



## Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	<b>Programmazione ed elementi di architettura degli elaboratori (Computer programming and architectural elements)</b>	
Corso di studio	LM Bioinformatica	
Anno Accademico	2023/24	
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS)	9 CFU	
Settore Scientifico Disciplinare	ING-INF/05	
Lingua di erogazione	Italiano	
Anno di corso	Primo	
Periodo di erogazione	1 <sup>^</sup> semestre, le date esatte sono indicate annualmente nel manifesto/regolamento	
Obbligo di frequenza	La frequenza è fortemente raccomandata	
Sito web del corso di studio	<a href="https://www.uniba.it/it/corsi/bioinformatica">https://www.uniba.it/it/corsi/bioinformatica</a>	

## Docente/i

Nome e cognome	Giovanni Dimauro
Indirizzo mail	<a href="mailto:giovanni.dimauro@uniba.it">giovanni.dimauro@uniba.it</a>
Telefono	080-5443294
Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n.617, 6 <sup>^</sup> piano.
Sede virtuale	Piattaforma ADA - <a href="https://elearning.di.uniba.it/">https://elearning.di.uniba.it/</a>
Sito web del docente	TEAMS codice 7s0qsql
Ricevimento (giorni, orari e modalità, es. su appuntamento)	Martedì 11:00 – 13:00, ma anche previo appuntamento per email

## Syllabus

<b>Obiettivi formativi</b>	Lo studente dovrà essere in grado di analizzare e risolvere semplici problemi, progettando e sviluppando programmi nel linguaggio C.
----------------------------	--




<b>Prerequisiti</b>	Buona comprensione della lingua inglese. Lettura individuale da parte dello studente del 1 <sup>^</sup> capitolo del testo di riferimento (v. sotto): “Introduzione ai computer, a Internet e al web”		
<b>Contenuti di insegnamento (Programma)</b>			
	<b>Mod</b>	<b>Argomenti</b>	<b>Ore</b>
	1	Presentazione del corso, contenuti, modalità d'esame, esoneri, frequenza alle lezioni, orari, modalità di esercitazione in aula, ecc. Introduzione alla programmazione Un semplice programma C: visualizzare una riga di testo. Un altro semplice programma C: sommare due interi.	3
	2	Uso del compilatore ambiente di sviluppo Xcode cenni al debugging esercitazioni e approfondimenti degli argomenti: Un semplice programma C: visualizzare una riga di testo.	3
	3	Nozioni sulla memoria L'aritmetica del C Prendere delle decisioni: gli operatori di uguaglianza e relazionali. Gli algoritmi Lo pseudocodice. Le strutture di controllo Il comando di selezione if	3
	4	Il comando di selezione if else. Il comando di iterazione while. Formulazione degli algoritmi: studio di un caso: l'iterazione controllata da un contatore. Formulazione degli algoritmi con processo top down per raffinamenti successivi: studio di un caso: iterazione controllata da un valore sentinella. Conversione di tipo, precisione.	4
	5	Strutture di controllo nidificate esercitazione con uso di pseudocodice Formulazione degli algoritmi con processo top down per raffinamenti successivi: studio di un caso: strutture di controllo nidificate	3
	6	Gli operatori di incremento e di decremento Gli elementi dell'iterazione Iterazione controllata da un contatore il comando di iterazione FOR esempi di utilizzo del comando FOR Il comando di selezione multipla switch introduzione al comando break, studio di un caso. Il comando di iterazione do...while Le istruzioni break e continue	4
	7	Gli operatori logici Operatori di uguaglianza (==) e di assegnamento (=); Riassunto della programmazione strutturata. I moduli di programma in C	3
8	Le funzioni della libreria matematica Le funzioni Le definizioni di funzione I prototipi di funzione Lo stack delle chiamate di funzione e i record di attivazione I file di intestazione Invocare le funzioni: chiamata per valore e per riferimento Generazione di numeri casuali	3	



