

Principali informazioni sull'insegnamento		A.A. 2020/2021	
Titolo insegnamento	Laboratorio di Informatica (corso B)		
Corso di studio	Informatica e Tecnologie per la Produzione del Software		
Crediti formativi	6 (3 + 3)		
Denominazione inglese	Computer Programming Lab		
Obbligo di frequenza	No		
Lingua di erogazione	Italiano		
Docente responsabile	Nome Cognome	Indirizzo Mail	
	Alessandra Vitanza	alessandra.vitanza@uniba.it	
Dettaglio credi formativi	Ambito disciplinare	SSD	Crediti
	Informatico	ING-INF/05	6
Modalità di erogazione			
Periodo di erogazione	II Semestre		
Anno di corso	I Anno		
Modalità di erogazione	Lezioni frontali + Esercitazioni in aula		
Organizzazione della didattica			
Ore totali	150 (69 + 81)		
Ore di corso	69 (24 + 45)		
Ore di studio individuale	81 (51 + 30)		
Calendario			
Inizio attività didattiche	1 Marzo 2021		
Fine attività didattiche	4 Giugno 2021		
Syllabus			
Prerequisiti	<ul style="list-style-type: none"> • Propedeuticità obbligatorie: nessuna • Propedeuticità consigliate: poiché il corso prevede alcune conoscenze di base e i classici paradigmi di programmazione è fortemente consigliato aver frequentato gli insegnamenti di Programmazione e Architettura degli Elaboratori e Sistemi Operativi erogati durante il Primo Semestre. 		
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	<p>Il corso si propone di fornire allo studente gli strumenti concettuali, teorici e metodologici indispensabili per una buona padronanza delle tecniche di programmazione.</p> <p>In particolare, l'insegnamento si propone di approfondire tutti quei temi, di carattere</p>		

<p>SUA, compreso i risultati di apprendimento trasversali)</p>	<p>prevalentemente pratico, che abilitano gli studenti a scrivere programmi coerenti con le finalità dei problemi e quesiti pratici proposti durante le esercitazioni. A tal fine, l'insegnamento di Laboratorio di Informatica approfondisce temi quali la programmazione difensiva (corretto stile di scrittura dei programmi, uso appropriato della nomenclatura, ecc.), testing e debugging, corretta documentazione del codice, progettazione modulare, problem solving, ecc.</p> <p>Al termine del corso lo studente avrà acquisito:</p> <ul style="list-style-type: none"> ✓ Capacità di comprensione: dimostrando di comprendere i principi della programmazione e progettazione modulare, insieme alle principali tecniche di testing, debugging e documentazione del codice. ✓ Capacità di comprensione applicate: dimostrando di comprendere come scrivere del codice sorgente efficiente, corretto e adeguatamente modularizzato che soddisfi nel miglior modo i requisiti assegnati; facendo uso anche degli opportuni strumenti di testing e debugging del codice sorgente. ✓ Capacità di apprendere: dimostrando di essere in grado di orientarsi agevolmente nel problem-solving e nella scelta nonché utilizzo delle opportune tecnologie per lo sviluppo di algoritmi e per la loro traduzione in programmi per computer e nella eventuale risoluzione di bug. ✓ Abilità comunicative: dimostrando di essere in grado di illustrare ed argomentare in modo appropriato e puntuale le diverse caratteristiche tecniche e i limiti degli strumenti e delle metodologie informatiche relative allo sviluppo di programmi di media complessità. ✓ Autonomia di giudizio: dimostrando di aver acquisito una buona capacità di applicazione della corretta strategia risolutiva con conseguente capacità di progettazione, individuazione della tecnica migliore e valutazione delle soluzioni e degli algoritmi scritti sia in prima persona che sviluppati da terzi.
<p>Contenuti di insegnamento</p>	<p>Modulo 1: Stili di programmazione</p> <ul style="list-style-type: none"> • Riepilogo dei concetti base di programmazione • Introduzione ai diversi Stili di programmazione.

