

Principali informazioni sull'insegnamento	
Titolo insegnamento	PROGRAMMAZIONE
Corso di studio	INFORMATICA triennale
Crediti formativi	12 (9+3)
Denominazione inglese	PROGRAMMING
Obbligo di frequenza	NO
Lingua di erogazione	ITALIANO

Docente responsabile	Nome Cognome	Indirizzo Mail
	TERESA ROSELLI	teresa.roselli@uniba.it

Dettaglio credi formativi	Ambito disciplinare	SSD	Crediti
	Informatica	INF/01	12 (9+3)

Modalità di erogazione	
Periodo di erogazione	Primo semestre
Anno di corso	Primo anno
Modalità di erogazione	Lezioni frontali, esercitazioni, laboratorio

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

Calendario	
Inizio attività didattiche	25 settembre 2017
Fine attività didattiche	12 gennaio 2018

Syllabus	
Prerequisiti	Non è richiesto alcun prerequisito in quanto trattasi di un insegnamento del primo semestre del primo anno
Risultati di apprendimento previsti	<ul style="list-style-type: none"> Conoscenza e capacità di comprensione Lo studente acquisirà la conoscenza degli aspetti teorici e pratici relativi alla progettazione delle soluzioni di problemi (problem solving) mediante l'uso del computer e la conoscenza degli strumenti che si utilizzano per la programmazione. Conoscerà i principi della programmazione strutturata, il concetto di tipo e di strutture dati predefinite e definite dall'utente. Acquisirà la conoscenza dei principi dell'astrazione funzionale, l'uso di funzioni e procedure e le modalità di passaggio dei parametri. Lo studente sarà in grado di comprendere l'effetto delle operazioni in termini di sintassi e semantica e acquisirà la

	<p>conoscenza del linguaggio di programmazione imperativo di riferimento (linguaggio C).</p> <ul style="list-style-type: none"> • <i>Conoscenza e capacità di comprensione applicate</i> Lo studente sarà in grado di applicare la conoscenza e gli strumenti acquisiti per risolvere computazionalmente semplici problemi anche in contesti concreti. Queste competenze saranno acquisite non solo mediante le lezioni teoriche ma, soprattutto, durante le esercitazioni e le attività di laboratorio in cui saranno affrontati e risolti in modalità partecipata problemi di varia natura che prevedono l'applicazione di ogni concetto presentato. • <i>Autonomia di giudizio</i> Lo studente sarà in grado di saper valutare e interpretare in maniera autonoma diverse strategie risolutive analizzando gli algoritmi proposti e fornendo soluzioni alternative. • <i>Abilità comunicative</i> Lo studente sarà in grado di redigere e presentare la documentazione relativa alle scelte progettuali applicando la metodologia di soluzione dei problemi e usando il livello di formalismo richiesto e la terminologia informatica. • <i>Capacità di apprendere</i> Lo studente sarà in grado di riutilizzare le conoscenze acquisite sia in situazioni problematiche nuove, sia in contesti nuovi di programmazione imperativa. Lo studente sarà in grado di consultare materiale didattico bibliografico anche attraverso piattaforme di e-learning.
<p>Contenuti di insegnamento</p>	
<p>Programma</p>	<p>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). Il paradigma imperativo (cenni). I traduttori (cenni).</p> <p>2. Linguaggi di programmazione: dati e controllo Tipi di dato. Tipi semplici. Compatibilità tra tipi di dato. Variabili e costanti. Istruzione di assegnazione. Strutture di controllo di base. Astrazione funzionale. Sottoprogrammi: procedure e funzioni. Visibilità degli identificatori. Parametri formali ed effettivi, tecniche di legame dei parametri (per valore, per riferimento). Effetti collaterali. Gestione delle attivazioni dei sottoprogrammi. Ricorsione. Tipi strutturati (array e record). Puntatori. File sequenziali.</p> <p>3. Metodologie di programmazione Cenni su programmazione in piccolo e sulle metodologie di</p>

