Principali informazioni sull'insegnamento	**A.A. 2018-2019**
Titolo insegnamento	Programmazione
Corso di studio	Informatica e Comunicazione Digitale Taranto
Crediti formativi	12
Denominazione inglese	Computer Programming
Obbligo di frequenza	
Lingua di erogazione	Italiano

Docente responsabile	Nome Cognome	Indirizzo Mail
	Carmelo Antonio Ardito	carmelo.ardito@uniba.it

Dettaglio credi formativi	Ambito disciplinare	SSD	Crediti
	Informatico	INF/01 - Informatica	12

Modalità di erogazione	
Periodo di erogazione	Primo Semestre
Anno di corso	Primo Anno
Modalità di erogazione	Lezioni frontali Esercitazioni in aula e Laboratorio

Organizzazione della didattica	
Ore totali	301
Ore di corso	117 (72 ore di lezioni frontali, 45 ore di laboratorio)
Ore di studio individuale	183

Calendario	
Inizio attività didattiche	25 settembre 2018
Fine attività didattiche	11 gennaio 2019

Syllabus	
Prerequisiti	Non sono richiesti prerequisiti particolari
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)	• *Conoscenza e capacità di comprensione* Le competenze relative ai concetti di base della programmazione attraverso l'uso di linguaggi imperativi, insieme a metodologie e tecniche per progettare e formalizzare soluzioni da implementare, successivamente, in programmi scritti nel linguaggio di programmazione C, verranno trasmesse dal docente tramite lezioni teoriche ed esercitazioni pratiche da svolgere in laboratorio sotto la supervisione e la guida del docente. • *Conoscenza e capacità di comprensione applicate* Verranno erogate lezioni frontali e di laboratorio per consentire agli studenti di mettere in pratica le conoscenze trasmesse dal docente a lezione e l'auto-verifica della comprensione delle stesse. Una prova in itinere, da svolgersi

	approssimativamente a metà corso, permetterà di applicare e verificare le conoscenze acquisite fino a quel momento. • *Autonomia di giudizio* Lo studente deve dimostrare di aver acquisito una notevole autonomia di giudizio e di gestione delle problematiche relative alla formulazione di algoritmi per la soluzione dei problemi posti e la loro implementazione nel linguaggio di programmazione di riferimento. L'autonomia di giudizio è parte della valutazione finale dello studente, che tiene anche conto delle discussioni avvenute durante le lezioni, le esercitazioni e la prova d'esame finale. • *Abilità comunicative* Lo studente dovrà essere in grado di illustrare in modo appropriato le competenze relative ai concetti di base della programmazione. L'abilità comunicativa sarà esercitata durante l'interazione con il docente che avviene nelle lezioni frontali e di laboratorio, quando gli studenti saranno chiamati ad illustrare gli esercizi svolti. Inoltre, l'abilità comunicativa sarà verificata durante la prova orale. • *Capacità di apprendere* Per stimolare la capacità di apprendere in modo autonomo, agli studenti sono consigliati, oltre al materiale di studio principale, altri testi in cui approfondire alcuni specifici argomenti, alcuni non trattati in dettaglio dal docente, sui quali lo studente deve poi discutere a lezione, e riferire anche durante l'esame.
Contenuti di insegnamento	Lo studente acquisirà le competenze relative ai concetti di base della programmazione attraverso l'uso di linguaggi imperativi. Dovrà essere in grado di analizzare semplici problemi e risolverli progettando e formalizzando soluzioni da implementare, successivamente, in programmi scritti nel linguaggio di programmazione C. 1.Introduzione Problem solving: algoritmi e programmi. Specifica di un algoritmo: diagrammi di flusso, albero di decomposizione, linguaggio naturale, pseudocodice. Programmazione strutturata. Teorema di Bohem-Jacopini (enunciato). Linguaggi assemblativi e di alto livello. Linguaggi imperativi. Cenni sulla loro evoluzione. Traduttori. 2.Linguaggi di programmazione: dati e controllo Tipi di dato. Tipi semplici. Compatibilità ed equivalenza tra tipi di dato. Variabili e costanti. Istruzione di assegnazione. Strutture di controllo di base. Astrazione funzionale mediante sottoprogrammi (procedure e funzioni).

