documentati. ● Capacità di condurre una verifica empirica della correttezza dei programmi mediante tecniche di testing. ● Capacità di valutare l'efficienza di una soluzione
Competenze trasversali	Autonomia di giudizio ● Capacità di verificare l'aderenza di un programma ai requisiti del problema da risolvere. ● Capacità di confrontare diverse soluzioni al problema in termini di efficienza. Abilità comunicative ● Capacità di illustrare in modo appropriato la soluzione creata, utilizzando un linguaggio tecnico appropriato. Capacità di apprendere in modo autonomo ● Capacità di orientarsi agevolmente nelle problematiche relative alla comprensione e all'utilizzo delle tecnologie e dei metodi per lo sviluppo di programmi, nonché ai diversi linguaggi di programmazione.

Valutazione	
Modalità di verifica dell'apprendimento	La prova d'esame consiste nella discussione orale di un caso di studio in cui si dovranno illustrare in modo appropriato gli algoritmi, gli strumenti e le metodologie informatiche utilizzate per la realizzazione dello stesso. Il caso di studio potrà essere realizzato individualmente oppure in collaborazione tra due studenti/esse. La traccia del caso di studio per la prima sessione d'esame (giugno/luglio) sarà pubblicata nella piattaforma di e-learning UNIBA durante l'erogazione dell'insegnamento, in modo tale che il caso di studio possa essere principalmente realizzato durante la frequenza. Per le altre sessioni, le tracce saranno pubblicate sempre nella piattaforma con congruo anticipo rispetto alla data dell'appello.
Criteri di valutazione	● Conoscenza e capacità di comprensione: ○ dimostrazione di conoscenze approfondite del linguaggio di programmazione C;



	○ dimostrazione di capacità di progettazione e sviluppo di programmi di complessità media; ○ dimostrazione di capacità di problem solving. ● Conoscenza e capacità di comprensione applicate: ○ dimostrazione di capacità di tradurre algoritmi in programmi (in C) corretti ed opportunamente documentati; ○ dimostrazione di capacità di condurre una verifica empirica della correttezza dei programmi mediante tecniche di testing; ○ dimostrazione di capacità di valutare l'efficienza di una soluzione. ● Autonomia di giudizio: ○ dimostrazione di capacità di verificare l'aderenza di un programma ai requisiti del problema da risolvere; ○ dimostrazione di capacità di confrontare diverse soluzioni al problema in termini di efficienza. ● Abilità comunicative: ○ dimostrazione di capacità di illustrare in modo appropriato la soluzione proposta per il caso di studio, utilizzando un linguaggio tecnico appropriato. ● Capacità di apprendere: ○ dimostrazione di capacità di astrazione, di ragionamento per analogia e creatività nelle soluzioni ideate per il caso di studio.
Criteri di misurazione dell'apprendimento e di attribuzione del voto finale	La determinazione del voto finale tiene conto dei seguenti criteri relativi al caso di studio ed alla sua presentazione durante la prova orale: 1) la correttezza della soluzione proposta; 2) la completezza della soluzione proposta; 3) la logica seguita nella definizione degli algoritmi; 4) l'efficienza della soluzione proposta; 5) chiarezza espositiva. Per ogni criterio, sarà attribuito un punteggio tra 0 e 6, in relazione alle risposte date in sede di prova orale. 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.
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>
- <https://www.uniba.it/it/ricerca/dipartimenti/informatica>
- <https://elearning.uniba.it/>

I programmi di tutti gli insegnamenti sono disponibili al seguente link:

- <https://elearning.uniba.it/>

Le informazioni che tutti gli studenti dovrebbero conoscere sono scritte nei regolamenti didattici dei Corsi di Studi disponibili nel sito:

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

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.

Link all'insegnamento nella piattaforma di e-learning:

<https://elearning.uniba.it/course/view.php?id=2841>



Main information on the course

Course name	Computer Science Laboratory (Track surnames M-Z)	
Degree	Computer Science	
Academic year	2024/25	
European Credit Transfer and Accumulation System (ECTS), in Italian Crediti Formativi Universitari (CFU)		6 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	Pierpaolo Basile
email	pierpaolo.basile@uniba.it
phone	0805442301
office	Department of Computer Science, Campus, Via E. Orabona 4, 70126, Bari. office n.758, 7th floor.
e-learning platform	UNIBA e-learning platform - https://elearning.uniba.it/
Teacher's homepage	https://www.uniba.it/it/docenti/pierpaolo-basile/pierpaolo-basile
Office hours	Tuesday 11:30 a.m. to 1:30 p.m. Booking via email is preferred.
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. Professor's Room, Mariscuola Taranto
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 professor to ask for an appointment.

