

<b>Principali informazioni sull'insegnamento</b>	
Titolo insegnamento	Programmazione
Corso di studio	Informatica e Tecnologie per la Produzione del Software
Crediti formativi	9+3
Denominazione inglese	Computer Programming
Obbligo di frequenza	No
Lingua di erogazione	Italiano

<b>Docente responsabile</b>	Nome Cognome	Indirizzo Mail
	Vito Renò	vito.reno@uniba.it

<b>Dettaglio credi formativi</b>	Ambito disciplinare	SSD	Crediti
	Informatico, fisico, matematico, economico, linguistico	ING-INF/05 – Sistemi di elaborazione delle informazioni	9 + 3

<b>Modalità di erogazione</b>	
Periodo di erogazione	I <sup>^</sup> semestre
Anno di corso	I <sup>^</sup>
Modalità di erogazione	Lezioni frontali, esercitazioni

<b>Organizzazione della didattica</b>	
Ore totali	117 (corso) + 183 (studio individuale)
Ore di corso	72 + 45
Ore di studio individuale	153 + 30

<b>Calendario</b>	
Inizio attività didattiche	Ottobre 2020
Fine attività didattiche	Gennaio 2021

<b>Syllabus</b>	
Prerequisiti	Buona comprensione della lingua inglese. Lettura individuale da parte dello studente del I <sup>^</sup> capitolo del testo di riferimento (v. sotto): "Introduzione ai computer, a Internet e al web"
Risultati di apprendimento previsti (declinare rispetto ai Descrittori di Dublino) (si raccomanda che siano coerenti con i risultati di apprendimento del CdS, riportati nei quadri A4a, A4b e A4c della SUA, compreso i risultati di apprendimento trasversali)	<ul style="list-style-type: none"> <li><i>Conoscenza e capacità di comprensione</i> Lo studente dovrà essere in grado di analizzare e risolvere semplici problemi, progettando e sviluppando programmi nel linguaggio C.</li> <li><i>Conoscenza e capacità di comprensione applicate</i> Lo studente dovrà acquisire competenze relative a: <ul style="list-style-type: none"> <li>- Traduzione di semplici algoritmi in programmi</li> </ul> </li> </ul>

	<p>correttamente funzionanti;</p> <ul style="list-style-type: none"> <li>- Verifica empirica della correttezza dei programmi mediante testing;</li> <li>- Capacità di individuazione di malfunzionamenti attraverso il debugging;</li> <li>- Capacità di problem-solving attraverso l'applicazione di nozioni apprese nelle discipline informatiche di base nella pratica della programmazione.</li> </ul> <ul style="list-style-type: none"> <li>• <i>Autonomia di giudizio</i> Lo studente deve dimostrare di aver acquisito autonomia di giudizio e di capacità di valutazione nell'ambito dello sviluppo di algoritmi.</li> <li>• <i>Capacità di apprendere</i> Lo studente dovrà mostrare di aver sviluppato capacità di apprendere e di orientarsi agilmente nelle problematiche relative alla comprensione e all'utilizzo delle tecnologie informatiche nel suo specifico campo di applicazione.</li> </ul>
<p>Contenuti di insegnamento</p>	<p><b>INTRODUZIONE ALLA PROGRAMMAZIONE IN C</b></p> <ul style="list-style-type: none"> <li>- Un semplice programma C: visualizzare una riga di testo</li> <li>- Un altro semplice programma C: sommare due interi</li> <li>- Nozioni sulla memoria</li> <li>- L'aritmetica del C</li> <li>- Prendere delle decisioni: gli operatori di uguaglianza e relazionali</li> </ul> <p><b>LO SVILUPPO DI PROGRAMMI STRUTTURATI IN C</b></p> <ul style="list-style-type: none"> <li>- Gli algoritmi</li> <li>- Lo pseudocodice</li> <li>- Le strutture di controllo</li> <li>- Il comando di selezione if</li> <li>- Il comando di selezione if else</li> <li>- Il comando di iterazione while</li> <li>- Formulazione degli algoritmi: studio di un caso (l'iterazione controllata da un contatore)</li> <li>- Formulazione degli algoritmi con processo top-down per raffinamenti successivi: studio di un caso (iterazione controllata da un valore sentinella)</li> <li>- Formulazione degli algoritmi con processo top-down per raffinamenti successivi: studio di un caso (strutture di controllo nidificate)</li> <li>- Gli operatori di assegnamento</li> <li>- Gli operatori di incremento e di decremento</li> </ul> <p><b>IL CONTROLLO DEL PROGRAMMA IN C</b></p> <ul style="list-style-type: none"> <li>- Gli elementi dell'iterazione</li> <li>- Iterazione controllata da un contatore</li> <li>- Il comando di iterazione for</li> <li>- Il comando for: note e osservazioni</li> <li>- Esempi di utilizzo del comando for</li> <li>- Il comando di selezione multipla switch</li> <li>- Il comando di iterazione do ... while</li> </ul>

