

| Principali informazioni sull'insegnamento | |
|--|----------------------------|
| Titolo insegnamento | Laboratorio di Informatica |
| Corso di studio | Informatica |
| Crediti formativi | 3+3 |
| Denominazione inglese | Computer Science Lab |
| Obbligo di frequenza | No |
| Lingua di erogazione | Italiano |

| Docente responsabile | Nome Cognome | Indirizzo Mail |
|-------------------------------|--------------------------|-----------------------------|
| | Fabio Abbattista | Fabio.abbattista@uniba.it |
| Luogo e orario di ricevimento | Dipartimento Informatica | Martedì dalle 16:00 – 17:00 |

| Dettaglio credi formativi | Ambito disciplinare | SSD | Crediti |
|----------------------------------|---|--------------------|---------|
| | Informatico, fisico, matematico, economico, linguistico | INF/01-Informatica | 3+3 |

| Modalità di erogazione | |
|-------------------------------|--|
| Periodo di erogazione | II° semestre |
| Anno di corso | I° |
| Modalità di erogazione | Lezioni frontali, esercitazioni, laboratorio |

| Organizzazione della didattica | |
|---------------------------------------|--------------------------------------|
| Ore totali | 69 (corso) + 81 (studio individuale) |
| Ore di corso | 24+45 |
| Ore di studio individuale | 51+30 |

| Calendario | |
|----------------------------|------------------|
| Inizio attività didattiche | 26 febbraio 2018 |
| Fine attività didattiche | 1 giugno 2018 |

| Syllabus | |
|-------------------------------------|--|
| Prerequisiti | |
| Risultati di apprendimento previsti | <ul style="list-style-type: none"> • <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. • <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 mediante testing; - Capacità di problem-solving attraverso l'applicazione di nozioni apprese nelle discipline informatiche di base nella pratica della programmazione. • <i>Autonomia di giudizio</i> |

| | |
|----------------------------------|--|
| | <p>Lo studente deve dimostrare di aver acquisito autonomia di giudizio e capacità di valutazione degli algoritmi sviluppati da lui o da terzi.</p> <ul style="list-style-type: none"> • <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'. • <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>Contenuti di insegnamento</p> | <ol style="list-style-type: none"> 1. Stili di programmazione Motivazioni Uso appropriato dei nomi Scrittura appropriata di espressioni e istruzioni Consistenza ed espressioni idiomatiche Commenti Convenzioni di programmazione 2. Testing e Debugging Bug Tecniche di debugging Strumenti per il debugging Generalità sul testing Il test di unità Tecniche di testing cUnit 3. Programmazione modulare Modularizzazione e strutturazione dei programmi Strutturazione dei file sorgente Strutturazione di progetti in Eclipse CDT 4. Documentazione del codice Generalità sulla documentazione di codice in linea Documentazione automatica di codice Doxygen 5. Puntatori in C Puntatori Procedure e funzioni I/O Memoria dinamica (cenni) |

| | |
|--|---------------------------|
| | 6. Algoritmi fondamentali |
|--|---------------------------|

| | |
|------------------------------|--|
| Programma | |
| Testi di riferimento | P. Deitel e H. Deitel, Il Linguaggio C - Fondamenti e tecniche di programmazione, Pearson, 2013 |
| Note ai testi di riferimento | Testi integrativi: B.W. Kernighan e R. Pike, Programmazione nella pratica, Addison-Wesley, 1999. J.R. Hanly, E.B. Koffman, Problem solving e programmazione in C, Apogeo, 2013 |
| Metodi didattici | Lezioni frontali ed esercitazioni pratiche in laboratorio |
| Metodi di valutazione | 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. Prova di laboratorio e orale. |
| Criteri di valutazione | Lo studente dovrà dimostrare di aver acquisito la capacità di progettare e realizzare soluzioni ottimali per lo sviluppo di sistemi sw di media complessità, ben documentati e correttamente testati. |
| Altro | |