9	Esercitazione/autovalutazione: sviluppo flowchart, attività svolta con assistenza del docente	4
10	Esempio: un gioco d'azzardo Le classi di memoria Le regole di visibilità	4
11	I vettori La dichiarazione dei vettori Esempi sui vettori	3
12	Esercitazione e studio individuale di materiale video prodotto dal Docente, a supporto delle lezioni teoriche. Il video descrive lo sviluppo di un flowchart prendendo spunto dall'esercizio 4.40 'Crescita della popolazione mondiale' del testo di riferimento. Il metodo di progettazione rispetta i principi della programmazione strutturata e utilizza strutture di controllo di base del linguaggio di programmazione C - discussione con il docente.	4
13	Ulteriori esempi sui vettori	3
14	Vettori statici ed automatici Passare i vettori alle funzioni, passaggio per riferimento (indirizzo) e per valore, qualificatore 'const' Algoritmi di ordinamento: bubble sort	4
15	Studio di un caso: calcolare la media, mediana e la moda usando i vettori La ricerca nei vettori (lineare), algoritmo e programma per la ricerca lineare in un vettore	4
16	La ricerca nei vettori (binaria) Vettori multidimensionali.	4
17	Esercitazione e studio individuale di materiale video prodotto dal Docente, a supporto delle lezioni teoriche. Il video descrive lo sviluppo di un flowchart dell'esercizio 'dado a 20 facce'. Il metodo di progettazione rispetta i principi della programmazione strutturata - discussione con il docente.	3
18	Esercitazione e studio individuale di materiale video prodotto dal docente, a supporto delle lezioni teoriche. Il video descrive il passaggio da un algoritmo descritto con flowchart al codice in Linguaggio C, prendendo spunto dall'esercizio 'dado a 20 facce'. Il metodo di progettazione rispetta i principi della programmazione strutturata e utilizza strutture di controllo di base del linguaggio di programmazione C - discussione con il docente.	3
19	Esercitazione e studio individuale di materiale video prodotto dal Docente, a supporto delle lezioni teoriche. Il video descrive lo sviluppo di un flowchart prendendo spunto dall'esercizio 'Craps - un gioco d'azzardo' del testo di riferimento. Il metodo di progettazione rispetta i principi della programmazione strutturata - discussione con il docente.	3
20	preparazione all'esonero.	4
21	simulazione autonoma in preparazione del 1 <sup>^</sup> esonero, assistita dal docente	3
22	Auto correzione esonero assistita dal docente	4
23	La gerarchia dei dati I file e gli stream Creare un file ad accesso sequenziale Leggere i dati da un file ad accesso sequenziale	4
24	Leggere i dati da un file ad accesso sequenziale (approfondimenti) Programma per l'interrogazione del credito.	3
25	I file ad accesso casuale Creare un file ad accesso casuale	4



	<table border="1"> <tr> <td>26</td> <td>Scrivere i dati in modo casuale in un file ad accesso casuale Leggere i dati in modo casuale da un file ad accesso casuale</td> <td>3</td> </tr> <tr> <td>27</td> <td>Esercitazione e Simulazione 2^ esonero</td> <td>3</td> </tr> <tr> <td></td> <td></td> <td></td> </tr> </table>			26	Scrivere i dati in modo casuale in un file ad accesso casuale Leggere i dati in modo casuale da un file ad accesso casuale	3	27	Esercitazione e Simulazione 2^ esonero	3			
26	Scrivere i dati in modo casuale in un file ad accesso casuale Leggere i dati in modo casuale da un file ad accesso casuale	3										
27	Esercitazione e Simulazione 2^ esonero	3										
<p><b>Testi di riferimento</b></p> 	<p>Testo da cui studiare:</p> <p>P. Deitel e H. Deitel Il linguaggio C – Fondamenti e tecniche di programmazione 8^edizione Pearson 2016 ISBN: 9788891901651 (vanno bene anche le edizioni successive e precedenti dalla 4^ in poi)</p> <p>Testo integrativo, facoltativo:</p> <p>Kim N. King, Programmazione in C, Maggioli Editore, ISBN 8838785821</p> <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 <a href="https://opac.uniba.it/easyweb/w8018/index.php?">https://opac.uniba.it/easyweb/w8018/index.php?</a> e contattare la biblioteca per concordare il prestito.</p>											
<p><b>Note ai testi di riferimento</b></p>	<p>Nel corso delle lezioni il docente utilizzerà delle slide che ripercorrono i contenuti del libro, pertanto non verranno fornite. Il testo di riferimento contiene tutti gli argomenti del corso, pertanto si consiglia di studiare dal testo e di svolgere in autonomia e costantemente tutti gli esercizi inseriti alla fine di ogni capitolo trattato a lezione.</p> <p>Sulla piattaforma ADA del dipartimento (v. sopra 'sede virtuale') sono disponibili:</p> <ul style="list-style-type: none"> <li>• materiale video di supporto utilizzato a lezione;</li> <li>• alcune tracce di prove scritte di esami, con esempi di tracce svolte;</li> </ul>											
<p><b>Organizzazione della didattica</b></p>												
<p><b>Ore</b></p> <table border="1"> <thead> <tr> <th>Totali</th> <th>Didattica frontale</th> <th>Laboratorio ed esercitazioni</th> <th>Studio individuale</th> </tr> </thead> <tbody> <tr> <td>225 ore</td> <td>48 ore</td> <td>45 ore</td> <td>132 ore</td> </tr> </tbody> </table>				Totali	Didattica frontale	Laboratorio ed esercitazioni	Studio individuale	225 ore	48 ore	45 ore	132 ore	
Totali	Didattica frontale	Laboratorio ed esercitazioni	Studio individuale									
225 ore	48 ore	45 ore	132 ore									
<p><b>CFU/ETCS</b></p> <table border="1"> <tbody> <tr> <td>9 CFU</td> <td>7 CFU</td> <td>2 CFU</td> <td></td> </tr> </tbody> </table>				9 CFU	7 CFU	2 CFU						
9 CFU	7 CFU	2 CFU										



<b>Metodi didattici</b>	
	Lezioni frontali, esercitazioni ed attività autonome e di gruppo in aula e a casa (come dettagliato nel programma). Gli studenti non frequentanti possono lavorare singolarmente prendendo accordi con il docente.

