# Laboratorio di programmazione

## Prof. Daniele Toti

***OBIETTIVO DEL CORSO E RISULTATI DI APPRENDIMENTO ATTESI***

L’insegnamento si propone di fornire agli studenti le nozioni di base della programmazione da un punto di vista teorico e pratico. Il linguaggio “didattico” scelto è Python (versione 3+), ma durante il corso verranno fatti dei confronti con altri linguaggi di programmazione, in particolare Java.

Saper programmare è una competenza fondamentale, richiesta nella maggior parte delle attività tecnico-scientifiche moderne. In quest’ottica, il corso ha un triplice obiettivo:

* fornire gli strumenti metodologici e tecnologici per permettere a studenti che non hanno mai affrontato un corso di programmazione di sviluppare programmi con una certa autonomia già al termine del corso;
* aiutare gli studenti che possiedono conoscenze basilari sulla programmazione a consolidarle e ad applicarle nel linguaggio Python, fornendo inoltre spunti di modellazione e design;
* trainare gli studenti verso corsi più avanzati, come Programmazione ad Oggetti, gettando le basi per affrontare diversi linguaggi e paradigmi di programmazione.

Al termine dell’insegnamento lo studente avrà acquisito familiarità con le tecniche fondamentali della programmazione e sarà in grado di sviluppare programmi di media complessità nel linguaggio Python; conoscerà il paradigma di programmazione procedurale; avrà un’infarinatura di base sul paradigma orientato agli oggetti e sulla programmazione orientata agli oggetti; sarà in grado di utilizzare alcune librerie aggiuntive per svolgere operazioni più complesse. Più in generale, sarà in grado di cominciare a pensare come un informatico.

***PROGRAMMA DEL CORSO***

* Introduzione alla programmazione
* Introduzione al linguaggio Python
* Sintassi di base, variabili, assegnazioni e tipi primitivi
* Cenni su Input/Output: lettura e scrittura da tastiera
* Ambiente di programmazione e IDE: PyCharm
* Meccanismi fondamentali: commenti, espressioni booleane, istruzioni condizionali, cicli, operatori
* Definizione di funzioni: segnatura, return type, portata delle variabili, variable-length arguments e parametri opzionali
* Funzioni *built-in* di Python
* Utilizzo di librerie
* Cenni metodologici alla scrittura e alla comprensione dei programmi
* Strutture dati fondamentali: stringhe e collezioni
* Operazioni su stringhe
* Operazioni su collezioni: liste, insiemi, tuple e mappe/dizionari
* Gestione di file e file system
* Gestione delle eccezioni
* Paradigma OO e programmazione OO in Python: classi, costruttori, metodi, incapsulamento, ereditarietà, polimorfismo, metodi speciali
* Elementi di progettazione del software: architettura multi-livello e introduzione ai design pattern
* Ricorsione e cenni sull’analisi di performance degli algoritmi
* Cenni sulla gestione delle date e del tempo in Python
* Introduzione a librerie e strutture di dati aggiuntive per il calcolo scientifico e la visualizzazione ed elaborazione di dati: Numpy, Matplotlib e Pandas.

***BIBLIOGRAFIA***

Slide delle lezioni e delle esercitazioni, appunti delle lezioni, soluzioni degli esercizi svolti, risorse online comunicate agli studenti.

Libro di testo consigliato:

* *Thinking Python, Seconda Edizione, traduzione Italiana*

messo a disposizione degli studenti sulla piattaforma Blackboard assieme al resto del materiale. Il corso non seguirà alla lettera tale testo, ma spazierà, integrerà e approfondirà i diversi argomenti e proporrà numerosi esercizi da svolgere in classe, per cui la frequenza è caldamente consigliata.

***DIDATTICA DEL CORSO***

Lezioni di teoria, esercitazioni con discussione delle soluzioni e assegnazione di homework.

***METODO E CRITERI DI VALUTAZIONE***

L’esame consiste in una prova scritta contenente esercizi di programmazione a risposta aperta e domande a risposta chiusa. Nell’esame possono essere proposti problemi da risolvere attraverso lo sviluppo di programmi, porzioni di codice da leggere e interpretare e domande su aspetti teorici degli argomenti affrontati nel corso. A seconda delle possibilità tecniche e logistiche, gli esercizi di programmazione potrebbero essere svolti direttamente su calcolatore.

Gli esercizi di programmazione saranno valutati sulla base della loro correttezza sintattica e semantica, della loro aderenza rispetto al comportamento atteso, della chiarezza, comprensibilità ed eleganza del codice, e del loro funzionamento complessivo.

L’elaborato è personale e non è consentito il lavoro di gruppo. Non è possibile consultare libri o appunti durante lo svolgimento della prova d’esame.

Il docente può richiedere, in casi eccezionali e a suo insindacabile giudizio, una prova orale di verifica supplementare a valle della correzione della prova scritta per determinati candidati. In generale, rientrano nelle possibili casistiche prove scritte di pochissimo al di sotto della sufficienza o situazioni in cui si sospettano potenziali irregolarità commesse durante la prova scritta. In ogni caso, la prova orale non può essere richiesta da parte degli studenti.

Il voto finale dell’esame è dato dal punteggio ottenuto nella prova scritta, a cui si sommano gli eventuali punti bonus ottenuti attraverso lo svolgimento degli homework.

I punti bonus sono assegnati agli studenti che svolgono correttamente gli homework durante il periodo di lezioni e valgono per la prima volta in cui si sostiene l’esame; sono azzerati subito dopo a prescindere dall’esito dell’esame.

***AVVERTENZE E PREREQUISITI***

Il corso non ha nessun prerequisito “formale”. È richiesta tuttavia familiarità nell’uso di un computer e una conoscenza di base degli argomenti illustrati nel corso di Fondamenti dell’Informatica.

*Orario e luogo di orario di ricevimento*

Le modalità di ricevimento studenti sono consultabili alla pagina personale del docente.