	<p>progetto top-down e bottom-up. Specifiche del problema. Progetto di un algoritmo: raffinamenti successivi e pseudo-codifica. Stile di programmazione. Auto-documentazione dei programmi.</p> <p>4. Algoritmi fondamentali Algoritmi elementari: conteggio, sommatoria e media di un insieme di numeri, fattoriale, conversione da caratteri a numeri in base 10, numero primo, massimo comun divisore, serie di Fibonacci. Algoritmi su array: stampa istogrammi mediante array, elimina duplicati su array ordinati. Algoritmi su matrici. Algoritmi di ordinamento: sort per selezione, per scambio, per inserzione. Ricerca lineare. Ricerca binaria. Fusione di array. Algoritmi ricorsivi: Fattoriale, Fibonacci, Sort per fusione su array. Algoritmi su file.</p> <p>5. Linguaggio C Struttura dei programmi C. Tipi di dati: semplici predefiniti – int, float, double e char-, enumerazione esplicita dei valori. Definizione di tipo. Tipi strutturati: il costruttore array, il costruttore struct ed il costruttore puntatore. 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. Array come parametri. Strutture come parametri. Effetti collaterali. Procedure e funzioni predefinite. Standard library. Puntatori. File.</p>
--	--

Programma	
Testi di riferimento	Batini et al. - Fondamenti di programmazione dei Calcolatori Elettronici. Franco Angeli 1992 Dromey R.G. - Algoritmi fondamentali, Jackson Libri 1990 Deitel P. & H. - Il linguaggio C - Fondamenti e tecniche di programmazione, ed. marzo 2013 - Pearson (ISBN: 9 788871 929378) Ed.italiana a cura di Salvatore Gaglio
Note ai testi di riferimento	In biblioteca sono disponibili copie dei testi. Ai fini dell'esame, da ciascun testo occorre studiare parti specifiche, come descritto di seguito. Batini et al. - Fondamenti di Programmazione dei Calcolatori Elettronici. Franco Angeli 1992 Introduzione, Cap.1, Cap.2; Cap.3: pp.91-105; Cap.4; Cap.5(cenni); Cap.6: paragrafo 4 Il problema dell'ordinamento; Cap.10: pp.401-409. Si consiglia la lettura approfondita delle appendici.

	<p>Dromey R. G. - Algoritmi fondamentali, Jackson Libri 1990 Da Cap. 1 a Cap.5; Capitolo 8: pp.371-377.</p> <p>Il materiale didattico presentato a lezione è disponibile sulla piattaforma di e-learning del Dipartimento di Informatica.</p>
Metodi didattici	<p>Lezioni frontali teoriche Esercitazioni e attività di laboratorio guidate</p>
Metodi di valutazione (indicare almeno la tipologia scritto, orale, altro)	<p>L'esame consiste in una prova di laboratorio e in una prova orale. La prova di laboratorio è propedeutica a quella orale. La prova di laboratorio è simile a quella proposta come prova in itinere, in più prevede la codifica nel linguaggio di programmazione imperativo.</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>Riguardo la prova di laboratorio, lo studente dovrà essere in grado di analizzare il problema proposto formulando anche ipotesi aggiuntive, individuare i dati necessari e sufficienti per la soluzione fornendone la descrizione. Dovrà dimostrare di saper individuare una strategia di soluzione che prevede la scomposizione del problema in sottoproblemi e di saper rappresentare sia la scomposizione sia gli algoritmi con adeguati linguaggi di descrizione presentati durante le lezioni. Lo studente, inoltre, dovrà dimostrare di saper implementare la soluzione proposta utilizzando il linguaggio imperativo di riferimento e di saperla testare su campioni di dati.</p> <p>Riguardo la prova orale, lo studente dovrà dimostrare di saper discutere la soluzione prodotta nella prova di laboratorio chiarendo le scelte progettuali e implementative. Inoltre, dovrà dimostrare di aver acquisito piena conoscenza dei concetti presentati a lezione nonché degli algoritmi fondamentali.</p>
Altro	