

<b>Insegnamento di:</b> Informatica				
<b>Classe di laurea:</b> L-35 – Scienze Matematiche		<b>Corso di Laurea in:</b> Matematica	<b>Anno accademico:</b> 2020/2021	
<b>Denominazione inglese insegnamento:</b> Computer Science		<b>Tipo di insegnamento:</b> Obbligatorio	<b>Anno:</b> 1	<b>Semestre:</b> 1
<b>Tipo attività formativa:</b> a – Attività di base	<b>Ambito disciplinare:</b> Formazione Informatica	<b>Settore scientifico-disciplinare:</b> INF/01	<b>CFU totali:</b> 6 di cui CFU lezioni: 4 CFU ese/lab/tutor: 2	
<b>Modalità di erogazione, ore di didattica assistita ed ore dedicate allo studio individuale</b> ore di lezione: 32                                      ore di esercitazione/laboratorio/tutorato: 16 totale ore didattica assistita: 48 totale ore di studio individuale: 102				
<b>Lingua di erogazione:</b> Italiano	<b>Obbligo di frequenza:</b> no			
<b>Docente:</b> Angelo Cardellicchio	<b>Tel:</b> +39 080 596 3312 <b>e-mail:</b> angelo.cardellicchio@uniba.it	<b>Ricevimento studenti:</b> Politecnico di Bari, DEI, Ex – Architettura, stanza 4.1	<b>Giorni e ore ricevimento:</b> Lunedì e mercoledì dalle 16:30 alle 17:30 per appuntamento	
<b>Conoscenze preliminari:</b> Nozioni elementari di matematica fornite dalle scuole secondarie superiori				
<b>Obiettivi formativi:</b> Acquisire concetti base riguardanti le architetture dei calcolatori. Acquisire concetti base riguardanti metodi e tecniche di programmazione. Applicare questi concetti per risolvere problemi per mezzo dei linguaggi C, C++ e Python. Nozioni e concetti base riguardanti algoritmi e complessità computazionale.				
<b>Risultati di apprendimento previsti</b>	<b>Conoscenza e capacità di comprensione:</b> Apprendere concetti fondamentali riguardanti metodi e tecniche di programmazione.			
	<b>Conoscenza e capacità di comprensione applicate:</b> Abilità di applicare i concetti appresi per risolvere ed implementare le soluzioni a problemi di varia natura.			
	<b>Autonomia di giudizio:</b> Capacità di giudicare la consistenza della struttura logica usata per la costruzione degli algoritmi. Capacità di identificare gli strumenti appropriati alla risoluzione di problemi di programmazione.			
	<b>Abilità comunicative:</b> Apprendere i metodi ed i linguaggi dell’informatica per comprendere testi, comunicare la conoscenza appresa, descrivere, analizzare e risolvere i problemi algoritmici.			
	<b>Capacità di apprendere:</b> Acquisire dei metodi di apprendimento duraturi, supportati da ricerche contestuali e risolvendo esercizi e domande fatte periodicamente durante il corso.			
<b>Programma del corso</b> Introduzione al corso  Introduzione all’architettura dei calcolatori ed alle reti  Introduzione alla programmazione Diagrammi di flusso Programmazione strutturata Costrutti di programmazione Strutture dati Variabili e funzioni  Introduzione alla computabilità				

Complessità computazionale  
Macchine di Turing  
Funzioni ricorsive

Introduzione al C

Introduzione al C++

Introduzione al Python

**Metodi di insegnamento:**

Insegnamento in classe ed esercitazioni

**Supporti alla didattica:**

Il materiale di supporto è disponibile al seguente URL: <https://anhelus.github.io/informatica-dm-uniba/>

**Controllo dell'apprendimento e modalità d'esame:**

Progetto d'anno.

Esami con esercizi scritti e verifica orale.

**Testi di riferimento principali:**

Andrew J. Tanenbaum, Architettura dei calcolatori. Un approccio strutturale.

Andrew J. Tanenbaum, Reti di calcolatori.

Deitel, Deitel, C++. Fondamenti di programmazione.

Slide ed altro materiale mostrato a lezione.