<b>Risultati di apprendimento previsti</b>	
<b>Conoscenza e capacità di comprensione</b>	<ul style="list-style-type: none"><li>• Acquisire conoscenze che consentano allo studente di comprendere come si può indicare ad un elaboratore elettronico (macchina automatica di impiego universale, hardware) la soluzione di un problema o di una classe di problemi, che l'elaboratore può risolvere, con un metodo ed un linguaggio appropriato, creando un apposito programma (software) eseguibile dall'elaboratore.</li><li>• Acquisire la capacità di ragionare ed individuare una soluzione ad un problema (algoritmo) secondo il paradigma della programmazione imperativa strutturata;</li></ul>
<b>Conoscenza e capacità di comprensione applicate</b>	<ul style="list-style-type: none"><li>• Comprendere l'uso di un linguaggio di progettazione non convenzionale (es. pseudocodice) e l'uso di una rappresentazione grafica (es. flow chart) per descrivere con un formalismo semplice un algoritmo;</li><li>• Comprendere il lessico, la sintassi e la semantica del linguaggio di programmazione C;</li><li>• Acquisire la capacità di scrivere un programma strutturato in linguaggio C;</li><li>• Acquisire la capacità di individuare casi di test per il dominio cui fa riferimento il programma creato;</li><li>• Acquisire la capacità di utilizzare un ambiente di sviluppo (es. Xcode, Visual Studio) per trasformare il programma sorgente (in C) in programma eseguibile ed eseguirlo.</li></ul>
<b>Competenze trasversali</b>	<p><b>Autonomia di giudizio</b></p> <ul style="list-style-type: none"><li>• Acquisire la capacità di verificare che l'algoritmo individuato risponda alle specifiche di un problema;</li><li>• Acquisire la capacità di verificare che i risultati ottenuti dopo l'esecuzione del programma siano quelli attesi.</li></ul> <p><b>Abilità comunicative</b></p> <ul style="list-style-type: none"><li>• Imparare a commentare il codice prodotto al fine di renderlo comprensibile e agevolmente modificabile da altri professionisti, con l'obiettivo di sviluppare in team.</li></ul> <p><b>Capacità di apprendere in modo autonomo</b></p> <ul style="list-style-type: none"><li>• Capacità di approfondire concetti attraverso lo studio autonomo di materiale video prodotto e proposto dal docente;</li><li>• Capacità di completare autonomamente il percorso formativo previsto dal testo di riferimento, oltre i contenuti previsti dal programma dell'insegnamento.</li></ul>



--	--

Valutazione	
<b>Modalità di verifica dell'apprendimento</b>	<p>Una prima prova di valutazione intermedia, con valore esonerante, si tiene in prossimità della settimana di interruzione delle lezioni, normalmente collocata intorno alla metà di novembre. La prova consiste nella soluzione di un problema individuando l'algoritmo e sviluppando il relativo programma in C, analogamente a quanto spiegato nel corso delle lezioni. Il voto sarà espresso in trentesimi.</p> <p>Una seconda prova di valutazione intermedia, con valore esonerante, si tiene immediatamente prima o dopo le vacanze natalizie, in base all'andamento del corso. La seconda prova consiste in un test a risposta multipla chiusa, in particolare saranno somministrati circa 50 domande ed il tempo previsto è di norma 30 minuti. Ogni risposta errata verrà penalizzata con 0,25/50esimi di punto. Il risultato ottenuto viene normalizzato a 30.</p> <p>Il voto risultante dalla media dei voti ottenuti nelle due prove di esonero viene proposto dal docente per essere verbalizzato. I voti dell'esonero possono essere utilizzati esclusivamente nella prima sessione di esami (gennaio/febbraio). Lo studente può, se preferisce, integrare l'esame con una prova orale. La votazione è in trentesimi.</p> <p>Gli studenti che scelgono di non partecipare alle prove di valutazione intermedie, ovvero non le superano, sostengono l'esame a partire da gennaio. La prova di teoria, se sostenuta e superata nella valutazione intermedia, può essere utilizzata dallo studente come prova di teoria dell'esame regolare.</p> <p>La modalità della prova d'esame regolare è analoga a quella descritta sopra per le due prove intermedie (prova scritta e test).</p> <p>Materiali permessi per sostenere la prima prova di valutazione intermedia e la prova scritta d'esame: testo di riferimento in formato esclusivamente cartaceo.</p> <p>I risultati di tutte le prove vengono comunicati con lista pubblica (numero di matricola e voto conseguito) per email o social. Il voto finale conseguito viene proposto esclusivamente sulla piattaforma Esse3.</p> <p>Incentivi alla frequenza: l'eventuale lode viene più frequentemente attribuita agli studenti che per la stragrande maggioranza delle lezioni hanno frequentato, interagito nel corso della lezione, proposto soluzioni e risolto i casi proposti dal docente a lezione.</p>
Criteri di valutazione	<ul style="list-style-type: none"><li>● <b>Conoscenza e capacità di comprensione:</b><ul style="list-style-type: none"><li>○ Lo studente dovrà essere in grado di analizzare e risolvere semplici problemi e di generalizzare soluzioni per una classe di problemi con lo stile della programmazione strutturata.</li></ul></li><li>● <b>Conoscenza e capacità di comprensione applicate:</b><ul style="list-style-type: none"><li>○ Lo studente dovrà essere in grado di codificare le soluzioni ideate descrivendole in pseudocodice, in flowchart strutturati e nel linguaggio di programmazione C;</li><li>○ Lo studente dovrà essere in grado di utilizzare un ambiente di sviluppo a sua scelta e dimostrare di conoscere il linguaggio C;</li></ul></li></ul>



