

<b>Principali informazioni sull'insegnamento</b>	<b>A.A. 2018-2019</b>
Titolo insegnamento	Laboratorio di Informatica
Corso di studio	Informatica
Crediti formativi	3 (Teoria) + 3 (Laboratorio)
Denominazione inglese	Computer Science Laboratory
Obbligo di frequenza	No
Lingua di erogazione	Italiano

<b>Docente responsabile</b>	Nome Cognome	Indirizzo Mail
	Veronica Rossano	veronica.rossano@uniba.it
Luogo ed Orario di Ricevimento	7 piano - stanza 772	Lunedì 9 - 12

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

<b>Modalità di erogazione</b>	
Periodo di erogazione	II°
Anno di corso	I
Modalità di erogazione	Lezioni frontali, Esercitazioni in Laboratorio

<b>Organizzazione della didattica</b>	
Ore totali	150
Ore di corso	24+45
Ore di studio individuale	81

<b>Calendario</b>	
Inizio attività didattiche	25 febbraio 2019
Fine attività didattiche	31 maggio 2019

<b>Syllabus</b>	
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)	<ol style="list-style-type: none"> <li>1. <i>Conoscenza e capacità di comprensione</i> Lo studente dovrà essere in grado di progettare e sviluppare programmi, di media complessità utilizzando il linguaggio C.</li> <li>2. <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 correttamente funzionanti e ben documentati;</li> <li>○ Utilizzo di tecniche di programmazione difensiva, per limitare l'introduzione di malfunzionamenti nei programmi;</li> <li>○ Verifica empirica della correttezza dei programmi</li> </ul> </li> </ol>

	<p>mediante testing;</p> <ul style="list-style-type: none"> <li>○ Capacità di problem-solving attraverso l'applicazione di nozioni apprese nelle discipline informatiche di base nella pratica della programmazione.</li> </ul> <p>3. <i>Autonomia di giudizio</i> Lo studente deve dimostrare di aver acquisito autonomia di giudizio e capacità di valutazione degli algoritmi sviluppati da lui o da terzi.</p> <p>4. <i>Abilità comunicative</i> Lo studente deve essere in grado di illustrare in modo appropriato le caratteristiche tecniche degli strumenti e delle metodologie informatiche apprese nel corso del primo anno di corso, relative allo sviluppo di programmi di media complessità.</p> <p>5. <i>Capacità di apprendere</i> Lo studente dovrà mostrare di essere in grado di orientarsi agevolmente nelle problematiche relative alla comprensione e all'utilizzo delle tecnologie e dei metodi di competenza per lo sviluppo di algoritmi e per la loro traduzione in programmi per computer.</p>
<p>Contenuti di insegnamento</p>	<ol style="list-style-type: none"> <li>1. Stili di programmazione <ul style="list-style-type: none"> <li>○ Motivazioni</li> <li>○ Uso appropriato dei nomi</li> <li>○ Scrittura appropriata di espressioni e istruzioni</li> <li>○ Consistenza ed espressioni idiomatiche</li> <li>○ Commenti</li> <li>○ Convenzioni di programmazione</li> </ul> </li> <li>2. Testing e Debugging <ul style="list-style-type: none"> <li>○ Bug</li> <li>○ Tecniche di debugging</li> <li>○ Strumenti per il debugging</li> <li>○ Generalità sul testing</li> <li>○ Il test di unità</li> <li>○ Tecniche di testing</li> </ul> </li> <li>3. Programmazione modulare <ul style="list-style-type: none"> <li>○ Modularizzazione e strutturazione dei programmi</li> <li>○ Strutturazione dei file sorgente</li> <li>○ Strutturazione di progetti in Eclipse CDT</li> </ul> </li> <li>4. Documentazione del codice <ul style="list-style-type: none"> <li>○ Generalità sulla documentazione di codice in linea</li> <li>○ Documentazione automatica di codice</li> </ul> </li> <li>5. Puntatori in C</li> </ol>

	<ul style="list-style-type: none"> <li>○ Puntatori</li> <li>○ Procedure e funzioni</li> <li>○ I/O</li> <li>○ Memoria dinamica (cenni)</li> </ul> <p>6. File</p>
--	---

<b>Programma</b>	
Testi di riferimento	<ol style="list-style-type: none"> <li>1. P. Deitel e H. Deitel, Corso completo di programmazione. Apogeo</li> <li>2. W. B. Kernighan, D.M. Ritchie. Il linguaggio C. Principi di programmazione e manuale di riferimento.</li> </ol>
Note ai testi di riferimento	<ol style="list-style-type: none"> <li>1. P. Deitel e H. Deitel, Il Linguaggio C - Fondamenti e tecniche di programmazione, Pearson, 2013</li> </ol> <p>I libri di testo sono integrati con dispense e articoli scientifici disponibili sulla piattaforma di e-learning all'indirizzo <a href="https://elearning.di.uniba.it/">https://elearning.di.uniba.it/</a></p>
Metodi didattici	Lezioni frontali e attività laboratoriali.
Metodi di valutazione (indicare almeno la tipologia scritto, orale, altro)	Alcune prove pratiche da svolgere in itinere, non obbligatorie. Il superamento delle prove in itinere e/o i risultati delle esercitazioni pratiche attribuiscono una premialità sul voto finale. Caso di studio e 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 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. Riguardo la prova orale, lo studente dovrà dimostrare di saper discutere la soluzione prodotta nel caso di studio chiarendo le scelte progettuali e implementative. Inoltre, dovrà dimostrare di aver acquisito piena conoscenza dei concetti presentati a lezione nonché degli algoritmi fondamentali.
Altro	