- Le istruzioni break e continue
- Gli operatori logici
- Operatori di uguaglianza (==) e di assegnamento (=)
- Riassunto della programmazione strutturata

#### LE FUNZIONI IN C

- I moduli di programma in C
- 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
- Esempio: un gioco d'azzardo
- Le classi di memoria
- Le regole di visibilità
- La ricorsione
- Esempio di utilizzo della ricorsione: la serie di Fibonacci
- Ricorsione e iterazione

#### I VETTORI IN C

- I vettori
- La dichiarazione dei vettori
- Esempi sui vettori
- Passare i vettori alle funzioni
- L'ordinamento dei vettori
- Studio di un caso: calcolare la media, mediana e la moda usando i vettori
- La ricerca nei vettori
- I vettori multidimensionali

#### I PUNTATORI IN C

- Dichiarazione e inizializzazione dei puntatori
- Gli operatori sui puntatori
- Passare gli argomenti per riferimento alle funzioni
- Utilizzare il qualificatore const con i puntatori
- L'ordinamento a bolle utilizzando una chiamata per riferimento
- L'operatore sizeof
- Le espressioni con i puntatori e l'aritmetica dei puntatori
- La relazione tra i puntatori e i vettori
- I vettori di puntatori
- Studio di un caso: simulazione di un mescolatore e distributore di carte
- I puntatori a funzioni

#### L'ELABORAZIONE DEI FILE IN C

- La gerarchia dei dati
- I file e gli stream
- Creare un file ad accesso sequenziale
- Leggere i dati da un file ad accesso sequenziale
- I file ad accesso casuale
- Creare un file ad accesso casuale

	<ul style="list-style-type: none"> <li>- Scrivere i dati in modo casuale in un file ad accesso casuale</li> <li>- Leggere i dati in modo casuale da un file ad accesso casuale</li> <li>- Studio di un caso: un programma per l'elaborazione delle transazioni</li> </ul>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Programma	
Testi di riferimento	P. Deitel e H. Deitel – Il linguaggio C – Fondamenti e tecniche di programmazione 8 <sup>a</sup> edizione Pearson 2016 ISBN: 9788891901651
Note ai testi di riferimento	Testo integrativo consigliato: Kim N. King, Programmazione in C, Maggioli Editore, ISBN 8838785821
Metodi didattici	Lezioni frontali ed esercitazioni in aula
Metodi di valutazione (indicare almeno la tipologia scritto, orale, altro)	<p>Una prova di esonero 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. Il voto ottenuto alla prova di esonero può essere speso esclusivamente nel primo appello utile, pertanto gli studenti sono fortemente invitati ad utilizzare il primo appello di Programmazione appositamente collocato immediatamente dopo la chiusura del corso. Ciò al fine di lasciare ampio margine per preparare e sostenere gli altri due esami del I semestre.</p> <p>Si confida e si stimola quindi lo studio costante anche nel corso delle vacanze natalizie con lo scopo di porre lo studente in condizione di sostenere i 3 esami del primo semestre prima dell'inizio del secondo.</p> <p>Incentivi alla frequenza: L'eventuale lode potrà essere attribuita solo 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 (per ogni risultato di apprendimento atteso su indicato, descrivere cosa ci si aspetta lo studente conosca o sia in grado di fare e a quale livello al fine di dimostrare che un risultato di apprendimento è stato raggiunto e a quale livello)	<p>Lo studente dovrà dimostrare di aver acquisito la capacità di progettare un algoritmo per la soluzione di problemi con caratteristiche diverse. Inoltre deve aver sviluppato buone competenze nell'utilizzo del linguaggio C.</p> <p>L'esame finale consiste nella soluzione di un problema individuando l'algoritmo e sviluppando il relativo programma in C (per chi non ha sostenuto/superato la prova di esonero) e nella prova orale (oppure scritta in forma di test a risposta chiusa/aperta) utile a verificare la preparazione teorica.</p>
Altro	<p>Gli studenti potranno unirsi al Forum del corso A.A. 2020/2021 su Telegram, utilizzato per scopi didattici, al quale aderisce anche il docente:</p> <p><a href="https://t.me/joinchat/AO1wehwKrnAu0ey_lpXJaw">https://t.me/joinchat/AO1wehwKrnAu0ey_lpXJaw</a></p>