

# STUD: una collezione di funzioni per la gestione della carriera degli studenti

Alessandro Bocci, Susanna Pelagatti

Primo assegnamento Python in itinere 622AA  
AA 2023-24

## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Materiale in linea . . . . .	1
1.2	Struttura degli assegnamenti, bonus, tempi di consegna e prova orale . . . . .	1
1.3	Consegna degli assegnamenti . . . . .	2
1.4	Valutazione dell'assegnamento . . . . .	2
<b>2</b>	<b>L'assegnamento STUD</b>	<b>2</b>
2.1	Istruzioni . . . . .	3

## 1 Introduzione

Il primo modulo (9 crediti) del corso di Programmazione e Analisi Dati (622AA) per la parte Python prevede lo svolgimento di due assegnamenti in itinere che esonerano dallo svolgimento del progetto finale. Questo documento descrive il primo assegnamento che prevede la realizzazione di un insieme di funzioni Python.

Il software viene sviluppato e documentato utilizzando gli strumenti, le tecniche e le convenzioni presentati durante il corso.

### 1.1 Materiale in linea

Tutto il materiale relativo al corso può essere reperito dal Moodle ufficiale del corso. Eventuali chiarimenti possono essere richiesti alle docenti per posta elettronica.

### 1.2 Struttura degli assegnamenti, bonus, tempi di consegna e prova orale

I due assegnamenti possono essere svolti individualmente o in gruppi di 2 studenti e possono essere consegnati entro la data e l'ora dell'appello di Febbraio 2024. Agli studenti che consegnano una realizzazione sufficiente entro 8 giorni

dalla pubblicazione su Moodle vengono assegnati 2 punti di Bonus nella valutazione complessiva di ciascun assegnamento. La valutazione complessiva viene data dalla media delle valutazioni dei due assegnamenti e costituisce la base per la prova orale. La prova orale sarà composta di due parti. La prima parte sarà una discussione sugli assegnamenti presentati e tenderà a stabilire se lo studente è realmente l'autore di quanto consegnato (in caso di dubbi la valutazione verrà opportunamente aggiustata) in particolare verrà chiesto di leggere e modificare il codice e di spiegare quanto usato (in particolare costrutti e moduli non facenti parte del programma del corso). La seconda parte verterà su tutto il programma del corso. In particolare, l'orale comprenderà:

- una discussione delle scelte implementative
- l'impostazione e la scrittura di semplici programmi Python (sequenziali e GUI) di difficoltà medio bassa rispetto a quelli visti nelle esercitazioni in classe
- domande su tutto il programma presentato durante il corso.

Il voto finale sarà la media (0-30L) fra la valutazione effettiva del progetto (emendata in caso di dubbi) e la prova orale.

### 1.3 Consegna degli assegnamenti

La consegna degli assegnamenti avviene *esclusivamente* per posta elettronica secondo le istruzioni presenti nel README di ciascun assegnamento.

### 1.4 Valutazione dell'assegnamento

All'assegnamento viene assegnata una fascia di valutazione da 0 a 30 che tiene conto dei seguenti fattori:

- motivazioni, originalità ed economicità delle scelte implementative
- strutturazione del codice (fattorizzazione del codice in funzioni, uso di strutture dati adeguate etc)
- efficienza e robustezza (numero di operazioni eseguite, fallimenti in caso di input inadeguati etc)
- aderenza alle specifiche
- qualità del codice Python e dei commenti

Tutti gli assegnamenti verranno confrontati automaticamente per verificare situazioni di plagio. Nel caso di elaborati uguali verranno presi provvedimenti per tutti i gruppi coinvolti.

## 2 L'assegnamento STUD

L'assegnamento prevede la realizzazione di un insieme di funzioni per la gestione della carriera semplificata degli studenti iscritti ad una ipotetica università. Ogni studente è rappresentato da una tupla:

```
(cognome, nome, matricola, lista esami sostenuti, note)
```

I campi `cognome`, `nome` e `note` sono di tipo stringa. Il campo `matricola` è un intero positivo. Il campo `lista esami sostenuti` è una lista che contiene coppie (`Codice_esame`, `Voto`) dove il codice dell'esame è una stringa e il voto un intero positivo compreso fra 18 e 33.

Ad esempio:

```
("Rossi", "Mario", 687245, ("235AF", 22), "In trasferimento")
```

L'insieme degli studenti è rappresentato come un dizionario con chiave il numero di matricola.

Si devono realizzare (almeno) le funzioni specificate nel file `stud.py` (leggere il file per le specifiche dettagliate).

È possibile definire funzioni ausiliarie per la soluzione ed estendere le funzionalità del catalogo definendo nuove funzioni.

Per ogni funzione, verificare che i tipi dei parametri passati durante la chiamata siano quelli attesi e segnalare le situazioni erranee con opportuni messaggi di errore stampati a schermo.

## 2.1 Istruzioni

Unitamente a questo file vengono forniti due file Python contenenti l'intestazione delle funzioni da realizzare (`stud.py`) e i test che devono essere superati per consegnare il codice (`main.py`) e un file di istruzioni (`README`).

Consigliamo di leggere attentamente le istruzioni e analizzare il codice fornito per capire come strutturare le funzioni prima di iniziare a progettare e implementare. Ricordiamo l'importanza di analizzare i vari casi prima di iniziare a scrivere codice e di effettuare test incrementali sul codice durante lo sviluppo.

Possono essere realizzate funzionalità in più rispetto a quelle richieste (ad esempio altre funzioni).

Le parti opzionali devono essere corredate da test appropriati e documentate da commenti chiari o (in caso sia necessario) da un breve documento descrittivo che può essere consegnato insieme al codice e che spiega le motivazioni e la struttura di quanto realizzato.