



Principali informazioni sull'insegnamento	
Denominazione dell'insegnamento	Pensiero computazionale e programmazione
Corso di studio	LM-84 Scienze Storiche e sociali
Anno accademico	2022-2023
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS):	6
SSD	INF/01
Lingua di erogazione	Italiano
Periodo di erogazione	Secondo semestre (27.02.2023 – 19.05.2023)
Obbligo di frequenza	La frequenza è disciplinata dal Regolamento Didattico del Corso che è consultabile al seguente link : https://w3.uniba.it/corsi/scienze-storiche-sociali/iscriversi/RegolamentoLM842223.pdf

Docente	
Nome e cognome	Ciro Castiello
Indirizzo mail	ciro.castiello@uniba.it
Telefono	080 5442135
Sede	Dipartimento di Informatica
Sede virtuale	Microsoft Teams
Ricevimento (giorni, orari e modalità)	Al termine della lezione e/o su appuntamento (il ricevimento può tenersi in presenza oppure online)

Syllabus	
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	Nessuno
Contenuti di insegnamento (Programma)	<p>Introduzione al pensiero computazione</p> <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 <p>Algoritmi</p> <ul style="list-style-type: none">- Ada Lovelace- Definizioni e concetti preliminari- Prime macchine e primi programmatori- Flowchart e strumenti di progettazione degli algoritmi <p>Computabilità</p> <ul style="list-style-type: none">- Alan Turing

	<ul style="list-style-type: none"> - La macchina di Turing - Complessità computazionale - I paradossi e il problema della terminazione <p>Linguaggi di programmazione</p> <ul style="list-style-type: none"> - Grace Hopper - Introduzione al Python - Valori booleani e operatori logici - Sviluppo guidato dai test <p>Strutture ordinate</p> <ul style="list-style-type: none"> - Donald Knuth - Strutture dati - Liste, pile, code <p>Forza bruta</p> <ul style="list-style-type: none"> - Betty Holberton - Il problema dell'ordinamento - Il concetto di iterazione - Ricerca lineare e ordinamento a inserimento <p>Strutture non ordinate</p> <ul style="list-style-type: none"> - Jorge Luis Borges - La biblioteca di Babele e il concetto di infinito - Insiemi e dizionari <p>Ricorsione</p> <ul style="list-style-type: none"> - Douglas Hofstadter - Piccolo labirinto armonico - La ricorsione nel mondo reale e nei programmi - Algoritmi ricorsivi <p>Divide et impera</p> <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 <p>Il linguaggio Python</p> <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 <p>Altri argomenti</p> <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	

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

Metodi didattici	
	Lezioni frontali con esperienze didattiche partecipative. Esercitazioni di programmazione guidate.

Risultati di apprendimento previsti	
Conoscenza e capacità di comprensione	Acquisizione dei principi generali, opportunamente presentati nel contesto storico e teorico, relativi alla risoluzione efficiente dei problemi mediante l'utilizzo di strumenti computazionali e meccanismi di elaborazione delle informazioni.
Conoscenza e capacità di comprensione applicate	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 visto computazionale, con la possibilità di implementare i suddetti algoritmi in uno specifico linguaggio di programmazione.
Competenze trasversali	<ul style="list-style-type: none"> • <i>Autonomia di giudizio</i> Gli studenti sono in grado di apprezzare i principi generali del pensiero computazionale e le tecniche per lo sviluppo e l'implementazione di algoritmi. L'autonomia di giudizio viene acquisita anche attraverso lo studio e l'interpretazione critica dei testi. Il raggiungimento dell'adeguata autonomia è verificato attraverso l'esame finale di profitto e l'eventuale svolgimento di esercitazioni durante il corso. • <i>Abilità comunicative</i> Gli studenti sono in grado di esporre le tematiche incluse nel programma del corso mediante il lessico specifico della disciplina. • <i>Capacità di apprendere in modo autonomo</i> Gli studenti sono in grado di approfondire in autonomia le tematiche incluse nel programma del corso anche ricorrendo a risorse non direttamente coinvolte nella erogazione delle ore di lezione.

Valutazione	
Modalità di verifica dell'apprendimento	Svolgimento di una prova scritta che includa: - esposizione degli argomenti teorici facenti parte del programma; - attività di verifica della comprensione degli argomenti trattati a lezione; - attività di sviluppo (con particolare riferimento al linguaggio Python) Eventuale discussione orale.
Criteri di valutazione	La prova d'esame mira a verificare che lo studente sia in grado di dimostrare: - la conoscenza delle tematiche trattate durante le lezioni; - la capacità di applicare le conoscenze apprese a problemi inerenti al contesto informatico; - la capacità di analisi e di comparazione fra soluzioni diverse e/o alternative; - la capacità di esporre concetti complessi mediante terminologia e formalismo appropriati;

	- la capacità di elaborare e organizzare le idee in modo critico e sistematico.
Criteria di misurazione dell'apprendimento e di attribuzione del voto finale	Le diverse attività incluse nella prova di verifica sono caratterizzate da un punteggio. Il voto finale deriva dalla somma dei punteggi ottenuti per ciascuna attività.
Altro	
	Si consiglia la frequenza delle lezioni e l'interazione col docente.