	<ul style="list-style-type: none"> <li>● <b>Autonomia di giudizio:</b> <ul style="list-style-type: none"> <li>○ Lo studente dovrà essere in grado di correggere e validare il corretto funzionamento dei programmi sviluppati.</li> </ul> </li> <li>● <b>Abilità comunicative:</b> <ul style="list-style-type: none"> <li>○ Lo studente dovrà essere in grado di rendere il codice scritto comprensibile ad altri, mediante la sua descrizione generale e commenti specifici alle istruzioni e alle strutture di controllo utilizzate.</li> </ul> </li> <li>● <b>Capacità di apprendere:</b> <ul style="list-style-type: none"> <li>○ Lo studente dovrà essere in grado di trasformare autonomamente algoritmi descritti con flowchart in programmi in linguaggio C;</li> <li>○ Lo studente dovrà essere in grado di utilizzare le soluzioni alternative descritte nel testo di riferimento, se non descritte nel corso delle lezioni, come ad esempio le diverse modalità di dichiarazione delle variabili.</li> </ul> </li> </ul>																
<p>Criteria di misurazione dell'apprendimento e di attribuzione del voto finale</p>	<table border="1"> <thead> <tr> <th>Voto</th> <th>Descrittori</th> </tr> </thead> <tbody> <tr> <td>&lt; 18 insufficiente</td> <td>Conoscenze frammentarie e superficiali dei contenuti, errori nell'applicare i concetti, descrizione carente.</td> </tr> <tr> <td>18 - 20</td> <td>Conoscenze dei contenuti sufficienti ma generali, descrizione semplice, incertezze nell'applicazione di concetti teorici.</td> </tr> <tr> <td>21 - 23</td> <td>Conoscenze dei contenuti appropriate ma non approfondite, capacità di applicare i concetti teorici, capacità di presentare i contenuti in modo semplice.</td> </tr> <tr> <td>24 - 25</td> <td>Conoscenze dei contenuti appropriate ed ampie, discreta capacità di applicazione delle conoscenze, capacità di presentare i contenuti in modo articolato.</td> </tr> <tr> <td>26 - 27</td> <td>Conoscenze dei contenuti precise e complete, buona capacità di applicare le conoscenze, capacità di analisi, descrizione chiara e corretta.</td> </tr> <tr> <td>28 - 29</td> <td>Conoscenze dei contenuti ampie, complete ed approfondite, buona applicazione dei contenuti, buona capacità di analisi e di sintesi, descrizione sicura e corretta.</td> </tr> <tr> <td>30 30 e lode</td> <td>Conoscenze dei contenuti molto ampie, complete ed approfondite, capacità ben consolidata di applicare i contenuti, ottima capacità di analisi, di sintesi e di collegamenti interdisciplinari, padronanza di descrizione.</td> </tr> </tbody> </table>	Voto	Descrittori	< 18 insufficiente	Conoscenze frammentarie e superficiali dei contenuti, errori nell'applicare i concetti, descrizione carente.	18 - 20	Conoscenze dei contenuti sufficienti ma generali, descrizione semplice, incertezze nell'applicazione di concetti teorici.	21 - 23	Conoscenze dei contenuti appropriate ma non approfondite, capacità di applicare i concetti teorici, capacità di presentare i contenuti in modo semplice.	24 - 25	Conoscenze dei contenuti appropriate ed ampie, discreta capacità di applicazione delle conoscenze, capacità di presentare i contenuti in modo articolato.	26 - 27	Conoscenze dei contenuti precise e complete, buona capacità di applicare le conoscenze, capacità di analisi, descrizione chiara e corretta.	28 - 29	Conoscenze dei contenuti ampie, complete ed approfondite, buona applicazione dei contenuti, buona capacità di analisi e di sintesi, descrizione sicura e corretta.	30 30 e lode	Conoscenze dei contenuti molto ampie, complete ed approfondite, capacità ben consolidata di applicare i contenuti, ottima capacità di analisi, di sintesi e di collegamenti interdisciplinari, padronanza di descrizione.
Voto	Descrittori																
< 18 insufficiente	Conoscenze frammentarie e superficiali dei contenuti, errori nell'applicare i concetti, descrizione carente.																
18 - 20	Conoscenze dei contenuti sufficienti ma generali, descrizione semplice, incertezze nell'applicazione di concetti teorici.																
21 - 23	Conoscenze dei contenuti appropriate ma non approfondite, capacità di applicare i concetti teorici, capacità di presentare i contenuti in modo semplice.																
24 - 25	Conoscenze dei contenuti appropriate ed ampie, discreta capacità di applicazione delle conoscenze, capacità di presentare i contenuti in modo articolato.																
26 - 27	Conoscenze dei contenuti precise e complete, buona capacità di applicare le conoscenze, capacità di analisi, descrizione chiara e corretta.																
28 - 29	Conoscenze dei contenuti ampie, complete ed approfondite, buona applicazione dei contenuti, buona capacità di analisi e di sintesi, descrizione sicura e corretta.																
30 30 e lode	Conoscenze dei contenuti molto ampie, complete ed approfondite, capacità ben consolidata di applicare i contenuti, ottima capacità di analisi, di sintesi e di collegamenti interdisciplinari, padronanza di descrizione.																
<p><b>Altro</b></p>	<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"> <li>● <a href="https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea">https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea</a></li> <li>● <a href="https://www.uniba.it/it/ricerca/dipartimenti/informatica">https://www.uniba.it/it/ricerca/dipartimenti/informatica</a></li> <li>● <a href="https://elearning.di.uniba.it/">https://elearning.di.uniba.it/</a></li> </ul> <p>I programmi degli insegnamenti sono disponibili qui:</p>																



