



UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Matematica e Informatica
ANNO ACCADEMICO OFFERTA	2023/2024
ANNO ACCADEMICO EROGAZIONE	2023/2024
CORSO DILAUREA	INTELLIGENZA ARTIFICIALE
INSEGNAMENTO	PROGRAMMAZIONE CON LABORATORIO
TIPO DI ATTIVITA'	A
AMBITO	50168-Formazione informatica di base
CODICE INSEGNAMENTO	16161
SETTORI SCIENTIFICO-DISCIPLINARI	INF/01
DOCENTE RESPONSABILE	SCIORTINO MARINELLA Professore Ordinario Univ. di PALERMO
ALTRI DOCENTI	
CFU	9
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	145
NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA	80
PROPEDEUTICITA'	
MUTUAZIONI	
ANNO DI CORSO	1
PERIODO DELLE LEZIONI	1° semestre
MODALITA' DI FREQUENZA	Facoltativa
TIPO DI VALUTAZIONE	Voto in trentesimi
ORARIO DI RICEVIMENTO DEGLI STUDENTI	SCIORTINO MARINELLA Martedì 13:00 14:00 Ufficio n. 201, Il piano del DMI Venerdì 12:00 13:30 Ufficio n. 201, Il piano del DMI

DOCENTE: Prof.ssa MARINELLA SCIORTINO

PREREQUISITI	Nessuno
RISULTATI DI APPRENDIMENTO ATTESI	Conoscenza e capacità di comprensione: acquisizione dei concetti fondamentali della programmazione strutturata; strutture dati elementari; semplici algoritmi fondamentali di ricerca e di ordinamento; definizione ricorsiva di soluzioni; padronanza dei costrutti fondamentali del linguaggio di programmazione Python; Capacità di applicare conoscenza e comprensione: capacità di misurarsi con semplici problemi computazionali mediante scelta di strutture dati e algoritmi appropriati; capacità di programmazione in linguaggio di programmazione Python; capacità di implementare, mediante la scrittura di semplici programmi, le nozioni e gli algoritmi studiati; capacità di progettare la soluzione algoritmica a problemi computazionalmente semplici; capacità di decomporre problemi complessi in problemi più semplici da un punto di vista computazionale. Autonomia di giudizio: gli studenti sono guidati ad apprendere in maniera critica e responsabile tutto ciò che viene spiegato in aula e ad arricchire le proprie capacità di giudizio attraverso lo studio del materiale didattico indicato o fornito dal docente e attraverso la realizzazione di semplici programmi nei quali è richiesta una scelta consapevole di parametri e funzioni o di strategie risolutive in base all'efficienza di calcolo. Abilità comunicative: Il corso mira a sviluppare le proprietà di espressione nella presentazione delle nozioni di base dei linguaggi di programmazione e della programmazione imperativa. Inoltre, attraverso l'interazione durante le lezioni e le attività di laboratorio previste, il corso tenderà a favorire lo sviluppo della capacità di comunicare in modo chiaro ed esaustivo le proprie idee ed argomentare le proprie conclusioni. Capacità d'apprendimento: Attraverso approfondimenti e consultazione dei testi di riferimento, gli studenti impareranno a sapere contestualizzare le abilità acquisite in problemi concreti e in applicazioni reali. Competenze acquisite: progettare algoritmi e appropriate strutture dati per la risoluzione di problemi semplici; realizzare, verificare e correggere programmi che usano i costrutti fondamentali della programmazione, la gestione dell'input e dell'output, le funzioni e la gestione di parametri; utilizzare le proprie competenze di programmazione per affrontare lo studio di linguaggi di programmazione diversi da quello utilizzato per le attività di laboratorio.
VALUTAZIONE DELL'APPRENDIMENTO	L'esame è costituito da una prova scritta, da una prova pratica e da un breve colloquio orale. La prova scritta è costituita da dieci domande a risposta multipla e a risposta aperta. La prova pratica consiste nello sviluppo di un semplice programma inerente gli argomenti affrontati durante le lezioni e le attività in laboratorio. Il colloquio consiste in ulteriori domande volte a valutare la conoscenza delle principali nozioni di programmazione. La prova pratica/orale è complessivamente valutata in trentesimi. Il voto finale tiene conto dei voti conseguiti nella prova scritta e nella prova pratica/orale. Le fasce di punteggio corrispondono ai giudizi qualitativi seguenti: 18-20: la conoscenza della materia e le competenze di programmazione sono sufficienti; 21-23: la conoscenza della materia e le competenze di programmazione sono discreti; 24-25: la conoscenza della materia e le competenze di programmazione sono buoni; 26-27: la conoscenza della materia e le competenze di programmazione sono molto buoni; 28-30: la conoscenza della materia e le competenze di programmazione sono ottimi. La lode è riservata agli studenti che dimostrano nella prova pratica e nella prova orale eccellenti competenze di programmazione e un'ottima padronanza della materia.
OBIETTIVI FORMATIVI	Il corso introduce gli studenti al concetto di algoritmo e alla programmazione, con l'obiettivo di risolvere semplici problemi computazionali mediante la progettazione e l'implementazione di algoritmi e di strutture dati elementari. Vengono i concetti di base della programmazione utilizzando il linguaggio di programmazione Python attraverso il paradigma imperativo (variabili, espressioni, cicli, funzioni, ricorsione, input / output) e orientato agli oggetti (classi e strutture dati astratte)
ORGANIZZAZIONE DELLA DIDATTICA	La didattica è organizzata mediante lezioni frontali in aula e attività in laboratorio, in cui si prevede sia un trasferimento della conoscenza ma anche, con il supporto del mezzo informatico, la contestuale verifica dell'acquisizione

