UNIVERSITÀ DEGLI STUDI DI ENNA "KORE"

Facoltà di Ingegneria e Architettura
Anno Accademico 2021/2022
Corso di studi in Ingegneria Informatica, classe di laurea L8

Insegnamento	Algoritmi e Strutture Dati
CFU	6
Settore Scientifico Disciplinare	ING-INF/05
Metodologia didattica	Lezioni frontali
Nr. ore di aula	36
Nr. ore di studio autonomo	114
Nr. ore di laboratorio	-
Mutuazione	No
Annualità	III anno
Periodo di svolgimento	II semestre

Docente	E-mail	Ruolo[i]	SSD docente
Giovanni Pau	giovanni.pau@unikore.it	PA	ING-INF/05

Propedeuticità	Nessuna
Sede delle lezioni	Facoltà di Ingegneria e Architettura

Moduli

No.

Orario delle lezioni

L'orario delle lezioni sarà pubblicato sulla pagina web del corso di laurea.

https://unikore.it/index.php/it/ingegneria-informatica-attivita-didattiche/calendario-lezioni

Obiettivi formativi

L'obiettivo del corso è fornire agli studenti gli strumenti teorici e pratici, attraverso il linguaggio Python, per affrontare la progettazione di soluzioni algoritmiche corrette ed efficienti, tramite l'uso di opportune strutture dati, per un'ampia gamma di problemi computazionali.

Contenuti del Programma

N.	Argomento	Tipologia	Durata in ore
1	• Caratteristiche di algoritmi e strutture dati • Python: operazioni preliminari, IDLE e shell, interprete	Frontale	3
2	• Python: commenti, valutazione espressioni, operazioni aritmetiche, variabili e operatori, funzioni predefinite, tipi di dato, stile di programmazione • Esempi ed esercizi	Frontale	3
3	• Python: sequenze e flow chart, controllo e selezione, blocchi di codice e indentazione, operatori logici • Esempi ed esercizi	Frontale	4
4	• Python: moduli, funzioni e numeri casuali, gestione variabili locali e globali, ambiente grafico turtle e	Frontale	4

	relativi metodi, cicli • Esempi ed esercizi		
5	• Python: stringhe, liste, tuple, insiemi e dizionari • Esempi ed esercizi	Frontale	3
6	• Python: errori, test e debugging, file, dati e statistica • Esempi ed esercizi	Frontale	3
7	• Analisi asintotica di algoritmi, strutture dati fondamentali	Frontale	4
8	• Pile, code e code doppie, liste concatenate, alberi, code prioritarie, mappe, tabelle di hash, alberi di ricerca	Frontale	8
9	• Ordinamento e selezione, algoritmi per grafi	Frontale	4

Risultati di apprendimento (descrittori di Dublino)

I risultati di apprendimento attesi sono definiti secondo i parametri europei descritti dai cinque descrittori di Dublino.

1. **Conoscenza e capacità di comprensione**: lo studente alla fine del corso acquisirà una buona conoscenza di algoritmi e delle più importanti strutture dati utilizzate nella programmazione avanzata. Inoltre, sarà in grado di analizzare e comprendere il codice sorgente in linguaggio Python di diversi algoritmi utilizzati per lo sviluppo del software.
2. **Conoscenza e capacità di comprensione applicate**: lo studente sarà in grado di valutare le caratteristiche, i vantaggi e le limitazioni di diversi algoritmi e strutture dati. Inoltre, attraverso il linguaggio Python, sarà in grado non solo di progettare, analizzare e valutare le soluzioni software a problemi di media complessità ma anche di sviluppare nuove soluzioni software, valutandone la qualità in termini di semplicità, efficacia ed efficienza.
3. **Autonomia di giudizio**: lo studente sarà in grado di effettuare l'analisi sulle problematiche relative alla progettazione software. Inoltre, a partire da precise specifiche, sarà in grado di implementare opportune soluzioni software mediante il linguaggio Python, valutandone la qualità in termini di semplicità, leggibilità, efficienza e possibilità di riutilizzo.
4. **Abilità comunicative**: lo studente acquisirà la capacità di comunicare ed esprimere problematiche inerenti all'oggetto del corso. Sarà in grado di sostenere conversazioni su tematiche relative alle implementazioni software di algoritmi e strutture dati efficienti. Inoltre, sarà in grado di utilizzare un linguaggio semplice e chiaro per la descrizione dei processi di analisi e di sintesi di soluzioni software a problemi di media complessità.
5. **Capacità di apprendere**: lo studente svilupperà la capacità di apprendere i processi di analisi e di sintesi relativi alla codifica di algoritmi di media complessità ed alla relativa implementazione mediante il linguaggio Python.