- <https://programmi.di.uniba.it/>

Le informazioni che tutti gli studenti dovrebbero conoscere sono scritte nei Regolamenti didattici e manifesti degli studi disponibili nel sito:

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

*Si suggerisce agli studenti di diffidare dalle informazioni circolanti su siti o gruppi social non ufficiali, poichè spesso sono risultate non affidabili, non corrette o incomplete.*

Gli studenti potranno unirsi al forum del corso A.A. 2022/23 su Telegram, utilizzato per scopi didattici, al quale aderisce anche il docente:

<https://t.me/+V1eWiJ67GTBjNzNk>

Link al corso sulla piattaforma e-learning del dipartimento ADA:

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





## General information

Course name	<b>Programmazione ed elementi di architettura degli elaboratori (Computer programming and architectural elements)</b>	
Year of the course	LM Bioinformatica	
Academic year	2023/24	
Credits (CFU/ECTS)		9 CFU
SSD	ING-INF/05	
Language	Italian	
Year	first	
Semester	1 <sup>^</sup> semester	
Attendance	Attendance is strongly recommended	
Web site	<a href="https://www.uniba.it/it/corsi/bioinformatica">https://www.uniba.it/it/corsi/bioinformatica</a>	

## Docente/i

Name and Surname	Giovanni Dimauro
E-mail	<a href="mailto:giovanni.dimauro@uniba.it">giovanni.dimauro@uniba.it</a>
Telephone	080-5443294
Department and address	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n.617, 6 <sup>^</sup> piano.
Virtual room	TEAMS codice 7s0qsql
Office Hours (and modalities: e.g., by appointment, on line, etc.	Tuesday 11:00 – 13:00, but also by appointment by email

## Syllabus

Learning Objectives	The student must be able to analyze and solve simple problems, designing and developing programs in the C language.
---------------------	---




