



Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	Programmazione (track cognomi M-Z)	
Corso di studio	Informatica e Tecnologie dell'Informazione	
Anno Accademico	2025/26	
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS)	12 CFU	
Settore Scientifico Disciplinare	INF/01	
Lingua di erogazione	Italiano	
Anno di corso	Primo	
Periodo di erogazione	1 [^] semestre, le date esatte sono indicate annualmente nel manifesto/regolamento	
Obbligo di frequenza	La frequenza è fortemente raccomandata	
Sito web del corso di studio	https://www.uniba.it/it/corsi/informatica-tecnologie-informazione-taranto/informatica-e-tecnologie-dellinformazione	

Docente/i	
Nome e cognome	Giovanni Dimauro
Indirizzo mail	giovanni.dimauro@uniba.it
Telefono	080-5443294
Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n.617, 6 [^] piano.
Sede virtuale	Piattaforma e-learning UNIBA - https://elearning.uniba.it/
Sito web del docente	http://www.di.uniba.it/~dimauro/
Ricevimento (giorni, orari e modalità, es. su appuntamento)	appuntamento per email

Syllabus	
Obiettivi formativi	Il corso si propone di introdurre gli elementi base della programmazione imperativa strutturata per formulare soluzioni algoritmiche a problemi di complessità limitata. In particolare lo studente acquisirà la capacità di usare il linguaggio di



	programmazione C come strumento per modellare problemi e formalizzarne le soluzioni.																											
Prerequisiti	Buona comprensione della lingua inglese. Lettura individuale da parte dello studente del 1 [^] capitolo del testo di riferimento (v. sotto): “Introduzione ai computer, a Internet e al web”																											
Contenuti di insegnamento (Programma)	<table border="1"><thead><tr><th>Mod</th><th>Argomenti</th><th>Ore</th></tr></thead><tbody><tr><td>1</td><td>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.</td><td>3</td></tr><tr><td>2</td><td>Uso del compilatore ambiente di sviluppo Xcode cenni al debugging esercitazioni e approfondimenti degli argomenti: Un semplice programma C: visualizzare una riga di testo.</td><td>3</td></tr><tr><td>3</td><td>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</td><td>3</td></tr><tr><td>4</td><td>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.</td><td>4</td></tr><tr><td>5</td><td>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</td><td>3</td></tr><tr><td>6</td><td>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</td><td>4</td></tr><tr><td>7</td><td>Gli operatori logici Operatori di uguaglianza (==) e di assegnamento (=); Riassunto della programmazione strutturata. I moduli di programma in C</td><td>3</td></tr><tr><td>8</td><td>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</td><td>3</td></tr></tbody></table>	Mod	Argomenti	Ore	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	3
Mod	Argomenti	Ore																										
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	3																										



	Invocare le funzioni: chiamata per valore e per riferimento Generazione di numeri casuali	
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	La ricorsione	2
12	I vettori La dichiarazione dei vettori Esempi sui vettori	3
13	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
14	Ulteriori esempi sui vettori	3
15	Vettori statici ed automatici Passare i vettori alle funzioni, passaggio per riferimento (indirizzo) e per valore, qualificatore 'const' Algoritmi di ordinamento: bubble sort	4
16	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
17	La ricerca nei vettori (binaria) Vettori multidimensionali.	4
18	Manipolazione delle matrici esercitazione sulle matrici bidimensionali (fig.6.22 del testo 4 ^{ed})	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 dell'esercizio 'dado a 20 facce'. Il metodo di progettazione rispetta i principi della programmazione strutturata - discussione con il docente.	3
20	Esercitazione: Individuazione dei numeri primi. Crivello di Eratostene.	3
21	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
22	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
23	preparazione all'esonero.	4
24	simulazione autonoma in preparazione del 1 ^o esonero, assistita dal docente	3
25	Auto correzione esonero assistita dal docente	4
26	La gerarchia dei dati	4