	Identificatori e scope / campo di visibilità di un identificatore. Parametri formali ed effettivi, tecniche di legame dei parametri. Effetti collaterali in procedure e funzioni. Gestione delle attivazioni dei sottoprogrammi. Ricorsione. Strutture dati fondamentali (array e struct). Puntatori. Cenni sulla allocazione dinamica di variabili. 3.Metodologie di programmazione Cenni su programmazione in grande e programmazione in piccolo e sulle metodologie di progetto top-down e bottom-up. Albero di decomposizione funzionale. Progetto di un algoritmo: raffinamenti successivi e pseudocodifica. 4.Algoritmi fondamentali Algoritmi elementari: conteggio, sommatoria di un insieme di numeri, calcolo del fattoriale, conversione da caratteri a numeri in base 10, e da numero in base 10 a caratteri, massimo comun divisore, calcolo di un numero di Fibonacci. Algoritmi su array: ricerca del massimo e minimo, calcolo del valore medio, inversione degli elementi e rimozione valori duplicati. Algoritmi di ordinamento, ricerca e fusione: ordinamento per selezione, per inserzione e per scambi, ricerca lineare e binaria, fusione di 2 vettori ordinati. 5.Linguaggio C Introduzione all'uso del linguaggio di programmazione Aritmetica in C. Struttura dei programmi. Tipi di dati semplici predefiniti e definiti dall'utente. Gli Array, le strutture e i puntatori. Compatibilità dei tipi. Strutture di controllo: istruzioni di selezione If, Switch. Istruzioni iterative: while, do-while, for. Funzioni e procedure: definizione, chiamata, prototipo, passaggio dei parametri. Ambito di visibilità delle variabili. Uso dei parametri di tipo array. Uso dei parametri di tipo struttura. Procedure e funzioni predefinite. Standard library. Concetti fondamentali su caratteri e stringhe. 6. File File. Stream Testuali e Binari. Operazioni su file. Argomenti di un programma (argc, argv). 7. Debugging Tecniche e strumenti di debugging.

Programma	
Testi di riferimento	H.M. Deitel, P.J. Deitel - Il linguaggio C. Fondamenti e tecniche di programmazione - Pearson (guida alla programmazione per il linguaggio C)

Note ai testi di riferimento	I libri di testo sono integrati con le slide e le dispense del docente, disponibili sulla piattaforma di e-learning usata dal CdS.
Metodi didattici	Lezioni frontali ed esercitazioni pratiche relative all' identificazione, definizione e implementazione di una soluzione ai problemi posti.
Metodi di valutazione (indicare almeno la tipologia scritto, orale, altro)	L'esame consiste in una prova pratica (identificata sul portale ESSE3 come "Prova Parziale") e in una prova orale. La prova pratica consiste di due fasi: 1. Progettazione di una soluzione al problema riportato nella traccia data, secondo la struttura (Definizione dei dati di Input/Output, scomposizione del problema, definizione della soluzione tramite flow-chart) studiata durante il corso. Il problema è quello proposto nella prova in itinere, ma presenta richieste ulteriori (ad esempio, ordinamento dei dati). 2. Codifica in linguaggio C di un programma che, rispettando i principi della buona programmazione, implementa la soluzione definita dallo studente nella prova scritta, utilizzando i PC presenti in laboratorio e l'ambiente di sviluppo su questi installato (Eclipse CDT). Si accede alla prova orale se la prova pratica è stata valutata come sufficiente dal docente. La prova orale consiste nella discussione della prova pratica e nell'eventuale approfondimento di alcuni argomenti. Sono inoltre previste due prove in itinere che, se superate entrambe, hanno valore esonerante dalla prova pratica finale. Alle prove non è possibile utilizzare alcun sussidio (libri, appunti personali, etc.).
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)	Verranno valutate le capacità di problem solving e di utilizzo degli strumenti informatici teorici e pratici appresi a lezione.
Altro	