

<b>Principali informazioni sull'insegnamento</b>	
Titolo insegnamento	Programmazione
Corso di studio	Informatica e Comunicazione Digitale (sede di Taranto)
Crediti formativi	9+3
Denominazione inglese	Computer Programming
Obbligo di frequenza	NO
Lingua di erogazione	Italiano

<b>Docente responsabile</b>	Nome Cognome	Indirizzo Mail
	Domenico Redavid	domenico.redavid1@uniba.it

<b>Dettaglio credi formativi</b>	Ambito disciplinare	SSD	Crediti
	Informatico	INF/01 - Informatica	9+3

<b>Modalità di erogazione</b>	
Periodo di erogazione	I° semestre
Anno di corso	I°
Modalità di erogazione	Lezioni frontali (9 CFU) ed esercitazioni guidate (3 CFU)

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

<b>Calendario</b>	
Inizio attività didattiche	05/11/2020
Fine attività didattiche	12/01/2021

<b>Syllabus</b>	
Prerequisiti	Buona comprensione della lingua inglese. Lettura individuale da parte dello studente del I° 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 relativa: <ul style="list-style-type: none"> <li>○ Traduzione di semplici algoritmi in programmi correttamente funzionanti;</li> <li>○ Verifica empirica della correttezza dei programmi</li> </ul> </li> </ul>

	<p>mediante testing;</p> <ul style="list-style-type: none"> <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>Modulo 1 - Risolvere problemi con il computer: dal problema all'algoritmo</b></p> <p>Astrazione del processo di calcolo: algoritmo e caratteristiche. Riduzione di problemi complessi a sottoproblemi primitivi. Decomposizione di problemi attraverso sequenze, selezioni, relazioni di ricorrenza (iterazione e ricorsione). Principi della programmazione strutturata. Teorema di Bohm-Jacopini (enunciato). Notazioni grafiche e lineari per rappresentare algoritmi.</p> <p><b>Modulo 2 - Dall'algoritmo al programma</b></p> <p>Delega della esecuzione di un procedimento di calcolo: il ruolo del linguaggio. Linguaggi a basso e ad alto livello. Cenni sui compilatori. Linguaggi imperativi.</p> <p><b>Modulo 3 - Linguaggi di programmazione: dati e controllo</b></p> <p>Dati: tipi di dato e rappresentazioni. Tipi semplici. Compatibilità ed equivalenza tra tipi di dato. Variabili ed istruzione di assegnazione. Strutture di controllo di base. Meccanismi di definizione di strutture di dati: trasformazioni finite (array), prodotto cartesiano (record), insieme potenza (set), sequenze (file). Astrazioni funzionali, procedure e funzioni. Dichiarazioni e campo di azione degli identificatori. Tecniche di legame dei parametri. Effetti collaterali in procedure e funzioni. Ricorsione. Attivazione delle procedure/funzioni.</p>

	<p><b>Modulo 4 – Il Linguaggio C</b></p> <p>Alfabeto. Costanti, tipi, variabili. Dichiarazioni. Tipi semplici. Tipi enumerativi. Istruzione di assegnazione ed espressioni. Procedure di Input/Output. Operatori aritmetici. Operatori di relazione. Operatori booleani. Selezione: istruzioni condizionali. Strutture iterative. Array, record, file. Procedure e funzioni: dichiarazione, parametri formali, attivazione e parametri attuali, puntatori in C, elaborazione di file in C.</p> <p><b>Modulo 5 - Algoritmi fondamentali</b></p> <p>Algoritmi elementari. Algoritmi su array: inversione, rimozione di valori duplicati, ricerca, ordinamento, partizionamento, fusione. Algoritmi ricorsivi: fattoriale, quicksort. Algoritmi su file di testo: ricerca di parole chiave.</p>
--	--

<b>Programma</b>	
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	Kim N. King, Programmazione in C, Maggioli Editore ISBN: 8838785821
Metodi didattici	Lezioni frontali teoriche Esercitazioni guidate
Metodi di valutazione (indicare almeno la tipologia scritto, orale, altro)	L'esame consiste in una prova di laboratorio e in una prova orale. La prova di laboratorio è propedeutica a quella orale.
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)	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. L'esame finale consiste nella soluzione di un problema individuando l'algoritmo e sviluppando il relativo programma in C e nella prova orale utile a verificare la preparazione teorica.
Altro	