

Principali informazioni sull'insegnamento	
Titolo insegnamento	Laboratorio di Informatica
Corso di studio	Informatica e Comunicazione Digitale - Taranto
Crediti formativi	6
Denominazione inglese	Computer Science Laboratory
Obbligo di frequenza	No, ma la frequenza è fortemente consigliata
Lingua di erogazione	Italiano

Docente responsabile	Nome Cognome	Indirizzo Mail
	Fedelucio Narducci	fedelucio.narducci@uniba.it
Luogo ed Orario di Ricevimento	7° piano Stanza 762	Venerdì ore 15.00 – 17.00

Dettaglio credi formativi	Ambito disciplinare	SSD	Crediti
	Informatico	ING-INF/051	6

Modalità di erogazione	
Periodo di erogazione	Secondo semestre
Anno di corso	Primo
Modalità di erogazione	Lezioni frontali Esercitazioni in aula

Organizzazione della didattica	
Ore totali	170
Ore di corso	69
Ore di studio individuale	81

Calendario	
Inizio attività didattiche	26/02/2018
Fine attività didattiche	01/06/2018

Syllabus	
Prerequisiti	Propedeuticità formali: nessuna Propedeuticità culturali: programmazione (indispensabile), architettura degli elaboratori (importante)
Risultati di apprendimento previsti (declinare rispetto ai Descrittori di Dublino) (si raccomanda che siano coerenti con i risultati di apprendimento del CdS, compreso i risultati di apprendimento trasversali)	<ul style="list-style-type: none"> • <i>Conoscenza e capacità di comprensione</i> Lo studente sarà in grado di comprendere: <ul style="list-style-type: none"> - i principi di programmazione modulare e di progettazione modulare - le principali tecniche di testing e debugging del codice - le principali metodologie di ricerca e ordinamento dei dati - • <i>Conoscenza e capacità di comprensione applicate</i> Lo studente sarà in grado di:

	<ul style="list-style-type: none"> - Scrivere del codice sorgente efficiente, corretto e adeguatamente modularizzato; - Utilizzare gli strumenti di debugging del codice sorgente; - Utilizzare le principali metodologie di testing del codice sorgente; - Utilizzare gli strumenti per la generazione automatica della documentazione; <ul style="list-style-type: none"> • <i>Autonomia di giudizio</i> Lo studente sarà in grado di: <ul style="list-style-type: none"> - comprendere le caratteristiche del problema da risolvere e applicare la corretta strategia risolutiva; - valutare la correttezza del codice sorgente scritto in un generico linguaggio di programmazione; - comprendere la natura dei messaggi di errore prodotti dal compilatore, ed applicare gli adeguati correttivi; - individuare l'algoritmo di ricerca o di ordinamento più adatto al problema da risolvere. • <i>Abilità comunicative</i> Lo studente sarà in grado di: <ul style="list-style-type: none"> - Illustrare le modalità di analisi del codice sorgente, finalizzata all'individuazione dei bug; - Illustrare le principali tecniche di debugging; - Spiegare i principali principi di progettazione modulare del codice sorgente; - Descrivere le caratteristiche degli strumenti per la generazione automatica della documentazione; - Descrivere le peculiarità di ciascuno degli algoritmi di ricerca e ordinamento a stato dell'arte. • <i>Capacità di apprendere</i> <ul style="list-style-type: none"> - Lo studente apprenderà la capacità di scrivere del codice sorgente corretto, ottimizzato, pulito e ben documentato, e ad applicare le metodologie adeguate per l'individuazione e la risoluzione di bug.
<p>Contenuti di insegnamento</p>	<p>MODULO 1 – Concetti Base di Programmazione</p> <ul style="list-style-type: none"> - Riepilogo concetti base di Programmazione - Linguaggio C e utilizzo degli IDE per la creazione di progetti <p>MODULO 2 – 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>MODULO 3 – Concetti Avanzati di Programmazione</p> <ul style="list-style-type: none"> - Utilizzo dei Puntatori - Utilizzo dei File - Cenni all'Utilizzo della Memoria Dinamica <p>MODULO 4 – Stili di Programmazione</p>