	delle competenze pratiche richieste. Per incrementare il coinvolgimento degli studenti si utilizzano anche metodologie di collaborative learning.
TESTI CONSIGLIATI	- Introduction to Computation and Programming Using Python, Third Edition With Application to Computational Modeling and Understanding Data By John V. Guttag. The MIT Press - Think Python: How to Think Like a Computer Scientist by Allen B. Downey Beijing: O'reilly Media - Problem Solving With Algorithms and Data Structures Using Python, 2nd edition, By Brad Miller and David Ranum Franklin Beedle & Assoc

PROGRAMMA

ORE	Lezioni
4	Introduzione al concetto di algoritmo e strategia algoritmiche per la risoluzione di problemi. Complessità di un algoritmo.
4	Architettura di un elaboratore secondo Von Neumann. Rappresentazione dell'informazione. Rappresentazione di interi, interi relativi, reali, caratteri, stringhe.
4	Paradigmi di programmazione e linguaggi di programmazione. Compilatori e Interpreti. Struttura di un programma. Esecuzione di un programma. Strutture di controllo di un programma.
6	Struttura di un programma in Python. Tipi di dato, operatori, variabili e assegnazioni. Istruzioni condizionali.
6	Costrutti e programmi iterativi. Equivalenza di costrutti iterativi.
6	Scomposizione di un problema in sottoproblemi. Funzioni. Visibilità di una variabile. Definizione di funzioni in Python
2	File di input e output
6	Ricorsione. Stack delle chiamate di funzioni ricorsive.
6	La ricerca sequenziale. Analisi della ricerca sequenziale. Ricerca binaria. Analisi della ricerca binaria.
8	Ordinamento di elementi atomici. Bubblesort, Selectionsort, InsertionSort, MergeSort. QuickSort e variante randomizzata. Counting Sort.
6	Programmazione orientata agli oggetti. Definizione di classi. Ereditarietà. Incapsulamento. Polimorfismo.
6	Strutture dati astratte. Strutture dati lineari: Pila e Coda
ORE	Laboratori
2	Ambiente di sviluppo per programmi in Python. Numeri, stringhe, variabili, assegnazioni e istruzioni condizionali
2	Programmi iterativi in python. Liste
2	Funzioni in Python
2	Uso di input e output file in Python. Uso di Librerie. Plot di funzioni
2	Programmi ricorsivi e algoritmi di ricerca in Python
2	Algoritmi di ordinamento in Python
2	Implementazione di pila e coda in Python
2	Implementazione di classi e oggetti in Python