UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

DIPARTIMENTO DI RICERCA E INNOVAZIONE UMANISTICA

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

ANNO ACCADEMICO *2023-2024*

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	*II semestre (26/02/24-17/05/24)*
Crediti formativi universitari (CFU/ETCS):	*6 CFU*
SSD	*INF/01 – Informatica*
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	
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	*48*		*102*
CFU/ETCS			
6	*6*		

Obiettivi formativi	- *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	*- Lezioni frontali con esperienze didattiche partecipative.* *- Esercitazioni di programmazione guidate.*

Risultati di apprendimento previsti ***Da indicare per ciascun Descrittore di Dublino (DD=*** ***DD1* Conoscenza e capacità di comprensione** ***DD2* Conoscenza e capacità di comprensione applicate** ***DD3-5* Competenze trasversali**	• *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.* • *Capacità di applicare conoscenza e comprensione* *- 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.* • *Autonomia di giudizio* *Gli studenti devono essere in grado di orientarsi fra i principi generali del pensiero computazionale e le tecniche per lo sviluppo e l'implementazione di algoritmi. L'autonomia di giudizio viene acquista anche attraverso lo studio e l'interpretazione critica del materiale didattico. Il raggiungimento dell'adeguata autonomia è verificato attraverso l'esame finale di profitto e lo svolgimento di esercitazioni durante il corso.* • *Abilità comunicative* *Gli studenti devono essere in grado di esporre (oralmente o in forma scritta) le tematiche incluse nel programma del corso mediante il lessico specifico della disciplina.* • *Capacità di apprendere in modo autonomo* *Gli studenti devono essere 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.*
Contenuti di insegnamento (Programma)	*Introduzione al pensiero computazione* *- Noam Chomsky* *- Il processo di astrazione* *- Apprendimento per trial-and-error e riutilizzo dell'esperienza* *- Grammatiche, computer e linguaggi* *Algoritmi* *- Ada Lovelace* *- Definizioni e concetti preliminari* *- Prime macchine e primi programmatori* *- Flowchart e strumenti di progettazione degli algoritmi* *Computabilità* *- Alan Turing* *- La macchina di Turing* *- Complessità computazionale*

	- *I paradossi e il problema della terminazione* *Linguaggi di programmazione* - *Grace Hopper* - *Introduzione al Python* - *Valori booleani e operatori logici* - *Sviluppo guidato dai test* *Strutture ordinate* - *Donald Knuth* - *Strutture dati* - *Liste, pile, code* *Forza bruta* - *Betty Holberton* - *Il problema dell'ordinamento* - *Il concetto di iterazione* - *Ricerca lineare e ordinamento a inserimento* *Strutture non ordinate* - *Jorge Luis Borges* - *La biblioteca di Babele e il concetto di infinito* - *Insiemi e dizionari* *Ricorsione* - *Douglas Hofstadter* - *Piccolo labirinto armonico* - *La ricorsione nel mondo reale e nei programmi* - *Algoritmi ricorsivi* *Divide et impera* - *John von Neumann* - *Ordinamento di grandi quantità di elementi* - *La funzione merge e l'algoritmo merge sort* - *Analisi di complessità* *Il linguaggio Python* - *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* - *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	

Materiali didattici	*Il materiale didattico sarà reso disponibile dal docente sul sito web dedicato al corso, ospitato sulla piattaforma di e-learning del Dipartimento di Informatica.*

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*
Criteri 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.* *Il voto finale è attribuito in trentesimi. L'esame si intende superato quando il voto è maggiore o uguale a 18.*
Altro	
	Si consiglia la frequenza delle lezioni e l'interazione col docente.