Syllabus

Course goals	The course aims to provide the knowledge necessary to design, implement and test programs using the imperative programming paradigm through modular design. Students will extend the basic skills already acquired from "Programming" course to formalize, using the C programming language, solutions to moderately complex problems.
Prerequisites/requirements	The following prior knowledge facilitates and accelerates understanding of the course topics:



	• from Programming: basics of imperative programming and structured programming, recursion, compilation process, structured data, problem-solving, techniques for representing a solution; • from Computer Architectures and OS: data types and memory occupancy, Boolean algebra and logic gates, process management.
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. 1. <u>Recall of the main programming concepts and constructs in C:</u> • From problem solving to programs • Basic constructs of structured programming • Hints on memory and data occupancy • Structured data and Data Structures: definition and management • First exercises: practice with IDE, problem solving Lecture: 7 Practice Session: 6 2. <u>Programming with C:</u> • Advanced string processing, functions for string handling • Pointers abstract concept and concrete usage in the C language • Pointer arrays and pointer arithmetic • Record management and manipulation • Text and binary files: handling and manipulation • C language preprocessor • Exercises: solving problems of medium complexity Lecture: 0 Practice Session: 20 3. <u>Programming Style:</u> • The choice of variable and function names • Indentation and format of expressions and statements • Use of Comments • Programming conventions • Exercises: modifying and improving the programs made Lecture: 1 Practice Session: 2 4. <u>Modular programming:</u> • Modularization and structuring of programs • Data and functional abstraction • Information Hiding • Memory Classes • Scope of identifiers • Header files • Exercises: creation of modular programs Lecture: 6 Practice Session: 4 5. <u>Testing e Debugging:</u> • Types of testing • The unit test: white box and black box • Debugging tools and techniques • Testing for equivalence classes • Testing for extreme values



	Lecture: 2 Practice Session: 2			
	6. <u>Fundamental Algorithms:</u> • Search algorithms (exhaustive, with sentinel, binary) • Sorting algorithms (bubble sort, selection sort, shell sort, insertion sort, merge sort, quick sort) • Concepts of computational complexity Lecture: 8 Practice Session: 5			
	7. <u>Case Study:</u> ○ Practical exercises for case study preparation Lecture: 0 Practice Session: 6			
Books of reference	1. [Deitel] P. Deitel e H. Deitel. Il linguaggio C – Fondamenti e tecniche di programmazione 9a edizione - Pearson 2022 - ISBN: 9788891901651 (latest edition available, previous editions from 4th are also usable) 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 book from which to study are specified below for each topic in the program. Topic 1: chapters 1-4 of the textbook. Topic 2-3: chapters 6-8, 13 of the textbook. Topic 4: chapter 5 of the textbook. Topic 5: chapter 13 of the textbook. Topic 6: appendix D of the textbook. On the e-learning platform 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 book. Furthermore, all the directions for preparing the case study that is the subject of the exam are made available on the platform.			
Organization of the didactic activities				
Hours				
Total	Lectures	Practice sessions	Project work	Individual study
150	24	45	0	81
CFU/ETCS				
6	3	3	0	
Teaching methods				
	• Classes given with the help of slides made available through the e-learning platform; • Guided practical session during class hours. Students are required to bring their own device;			



- Homework.

Expected learning outcomes	
Knowledge and understanding	• Acquire in-depth knowledge of the C programming language. • Acquire skills in designing, developing and testing programs of higher complexity than those addressed in the first approach to imperative programming. • Improve problem solving skills.
Applying knowledge and understanding	• Ability to translate algorithms into correct and properly documented programs (in C). • Ability to conduct empirical verification of the correctness of programs using testing techniques. • Ability to evaluate the efficiency of a solution.
Other skills	Autonomy of judgment • Ability to verify the adherence of a program to the requirements of the problem to be solved. • Ability to compare different solutions to the problem in terms of efficiency. Communication Skills • Ability to appropriately illustrate the solution created, using appropriate technical language. Ability to learn independently • Ability to easily navigate issues related to understanding and using program development technologies and methods, as well as different programming languages.

Assessment	
Assessment methods	The exam consists of a discussion of a case study in which the algorithms, tools and methodologies used in its realization must be appropriately illustrated. The case study may be done individually or in collaboration between two students. The outline of the case study for the first exam session (June/July) will be posted on the UNIBA e-learning platform during the course, so that the case study can be mainly carried out during attendance. For other sessions, the outline of the case study will always be published in the platform well in advance of the official date of the exam.
Evaluation criteria	• Knowledge and understanding skills: ◦ demonstration of in-depth knowledge of the C programming language;



	○ demonstration of ability to design and develop programs of medium complexity; ○ demonstration of problem solving skills. ● Applied knowledge and understanding skills: ○ demonstration of ability to translate algorithms into correct and properly documented programs (in C); ○ demonstration of ability to conduct empirical verification of program correctness using testing techniques; ○ demonstration of ability to evaluate the efficiency of a solution. ● Autonomy of judgment: ○ demonstration of ability to verify the adherence of a program to the requirements of the problem to be solved; ○ demonstration of ability to compare different solutions to the problem in terms of efficiency. ● Communication skills: ○ demonstration of ability to appropriately illustrate the proposed solution to the case study, using appropriate technical language. ● Learning skills: ○ demonstration of ability to abstract, reason by analogy and creativity in the solutions devised for the case study.
Measurements and final grade	The determination of the final grade considers the following criteria related to the case study and its presentation during the oral examination: 1) the correctness of the proposed solution; 2) the completeness of the proposed solution; 3) the logic followed in the definition of the algorithms; 4) the efficiency of the proposed solution; 5) clarity of presentation. For each criterion, a score between 0 and 6 will be given, depending on the answers given in the oral test. 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.
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 ● https://www.uniba.it/it/ricerca/dipartimenti/informatica ● https://elearning.uniba.it/ The programs of the courses are available at the following URL: ● https://elearning.uniba.it/ 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/corsi/informatica/studiare/piano-di-studi/regolamento-didattico



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.

Link to the course on the e-learning platform:

<https://elearning.uniba.it/course/view.php?id=2841>