



CORSO DI STUDIO *Patrimonio Digitale. Musei, Archivi, Biblioteche*

ANNO ACCADEMICO 2024-2025

DENOMINAZIONE DELL'INSEGNAMENTO *Pensiero computazionale e programmazione – Computational thinking and programming (6 CFU). Modulo di un corso integrato che prevede anche Modelli dei dati (6 CFU)*

Principali informazioni sull'insegnamento	
Anno di corso	I anno
Periodo di erogazione	Il semestre (ultima settimana di febbraio 2025-metà maggio 2025)
Crediti formativi universitari (CFU/ETCS):	6 CFU
SSD	INF/01
Lingua di erogazione	Lingua italiana
Modalità di frequenza	La frequenza non è obbligatoria, ma è fortemente consigliata

Docente	
Nome e cognome	Ciro Castiello
Indirizzo mail	ciro.castiello@uniba.it
Telefono	080 5442135
Sede	Dipartimento di Informatica
Sede virtuale	Piattaforma di e-learning UNIBA - https://elearning.uniba.it/
Ricevimento	Il docente si trattiene in aula al termine delle lezioni per consentire il ricevimento. È comunque possibile concordare incontri di ricevimento (eventualmente anche online) mediante e-mail.

Organizzazione della didattica			
Ore			
Totali	Didattica frontale	Pratica (laboratorio, campo, esercitazione, altro)	Studio individuale
150	42		108
CFU/ETCS			
6	6		

Obiettivi formativi	<ul style="list-style-type: none">- Acquisire le competenze necessarie per affrontare e risolvere problemi computazionali- Sviluppare le capacità di analisi, di astrazione, di decomposizione di problemi complessi in problemi più semplici- Conoscere i fondamenti storici e teorici relativi ad alcuni aspetti rilevanti nell'ambito dello studio dell'informatica- Comprendere i concetti alla base della modellizzazione degli algoritmi e delle principali strutture dati- Apprendere i rudimenti di un linguaggio di programmazione al fine di utilizzarlo per l'implementazione di algoritmi
Prerequisiti	Non vi sono prerequisiti specifici differenti da quelli richiesti per l'accesso al corso di laurea.



Metodi didattici	<ul style="list-style-type: none">- Lezioni frontali con esperienze didattiche partecipative.- Esercitazioni di programmazione guidate.
Risultati di apprendimento previsti <i>Da indicare per ciascun Descrittore di Dublino (DD=</i> DD1 Conoscenza e capacità di comprensione DD2 Conoscenza e capacità di comprensione applicate DD3-5 Competenze trasversali	<ul style="list-style-type: none">• Conoscenza e capacità di comprensione<ul style="list-style-type: none">- Acquisizione e inquadramento nel contesto storico e teorico dei principi generali relativi alla risoluzione efficiente dei problemi mediante strumenti computazionali e meccanismi di elaborazione delle informazioni.- Sviluppo del pensiero computazionale.- Acquisizione dei principi di base della programmazione.• Conoscenza e capacità di comprensione applicate<ul style="list-style-type: none">- Comprensione e capacità di utilizzo delle principali strutture dati per l'organizzazione delle informazioni.- Capacità di sviluppo di algoritmi destinati alla risoluzione di problemi che risultino interessanti da un punto di vista computazionale- Capacità di implementare i suddetti algoritmi in uno specifico linguaggio di programmazione.• Autonomia di giudizio<ul style="list-style-type: none">- Gli/le studenti devono essere in grado di orientarsi fra i principi generali del pensiero computazionale al fine di applicare le tecniche per lo sviluppo e l'implementazione di algoritmi.- Gli/le studenti devono sapere riconoscere e applicare le strutture dati opportune da utilizzare nella gestione di problemi computazionali.- Gli/le studenti devono essere in grado di testare la correttezza dei modelli di soluzione dei problemi e del codice scritto in un linguaggio di programmazione.• Abilità comunicative<ul style="list-style-type: none">- Gli/le studenti devono essere in grado di esporre (oralmente e in forma scritta) le tematiche incluse nel programma del corso facendo uso opportuno del lessico specifico della disciplina.
Contenuti di insegnamento (Programma)	Introduzione al pensiero computazione <ul style="list-style-type: none">- Noam Chomsky- Il processo di astrazione- Apprendimento per trial-and-error e riutilizzo dell'esperienza- Grammatiche, computer e linguaggi Algoritmi <ul style="list-style-type: none">- Ada Lovelace- Definizioni e concetti preliminari- Prime macchine e primi programmatori- Flowchart e strumenti di progettazione degli algoritmi Computabilità <ul style="list-style-type: none">- Alan Turing- La macchina di Turing- Complessità computazionale