	<ul style="list-style-type: none"> - Uso appropriato dei nomi - Scrittura appropriata di istruzioni ed espressioni - Consistenza ed espressioni idiomatiche - Commenti - Convenzioni di Programmazione <p>MODULO 5 – Testing e Debugging</p> <ul style="list-style-type: none"> - Bug - Tecniche di debugging - Strumenti per il debugging - Generalità sul testing - Il test di unità - Tecniche di testing <p>MODULO 6 – Documentazione del Codice</p> <ul style="list-style-type: none"> - Generalità sulla documentazione del codice in linea - Software Documentation Sheet <p>MODULO 7 – Algoritmi Fondamentali</p> <ul style="list-style-type: none"> - Algoritmi di Ricerca - Algoritmi di Ordinamento di Base - Algoritmi di Ordinamento Avanzati - Cenni alla Complessità Computazionale
--	---

Programma	
Testi di riferimento	D. Chieri, B. Kernighan. Il linguaggio C: Corso di Programmazione, Prentice Hall - Pearson (2002)
Note ai testi di riferimento	Le trasparenze mostrate a lezione, dispense integrative e tracce di esempio delle prove d'esame sono rese disponibili online dal docente
Metodi didattici	Lezioni frontali: 24 ore Esercitazioni in aula: 45 ore
Metodi di valutazione (indicare almeno la tipologia scritto, orale, altro)	<p>1) Appelli d'esame (al termine dell'insegnamento) - L'esame prevede la consegna di un elaborato che mostri la corretta comprensione dei concetti illustrati durante il corso. L'elaborato dovrà dimostrare le capacità di produrre codice corretto, leggibile, ben documentato, correttamente progettato e strutturato ed esente da bug.</p> <p>2) 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.</p>
Criteri di valutazione (per ogni risultato di apprendimento atteso su indicato,	Nell'ambito delle prove d'esame lo studente dovrà essere in grado di:

<p>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>	<ul style="list-style-type: none"> - Scrivere programmi che garantiscano un'adeguata correttezza e aderenza ai requisiti di progetto; - Applicare le metodologie di progettazione modulare per ottimizzare la struttura del codice sorgente; - Scrivere correttamente una documentazione di progetto; - Scrivere del codice sorgente che risponda ai dettami del corretto stile di programmazione, garantendo leggibilità e semplicità; - Saper applicare correttamente le metodologie di testing del codice sorgente ed eventualmente adottare strumenti di debugging. <p>La valutazione della prova è espressa in trentesimi. La prova intermedia è superata con un minimo di 16/30, conseguito svolgendo correttamente almeno uno dei quesiti posti. La prova d'appello è superata con un minimo di 18/30.</p> <p>Per gli studenti non esonerati, l'esame si concluderà con una discussione orale dell'elaborato consegnato.</p> <p>La determinazione del voto tiene conto dei seguenti aspetti:</p> <ol style="list-style-type: none"> 1) la correttezza della soluzione; 2) la capacità di progettare correttamente il codice sorgente; 3) la capacità di scrivere del codice sorgente pulito e leggibile; 4) la capacità di produrre appropriata documentazione; 5) la capacità di applicare le metodologie di testing e debugging; <p>Per superare la prova d'esame o la prova intermedia, lo studente deve essere in grado di proporre una soluzione che soddisfi almeno gli aspetti (1) e (2). I voti superiori al minimo vengono attribuiti agli studenti la cui soluzione soddisfi anche gli altri aspetti.</p>
<p>Altro</p>	<p>Si suggerisce la partecipazione alle lezioni di esercitazione, che prevedono lo svolgimento in aula di esercizi analoghi a quelli previsti per le prove d'esame.</p> <p>Per gli studenti non frequentanti, si suggerisce di svolgere a casa le esercitazioni messe a disposizione online, e di verificare la correttezza delle soluzioni proposte.</p>