Course prerequisites	Good understanding of the English language. Individual reading by the student of the 1st chapter of the reference text (see below): "Introduction to computers, the Internet and the web"
Syllabus	<p>Presentation of the course, contents, exam methods, exemptions, attendance at lessons, timetables, methods of classroom practice, etc.</p> <p>Introduction to programming</p> <p>A simple C program: display a line of text.</p> <p>Another simple C program: adding two integers.</p> <p>Using the compiler</p> <p>Xcode development environment</p> <p>introduction to debugging</p> <p>tutorials and insights into the topics: A simple C program: display a line of text.</p> <p>Memory notions</p> <p>The arithmetic of C</p> <p>Making decisions: equality and relational operators.</p> <p>The algorithms</p> <p>The pseudocode.</p> <p>The control structures</p> <p>The if selection command</p> <p>The if else selection command.</p> <p>The while iteration command.</p> <p>Formulation of algorithms: case study: the iteration controlled by a counter.</p> <p>Algorithms formulation with top down process for successive refinements: case study: iteration controlled by a sentinel value.</p> <p>Type conversion, accuracy.</p> <p>Nested control structures</p> <p>exercise using pseudocode</p> <p>Algorithms formulation with top down process for successive refinements: case study: nested control structures</p> <p>The increment and decrement operators</p> <p>The elements of iteration</p> <p>Iteration controlled by a counter</p> <p>the FOR iteration command</p> <p>examples of using the FOR command</p> <p>The multi-select switch command</p> <p>introduction to the break command, case study.</p> <p>The do...while iteration command</p> <p>The break and continue statements</p> <p>Logical operators</p> <p>Equality (=) and assignment (=) operators;</p> <p>Summary of structured programming.</p> <p>Program modules in C</p> <p>The functions of the math library</p> <p>Functions</p> <p>Function definitions</p> <p>Function prototypes</p> <p>The function call stack and activation records</p> <p>The header files</p> <p>Invoking functions: calling by value and by reference</p> <p>Random number generation</p> <p>Exercise/self-assessment: flowchart development, activity carried out with the assistance of the teacher</p> <p>Example: a gambling game</p> <p>Memory classes</p> <p>Visibility rules</p> <p>The carriers</p> <p>The declaration of carriers</p> <p>Vector examples</p>



	<p>Exercise and individual study of video material produced by the teacher, to support the theoretical lessons. The video describes the development of a flowchart inspired by exercise 4.40 'World population growth' of the reference text. The design method respects the principles of structured programming and uses basic control structures of the C programming language - discussion with the teacher.</p> <p>More vector examples Static and automatic vectors Passing vectors to functions, passing by reference (address) and by value, 'const' qualifier Sorting algorithms: bubble sort Case study: Computing the mean, median, and mode using vectors Vector search (linear), algorithm and program for linear vector search Vector search (binary) Multidimensional vectors.</p> <p>Exercise and individual study of video material produced by the teacher, to support the theoretical lessons. The video describes the development of a 20-sided die exercise flowchart. The design method respects the principles of structured programming - discussion with the teacher.</p> <p>Exercise and individual study of video material produced by the teacher, to support the theoretical lessons. The video describes the transition from an algorithm described with a flowchart to the C language code, inspired by the '20-sided die' exercise. The design method respects the principles of structured programming and uses basic control structures of the C programming language - discussion with the teacher.</p> <p>Exercise and individual study of video material produced by the teacher, to support the theoretical lessons. The video describes the development of a flowchart inspired by the exercise 'Craps - a game of chance' in the reference text. The design method respects the principles of structured programming - discussion with the teacher.</p> <p>preparation for exemption. autonomous simulation in preparation for the 1st exemption, assisted by the teacher Teacher-assisted exemption self-correction The data hierarchy Files and streams Create a sequential access file Read data from a sequential access file Reading data from a sequential access file (more) Credit inquiry program. Random access files Create a random access file Write data randomly to a random access file Read data randomly from a random access file Exercise and Simulation 2nd exemption</p>
<b>Texts and readings</b>	<p>Text to study from:</p> <p>P. Deitel e H. Deitel Il linguaggio C – Fondamenti e tecniche di programmazione 8<sup>a</sup> edizione Pearson 2016 ISBN: 9788891901651 (vanno bene anche le edizioni successive e precedenti dalla 4<sup>a</sup> in poi)</p> <p>Additional text, optional:</p> <p>Kim N. King, Programmazione in C, Maggioli Editore, ISBN 8838785821</p>



	<p>Students who wish can obtain texts on loan from the Library. Could it be convenient to check their availability through the University Library System <a href="https://opac.uniba.it/easyweb/w8018/index.php?">https://opac.uniba.it/easyweb/w8018/index.php?</a> and contact the library to arrange the loan.</p>		
<p><b>Notes to the reference texts</b></p>	<p>During the lessons the teacher will use slides that retrace the contents of the book, therefore they will not be provided. The reference text contains all the topics of the course, therefore it is advisable to study from the text and to carry out independently and constantly all the exercises included at the end of each chapter covered in class.</p> <p>On the department's ADA platform (see above 'virtual office') are available:</p> <ul style="list-style-type: none"> <li>• supporting video material used in class;</li> <li>• some traces of written tests of exams, with examples of traces carried out;</li> </ul>		
<p><b>Organization of teaching</b></p>			
<p><b>Hours</b></p>			
<p>Total</p>	<p>Frontal teaching</p>	<p>Laboratory and exercises</p>	<p>Self study</p>
<p>225</p>	<p>48</p>	<p>45</p>	<p>132</p>
<p><b>CFU/ETCS</b></p>			
<p>9 CFU</p>	<p>7 CFU</p>	<p>2 CFU</p>	

