

Principali informazioni sull'insegnamento	A.A. 2020-2021
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

Docente responsabile	Nome Cognome	Indirizzo Mail
	Veronica Rossano	veronica.rossano@uniba.it
Luogo ed Orario di Ricevimento	7 piano - stanza 771	Martedì 9.30 – 11.30

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

Modalità di erogazione	
Periodo di erogazione	II°
Anno di corso	I
Modalità di erogazione	Lezioni frontali, Esercitazioni in Laboratorio

Organizzazione della didattica	
Ore totali	150
Ore di corso	24+45
Ore di studio individuale	81

Calendario	
Inizio attività didattiche	1 marzo 2021
Fine attività didattiche	4 giugno 2021

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)	<ol style="list-style-type: none"> 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. 2. <i>Conoscenza e capacità di comprensione applicate</i> Lo studente dovrà acquisire competenze relative a: <ul style="list-style-type: none"> ○ Traduzione di semplici algoritmi in programmi correttamente funzionanti e ben documentati; ○ Utilizzo di tecniche di programmazione difensiva, per limitare l'introduzione di malfunzionamenti nei programmi; ○ Verifica empirica della correttezza dei programmi

	<p>mediante testing;</p> <ul style="list-style-type: none"> ○ Capacità di problem-solving attraverso l'applicazione di nozioni apprese nelle discipline informatiche di base nella pratica della programmazione. <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>	<p>Programmazione di base in C</p> <ul style="list-style-type: none"> • Concetti base di Programmazione • IDE per la creazione di progetti <p>Programmazione avanzata</p> <ul style="list-style-type: none"> • Puntatori, Struct e File <p>Stili di Programmazione</p> <ul style="list-style-type: none"> • Uso appropriato dei nomi • Scrittura appropriata di espressioni e istruzioni • Consistenza ed espressioni idiomatiche • Commenti • Convenzioni di programmazione <p>Programmazione modulare</p> <ul style="list-style-type: none"> • Modularizzazione e strutturazione dei programmi • Strutturazione dei file sorgente • Strutturazione di progetti in Eclipse CDT <p>Testing e Debugging</p> <ul style="list-style-type: none"> • Strumenti e tecniche di debugging • Generalità sul testing • Tecniche di testing

	<p>Documentazione del codice</p> <ul style="list-style-type: none"> • Generalità sulla documentazione di codice in linea • Doxygen per la documentazione automatica di codice <p>Algoritmi Fondamentali</p> <ul style="list-style-type: none"> • Algoritmi di Ricerca • Algoritmi di Ordinamento • Complessità Computazionale (cenni)
--	--

Programma	
Testi di riferimento	<ol style="list-style-type: none"> 1. P. Deitel e H. Deitel, Corso completo di programmazione. Apogeo 2. W. B. Kernighan, D.M. Ritchie. Il linguaggio C. Principi di programmazione e manuale di riferimento.
Note ai testi di riferimento	<ol style="list-style-type: none"> 1. P. Deitel e H. Deitel, Il Linguaggio C - Fondamenti e tecniche di programmazione, Pearson, 2013 <p>I libri di testo sono integrati con dispense e articoli scientifici disponibili sulla piattaforma di e-learning all'indirizzo https://elearning.di.uniba.it/</p>
Metodi didattici	Lezioni frontali e attività laboratoriali.
Metodi di valutazione (indicare almeno la tipologia scritto, orale, altro)	<ol style="list-style-type: none"> 1) Prove pratiche da svolgere in itinere, non obbligatorie. La partecipazione e il superamento delle prove pratiche attribuiscono una premialità. 2) Caso di Studio. Per superare l'esame è richiesta la realizzazione di un caso di studio da cui si evinca l'acquisizione delle competenze relative ai diversi argomenti illustrati durante il corso. Il progetto deve essere completo di documentazione e codice sorgente. 3) Prova orale. Discussione orale del caso di studio in cui lo studente dovrà dimostrare di aver acquisito le conoscenze illustrate durante il corso. 4) Prova intermedia (durante lo svolgimento l'insegnamento) – E' prevista una prova scritta intermedia, articolata su una specifica parte del programma indicata dal docente prima dell'interruzione delle lezioni prevista da calendario accademico. Il superamento della prova intermedia esonera lo studente dallo svolgimento di una parte della prova d'esame, ed è valida per l'intero anno accademico.
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	<p>Lo studente dovrà essere in grado di:</p> <ul style="list-style-type: none"> - Analizzare il problema proposto, formulando anche ipotesi aggiuntive, individuare i dati necessari e sufficienti per la soluzione

<p>apprendimento è stato raggiunto e a quale livello)</p>	<ul style="list-style-type: none"> - Saper individuare una strategia di soluzione corretta e progettartela utilizzando un linguaggio strutturato - Saper utilizzare l'approccio modulare per la realizzazione di una soluzione - Saper scrivere programmi leggibili, ben commentati e documentati - Saper utilizzare metodi e tecniche di testing e di debugging. <p>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.</p>
<p>Altro</p>	