	I file e gli stream Creare un file ad accesso sequenziale Leggere i dati da un file ad accesso sequenziale	
27	Leggere i dati da un file ad accesso sequenziale (approfondimenti) Programma per l'interrogazione del credito.	3
28	I file ad accesso casuale Creare un file ad accesso casuale	4
29	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
30	Studio di un caso: un programma per l'elaborazione delle transazioni	3
31	Approfondimenti sulla ricorsione ed esercizi: fattoriale ricorsivo, Fibonacci, Ricerca binaria ricorsiva e Torre di Hanoi.	3
32	Puntatori: introduzione dichiarazione e inizializzazione operatore di indirizzo (&) e operatore di dereferenziazione (*) passaggio per valore e per riferimento puntatori come parametri di funzioni Esercizi: dichiarazione e stampa dei valori contenuti in variabili e puntatori, elevare al cubo una variabile (chiamata per valore e chiamata per riferimento), trovare il massimo e il minimo di un vettore usando il passaggio per riferimento.	3
33	Puntatori: puntatori const, operatori sui puntatori, esercizio bubble sort con puntatori, esercizio bubble sort in ordine crescente e decrescente usando puntatori a funzione, Array multidimensionali ed esercizio mescola carte	4
	esercitazione autonoma, assistita dal docente; Correzione collettiva esercitazione svolta. Preparazione al 2 ^o esonero.	3
	Esercitazione e Simulazione 2 ^o esonero	3
Testi di riferimento	Testo da cui studiare: P. Deitel e H. Deitel Il linguaggio C – Fondamenti e tecniche di programmazione 8 ^a edizione Pearson 2016 ISBN: 9788891901651 (vanno bene anche le edizioni successive e precedenti dalla 4 ^a in poi) 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.	
Note ai testi di riferimento		





	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. Sulla piattaforma e-learning di Uniba (v. sopra 'sede virtuale') sono disponibili: <ul style="list-style-type: none">• materiale video di supporto utilizzato a lezione;• alcune tracce di prove scritte di esami, con esempi di tracce svolte;		
Organizzazione della didattica			
Ore			
Totali	Didattica frontale	Laboratorio ed esercitazioni	Studio individuale
300 ore	72 ore	45 ore	183 ore
CFU/ETCS			
12 CFU	9 CFU	3 CFU	

Metodi didattici	
	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.

Risultati di apprendimento previsti	
Conoscenza e capacità di comprensione	<ul style="list-style-type: none">• 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.• Acquisire la capacità di ragionare ed individuare una soluzione ad un problema (algoritmo) secondo il paradigma della programmazione imperativa strutturata;
Conoscenza e capacità di comprensione applicate	<ul style="list-style-type: none">• 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;• Comprendere il lessico, la sintassi e la semantica del linguaggio di programmazione C;• Acquisire la capacità di scrivere un programma strutturato in linguaggio C;



	<ul style="list-style-type: none">• Acquisire la capacità di individuare casi di test per il dominio cui fa riferimento il programma creato;• 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.
Competenze trasversali	<p>Autonomia di giudizio</p> <ul style="list-style-type: none">• Acquisire la capacità di verificare che l'algoritmo individuato risponda alle specifiche di un problema;• Acquisire la capacità di verificare che i risultati ottenuti dopo l'esecuzione del programma siano quelli attesi. <p>Abilità comunicative</p> <ul style="list-style-type: none">• Imparare a commentare il codice prodotto al fine di renderlo comprensibile e agevolmente modificabile da altri professionisti, con l'obiettivo di sviluppare in team. <p>Capacità di apprendere in modo autonomo</p> <ul style="list-style-type: none">• Capacità di approfondire concetti attraverso lo studio autonomo di materiale video prodotto e proposto dal docente;• Capacità di completare autonomamente il percorso formativo previsto dal testo di riferimento, oltre i contenuti previsti dal programma dell'insegnamento.

Valutazione	
Modalità di verifica dell'apprendimento	<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>								
<p>Criteria di valutazione</p>	<ul style="list-style-type: none"> ● Conoscenza e capacità di comprensione: <ul style="list-style-type: none"> ○ 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. ● Conoscenza e capacità di comprensione applicate: <ul style="list-style-type: none"> ○ Lo studente dovrà essere in grado di codificare le soluzioni ideate descrivendole in pseudocodice, in flowchart strutturati e nel linguaggio di programmazione C; ○ Lo studente dovrà essere in grado di utilizzare un ambiente di sviluppo a sua scelta e dimostrare di conoscere il linguaggio C; ● Autonomia di giudizio: <ul style="list-style-type: none"> ○ Lo studente dovrà essere in grado di correggere e validare il corretto funzionamento dei programmi sviluppati. ● Abilità comunicative: <ul style="list-style-type: none"> ○ 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. ● Capacità di apprendere: <ul style="list-style-type: none"> ○ Lo studente dovrà essere in grado di trasformare autonomamente algoritmi descritti con flowchart in programmi in linguaggio C; ○ 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. 								
<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>< 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> </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.
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.								



	<table border="1"><tbody><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>	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.
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.								
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• https://elearning.di.uniba.it/ <p>I programmi degli insegnamenti sono disponibili qui:</p> <ul style="list-style-type: none">• https://elearning.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><i>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.</i></p> <p>Gli studenti potranno unirsi al forum del corso di Programmazione (ITI) A.A. 2025/26 su Telegram, utilizzato per scopi didattici, al quale aderisce anche il docente: https://t.me/+zCJY-BioRxJkYWE0</p> <p>Link al corso sulla piattaforma e-learning di Uniba: https://elearning.uniba.it/</p>								