<p><b>Teaching methods</b></p>	
	<p>Lectures, exercises and autonomous and group activities in the classroom and at home (as detailed in the program). Non-attending students can work individually by making arrangements with the teacher.</p>

<p><b>Expected learning outcomes</b></p>	
<p><b>Knowledge and understanding</b></p>	<ul style="list-style-type: none"> <li>• Acquire knowledge that allows the student to understand how it is possible to indicate to an electronic computer (automatic machine for universal use, hardware) the solution of a problem or a class of problems, which the computer</li> </ul>



	<p>can solve, with a method and a appropriate language, creating a special program (software) executable by the computer.</p> <ul style="list-style-type: none"><li>• Acquire the ability to reason and identify a solution to a problem (algorithm) according to the paradigm of structured imperative programming;</li></ul>
<b>Applied knowledge and understanding</b>	<ul style="list-style-type: none"><li>• Understand the use of an unconventional design language (eg pseudocode) and the use of a graphical representation (eg flow chart) to describe an algorithm with a simple formalism;</li><li>• Understand the lexicon, syntax and semantics of the C programming language;</li><li>• Acquire the ability to write a structured program in C language;</li><li>• Acquire the ability to identify test cases for the domain to which the created program refers;</li><li>• Acquire the ability to use a development environment (eg Xcode, Visual Studio) to transform the source program (in C) into an executable program and run it.</li></ul>
<b>Transversal skills</b>	<p><b>Judgment autonomy</b></p> <ul style="list-style-type: none"><li>• Acquire the ability to verify that the identified algorithm meets the specifics of a problem;</li><li>• To acquire the ability to verify that the results obtained after the execution of the program are those expected.</li></ul> <p><b>Communication skills</b></p> <ul style="list-style-type: none"><li>• Learn how to comment the product code in order to make it understandable and easily modifiable by other professionals, with the aim of developing in a team.</li></ul> <p><b>Ability to learn independently</b></p> <ul style="list-style-type: none"><li>• Ability to deepen concepts through the independent study of video material produced and proposed by the teacher;</li><li>• Ability to autonomously complete the educational path envisaged by the reference textbook, in addition to the contents envisaged by the teaching programme.</li></ul>

<b>Valutazione</b>	
<b>Assessment</b>	<p>A first mid-term evaluation test, with an exempt value, is held close to the week of interruption of lessons, normally located around mid-November. The test consists in solving a problem by identifying the algorithm and developing the relative program in C, similarly to what is explained in the lessons. The vote will be expressed in thirtieths.</p> <p>A second midterm evaluation test, with exempt value, is held immediately before or after the Christmas holidays, depending on the progress of the course. The second test consists of a closed multiple choice test, in particular about 50 questions will be administered and the expected time is normally 30 minutes. Each incorrect answer will be penalized with 0.25/50th of a point. The result obtained is normalized to 30.</p> <p>The mark resulting from the average of the marks obtained in the two exemption tests is proposed by the teacher to be recorded. The grades from the exemption can only be used in the first exam session (January/February).</p>



	<p>The student can, if he prefers, integrate the exam with an oral test. The vote is out of thirty.</p> <p>Students who choose not to take part in the mid-term assessment tests, or fail to pass them, take the exam starting in January. The theory test, if taken and passed in the mid-term evaluation, can be used by the student as a theory test in the regular exam.</p> <p>The modality of the regular exam is similar to that described above for the two intermediate tests (written exam and test).</p> <p>Materials allowed to take the first midterm evaluation test and the written exam test: reference text in paper format only.</p> <p>The results of all the tests are communicated with a public list (matriculation number and grade achieved) by email or social media. The final grade obtained is proposed exclusively on the Esse3 platform.</p> <p>Attendance incentives: any praise is most frequently given to students who, for the vast majority of lessons, have attended, interacted during the lesson, proposed solutions and solved the cases proposed by the teacher in class.</p>								
<p>Evaluation criteria</p>	<ul style="list-style-type: none"> <li>● Knowledge and understanding: <ul style="list-style-type: none"> <li>○ The student must be able to analyze and solve simple problems and to generalize solutions for a class of problems with the style of structured programming.</li> </ul> </li> <li>● Applied knowledge and understanding: <ul style="list-style-type: none"> <li>○ The student must be able to codify the solutions conceived by describing them in pseudocode, in structured flowcharts and in the C programming language;</li> <li>○ The student must be able to use a development environment of his choice and demonstrate knowledge of the C language;</li> </ul> </li> <li>● Judgment autonomy: <ul style="list-style-type: none"> <li>○ The student must be able to correct and validate the correct functioning of the developed programs.</li> </ul> </li> <li>● Communication skills: <ul style="list-style-type: none"> <li>○ The student must be able to make the written code understandable to others, through its general description and specific comments on the instructions and control structures used.</li> </ul> </li> <li>● Ability to learn: <ul style="list-style-type: none"> <li>○ The student must be able to autonomously transform algorithms described with flowcharts into programs in C language;</li> <li>○ The student must be able to use the alternative solutions described in the reference text, if not described in the lessons, such as for example the different ways of declaring variables.</li> </ul> </li> </ul>								
<p>Measurement criteria of learning and assignment of the final mark</p>	<table border="1"> <thead> <tr> <th>Voto</th> <th>Descrittori</th> </tr> </thead> <tbody> <tr> <td>&lt; 18 insufficient</td> <td>Fragmentary and superficial knowledge of the contents, errors in applying the concepts, deficient description.</td> </tr> <tr> <td>18 - 20</td> <td>Sufficient but general content knowledge, simple description, uncertainties in the application of theoretical concepts.</td> </tr> <tr> <td>21 - 23</td> <td>Appropriate but not in-depth knowledge of content, ability to apply theoretical concepts, ability to present content in a simple way.</td> </tr> </tbody> </table>	Voto	Descrittori	< 18 insufficient	Fragmentary and superficial knowledge of the contents, errors in applying the concepts, deficient description.	18 - 20	Sufficient but general content knowledge, simple description, uncertainties in the application of theoretical concepts.	21 - 23	Appropriate but not in-depth knowledge of content, ability to apply theoretical concepts, ability to present content in a simple way.
Voto	Descrittori								
< 18 insufficient	Fragmentary and superficial knowledge of the contents, errors in applying the concepts, deficient description.								
18 - 20	Sufficient but general content knowledge, simple description, uncertainties in the application of theoretical concepts.								
21 - 23	Appropriate but not in-depth knowledge of content, ability to apply theoretical concepts, ability to present content in a simple way.								