Testi per lo studio della disciplina

Maurizio Boscaini, "Imparare a programmare con Python - il manuale per programmatori dai 13 anni in su," Apogeo
https://www.apogeonline.com/libri/imparare-a-programmare-con-python-maurizio-boscaini/

Kenneth A. Lambert, "Programmazione in Python", Maggioli Editore
http://www.apogeoeducation.com/programmazione-in-python.html

Michael T. Goodrich - Roberto Tamassia - Michael H. Goldwasser, "Data structures and Algorithms in Python," Wiley
https://www.amazon.it/Structures-Algorithms-Python-Michael-Goodrich/dp/1118290275

Michael T. Goodrich - Roberto Tamassia - Michael H. Goldwasser, "Algoritmi e strutture dati in Java," Apogeo
http://www.apogeoeducation.com/9788891613394-algoritmi-e-strutture-dati-in-java.html

Alan A. Bertossi, Alberto Montresor, "Algoritmi e strutture di dati," CittàStudi
https://www.libreriauniversitaria.it/algoritmi-strutture-dati-bertossi-alan/libro/9788825173956

Cay Horstmann - Rance D. Necaise, "Concetti di informatica e fondamenti di Python," Apogeo
http://www.apogeoeducation.com/informatica/concetti-di-informatica-e-fondamenti-di-python.html

Le slide proiettate a lezione che non sono protette da copyright sono fornite dal docente titolare dell'insegnamento e messe a disposizione degli studenti sul sito web dell'Università.

Modalità di accertamento delle competenze

L'accertamento delle competenze avverrà attraverso una prova orale interamente basata sulla proposta progettuale dallo studente (o da un gruppo di studenti nel caso di lavoro in gruppo). L'elaborato deve rispettare le direttive, relative all'utilizzo delle strutture dati, che saranno pubblicate dal docente nella sua pagina web. La proposta progettuale è volta a dimostrare l'acquisizione degli argomenti erogati durante le lezioni frontali. La durata del colloquio dipenderà dalla profondità ed ampiezza della proposta progettuale ed indicativamente durerà tra 20 ed i 60 minuti. Ove fosse necessario, gli esaminandi saranno ripartiti in più giornate, secondo un calendario determinato nel giorno dell'appello ovvero, se possibile, anticipatamente sulla base delle prenotazioni pervenute. La calendarizzazione sarà, in tal caso, opportunamente pubblicizzata. La valutazione dell'apprendimento sarà focalizzata sulla valutazione dei risultati attesi, in accordo con i descrittori di Dublino.

La prova di esame si intende superata con una votazione minima di 18/30 quando lo studente dimostra:
- minima conoscenza e comprensione degli argomenti trattati;
- limitata capacità nell'applicazione delle conoscenze acquisite;
- sufficiente capacità espositiva.

La votazione di 30/30, eventualmente con lode, è assegnata quando lo studente dimostra:
- ottima conoscenza e comprensione degli argomenti trattati;
- ottima capacità nell'applicazione delle conoscenze acquisite;
- eccellente capacità espositiva.

La prova di esame si intende non superata se lo studente mostra un livello insufficiente di conoscenza degli argomenti trattati e non dimostra una sufficiente capacità nell'applicazione delle conoscenze acquisite.

Date di esame

Le date di esami saranno pubblicate sulla pagina web del corso di laurea.

https://unikore.it/index.php/it/ingegneria-informatica-esami/calendario-esami

Modalità e orario di ricevimento

Gli orari di ricevimento saranno pubblicati nella pagina personale del docente:

https://unikore.it/index.php/it/ingegneria-informatica-persone/docenti-del-corso/itemlist/category/2395-prof-pau-giovanni

[i] PO (professore ordinario), PA (professore associato), RTD (ricercatore a tempo determinato), RU (Ricercatore a tempo indeterminato), DC (Docente a contratto).