	<ul style="list-style-type: none"> • Scrittura appropriata del codice • Convenzioni <p>Modulo 2: Approfondimenti Linguaggio C</p> <ul style="list-style-type: none"> • Strutture di controllo • Tipi di dati • Tipi strutturati (Array, Stringhe) • Strutture e gestione dinamica della memoria (Puntatori) • Files <p>Modulo 3: Progettazione modulare</p> <ul style="list-style-type: none"> • Modularizzazione • Strutturazione dei programmi • Procedure e Funzioni <p>Modulo 4: Algoritmi fondamentali e Applicazioni</p> <ul style="list-style-type: none"> • Algoritmi di Ricerca • Algoritmi di Ordinamento <p>Modulo 5: Testing, Debugging, Documentazione</p> <ul style="list-style-type: none"> • Generalità sul Testing • Tecniche di Testing • Tecniche di Debugging • Documentazione del codice
Programma	
Testi di riferimento	H. M. Deitel, P. J. Deitel, "Il linguaggio C. Fondamenti e tecniche di programmazione", Pearson, 2016 (ISBN-10 : 8891901652, ISBN-13 :978-8891901651)
Note ai testi di riferimento	Il testo di riferimento sarà integrato dal materiale fornito dalla docente attraverso le dispense proiettate a lezione, le risoluzioni delle esercitazioni svolte e fornendo tracce d'esempio simili al caso di studio finale.
Metodi didattici	Lezioni frontali (24 ore) intervallate da esercitazioni in aula (45 ore)
Metodi di valutazione (indicare almeno la tipologia scritto, orale, altro)	L'esame prevederà la risoluzione di un caso di studio attraverso la progettazione ed implementazione della soluzione nella quale dimostrare la corretta comprensione dei concetti illustrati durante il corso. In sede d'orale, si discuterà tale prodotto che dovrà evidenziare la capacità di produrre codice corretto, leggibile, ben documentato, correttamente progettato e strutturato nonché esente da bug.

<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>	<p>Nell'ambito delle prove d'esame lo studente dovrà essere in grado di:</p> <ul style="list-style-type: none"> • C1 - mostrare la capacità di problem-solving acquisita attraverso la progettazione della soluzione conforme ai requisiti di progetto; • C2 - scrivere programmi che garantiscano un'adeguata correttezza e aderenza ai requisiti richiesti; • C3 - applicare le migliori metodologie di progettazione modulare per ottimizzare la struttura del codice sorgente; • C4 - scrivere correttamente una documentazione di progetto; • C5 - scrivere del codice sorgente che risponda ai dettami del corretto stile di programmazione, garantendo leggibilità e semplicità; • C6 - dimostrare di saper applicare correttamente le metodologie di testing del codice sorgente ed eventualmente adottare strumenti di debugging. <p>Matrice di Valutazione</p> <table border="1" data-bbox="699 1010 1433 1693"> <thead> <tr> <th>Criteri</th> <th>Correttezza elaborato 50%</th> <th>Leggibilità e Doc 20%</th> <th>Progettazione e contributo personale 30%</th> </tr> </thead> <tbody> <tr> <td>C1</td> <td></td> <td></td> <td>X</td> </tr> <tr> <td>C2</td> <td>X</td> <td></td> <td></td> </tr> <tr> <td>C3</td> <td>X</td> <td></td> <td></td> </tr> <tr> <td>C4</td> <td></td> <td>X</td> <td></td> </tr> <tr> <td>C5</td> <td></td> <td>X</td> <td></td> </tr> <tr> <td>C6</td> <td></td> <td></td> <td>X</td> </tr> </tbody> </table> <p>La valutazione della prova avverrà attraverso i criteri riportati nella succitata matrice e sarà espressa in trentesimi. Il voto minimo per ritenere la prova superata è 18/30.</p>	Criteri	Correttezza elaborato 50%	Leggibilità e Doc 20%	Progettazione e contributo personale 30%	C1			X	C2	X			C3	X			C4		X		C5		X		C6			X
Criteri	Correttezza elaborato 50%	Leggibilità e Doc 20%	Progettazione e contributo personale 30%																										
C1			X																										
C2	X																												
C3	X																												
C4		X																											
C5		X																											
C6			X																										
<p>Altro</p>	<p>Sebbene le risoluzioni delle esercitazioni siano sempre fornite e argomentate in classe, si invitano gli studenti a svolgere gli esercizi in autonomia anche in maniera asincrona, al fine di sviluppare e consolidare le proprie capacità di analisi e giudizio.</p>																												