	<table border="1"><tbody><tr><td data-bbox="528 293 643 376">24 - 25</td><td data-bbox="643 293 1437 376">Appropriate and extensive knowledge of the contents, good ability to apply knowledge, ability to present the contents in an articulated way.</td></tr><tr><td data-bbox="528 376 643 459">26 - 27</td><td data-bbox="643 376 1437 459">Precise and complete content knowledge, good ability to apply knowledge, analytical skills, clear and correct description.</td></tr><tr><td data-bbox="528 459 643 568">28 - 29</td><td data-bbox="643 459 1437 568">Wide, complete and in-depth knowledge of the contents, good application of the contents, good capacity for analysis and synthesis, safe and correct description.</td></tr><tr><td data-bbox="528 568 643 678">30 30 cum laude</td><td data-bbox="643 568 1437 678">Very broad, complete and in-depth knowledge of the contents, well-established ability to apply the contents, excellent capacity for analysis, synthesis and interdisciplinary connections, mastery of description.</td></tr></tbody></table>	24 - 25	Appropriate and extensive knowledge of the contents, good ability to apply knowledge, ability to present the contents in an articulated way.	26 - 27	Precise and complete content knowledge, good ability to apply knowledge, analytical skills, clear and correct description.	28 - 29	Wide, complete and in-depth knowledge of the contents, good application of the contents, good capacity for analysis and synthesis, safe and correct description.	30 30 cum laude	Very broad, complete and in-depth knowledge of the contents, well-established ability to apply the contents, excellent capacity for analysis, synthesis and interdisciplinary connections, mastery of description.
24 - 25	Appropriate and extensive knowledge of the contents, good ability to apply knowledge, ability to present the contents in an articulated way.								
26 - 27	Precise and complete content knowledge, good ability to apply knowledge, analytical skills, clear and correct description.								
28 - 29	Wide, complete and in-depth knowledge of the contents, good application of the contents, good capacity for analysis and synthesis, safe and correct description.								
30 30 cum laude	Very broad, complete and in-depth knowledge of the contents, well-established ability to apply the contents, excellent capacity for analysis, synthesis and interdisciplinary connections, mastery of description.								
<b>Further information</b>	<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"><li>• <a href="https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea">https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea</a></li><li>• <a href="https://www.uniba.it/it/ricerca/dipartimenti/informatica">https://www.uniba.it/it/ricerca/dipartimenti/informatica</a></li><li>• <a href="https://elearning.di.uniba.it/">https://elearning.di.uniba.it/</a></li></ul> <p>Course schedules are available here:</p> <ul style="list-style-type: none"><li>• <a href="https://programmi.di.uniba.it/">https://programmi.di.uniba.it/</a></li></ul> <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"><li>• <a href="https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea">https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea</a></li></ul> <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.</p> <p>Students will be able to join the forum of the A.Y. 2022/23 on Telegram, used for educational purposes, to which the teacher also adheres: <a href="https://t.me/+V1eWiJ67GTBjNzNk">https://t.me/+V1eWiJ67GTBjNzNk</a></p> <p>Link to the course on the ADA department e-learning platform: <a href="https://elearning.di.uniba.it/">https://elearning.di.uniba.it/</a></p>								