	<ul style="list-style-type: none">- I paradossi e il problema della terminazione Linguaggi di programmazione <ul style="list-style-type: none">- Grace Hopper- Introduzione al Python- Valori booleani e operatori logici- Sviluppo guidato dai test Strutture ordinate <ul style="list-style-type: none">- Donald Knuth- Strutture dati- Liste, pile, code Forza bruta <ul style="list-style-type: none">- Betty Holberton- Il problema dell'ordinamento- Il concetto di iterazione- Ricerca lineare e ordinamento a inserimento Strutture non ordinate <ul style="list-style-type: none">- Jorge Luis Borges- La biblioteca di Babele e il concetto di infinito- Insiemi e dizionari Ricorsione <ul style="list-style-type: none">- Douglas Hofstadter- Piccolo labirinto armonico- La ricorsione nel mondo reale e nei programmi- Algoritmi ricorsivi Divide et impera <ul style="list-style-type: none">- John von Neumann- Ordinamento di grandi quantità di elementi- La funzione merge e l'algoritmo merge sort- Analisi di complessità Il linguaggio Python <ul style="list-style-type: none">- Python e Colab- Dai primi passi al concetto di variabile- Sintassi di base- Tipi di dati- Funzioni- Liste e tuple- Iterazioni- Insiemi e dizionari- Ricorsione- Gestione dei file- Esercitazioni guidate Altri argomenti <ul style="list-style-type: none">- Programmazione dinamica- Backtracking- Alberi e grafi- Algoritmi greedy
Testi di riferimento	S. Peroni – Computational Thinking and Programming (disponibile online). Altro materiale eventualmente suggerito o fornito dal docente durante lo svolgimento del corso.
Note ai testi di riferimento	Il testo di riferimento è disponibile gratuitamente.



Materiali didattici	Il materiale didattico personale del docente sarà reso disponibile sul sito web dedicato al corso, ospitato sulla piattaforma di e-learning dell'Università di Bari.
Valutazione	
Modalità di verifica dell'apprendimento	<p>Svolgimento di una prova scritta (da eseguirsi, tipicamente, al computer in laboratorio). Le modalità di verifica mirano ad accertare le seguenti capacità:</p> <ul style="list-style-type: none">- Esposizione (con riferimento agli argomenti teorici facenti parte del programma del corso)- Comprensione (con riferimento alle tecniche e ai modelli studiati, con la proposta di analisi di casi d'esempio ed esercizi)- Sviluppo (con riferimento al linguaggio Python, con la proposta di una descrizione di un problema computazionale per il quale occorre definire, implementare e testare un algoritmo di risoluzione) <p>La prova ha una durata minima di 90 minuti. Non è consentito l'utilizzo di materiale didattico per le attività di Esposizione e Comprensione. È consentita la consultazione di tutto il materiale didattico per l'attività di Sviluppo.</p> <p>La prenotazione sulla piattaforma esse3 (entro i termini indicati) costituisce una condizione necessaria per la partecipazione alla prova. La stessa piattaforma è utilizzata per la pubblicazione dei risultati della prova. I risultati compaiono su esse3 entro pochi giorni dalla conclusione della prova (generalmente non oltre una settimana).</p>
Criteri di valutazione	<p>Con riferimento ai risultati di apprendimento e alle modalità di verifica dell'apprendimento presentati in precedenza, i criteri di valutazione mirano a verificare specificamente:</p> <ul style="list-style-type: none">• Conoscenza e capacità di comprensione: Conoscenza adeguata dei formalismi e degli argomenti teorici facenti parte del programma del corso (attività di Esposizione), dei costrutti e delle funzioni fondamentali del linguaggio Python (attività di Sviluppo) e comprensione dei modelli e delle tecniche trattate durante il corso (attività di Comprensione).• Conoscenza e capacità di comprensione applicate: Efficacia nello sviluppo di soluzioni basate sul pensiero computazionale mediante la definizione e/o l'analisi di algoritmi e l'applicazione di opportune strutture dati (attività di Comprensione e di Sviluppo).• Autonomia di giudizio: Capacità di elaborare e organizzare le idee in modo critico e sistematico (attività di Esposizione), capacità di ragionamento critico sullo studio realizzato (attività di Esposizione e di Comprensione), capacità di analisi delle proprie proposte di risoluzione dei problemi computazionali (attività di Sviluppo).• Abilità comunicative: Competenza nell'impiego del lessico specifico della disciplina con adeguata qualità di argomentazione (attività di Esposizione), capacità di sintesi e di chiarezza espositiva nella presentazione delle proprie proposte di risoluzione dei quesiti (attività di Comprensione e di Sviluppo).
Criteri di misurazione dell'apprendimento e di attribuzione del voto finale	Le diverse attività incluse nella prova di verifica (Esposizione, Comprensione, Sviluppo) sono caratterizzate da un punteggio. Il voto finale deriva dalla somma dei punteggi ottenuti per ciascuna attività.



	<p><i>Il voto finale è espresso in trentesimi. L'esame si intende superato quando il voto è maggiore o uguale a 18.</i></p> <p><i>18-21: valutazione sufficiente</i></p> <p><i>22-24: valutazione discreta</i></p> <p><i>25-27: valutazione buona</i></p> <p><i>28-30 (con eventuale attribuzione della lode): valutazione ottima o eccellente</i></p>
Altro	
	Si consiglia la frequenza delle lezioni e l'interazione col docente.