

# Conclusione del corso

Informatica@SEFA 2018/2019 - Lezione 26

**Massimo Lauria** <massimo.lauria@uniroma1.it>  
<http://massimolauria.net/courses/infosefa2018/>

Venerdì, 21 Dicembre 2018

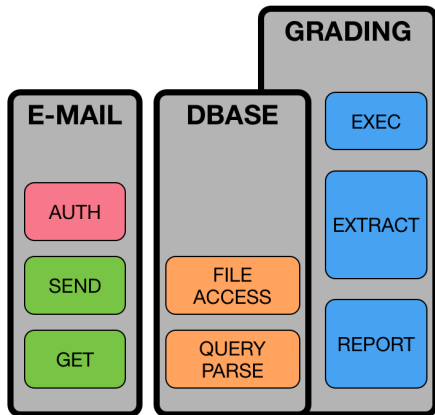
# In questo corso...

Dovreste aver appreso i rudimenti per

- ▶ ragionare in maniera astratta e generale
- ▶ riportare le astrazioni ad una dimensione tecnica
- ▶ programmare
- ▶ valutare la qualità ed efficienza di un algoritmo
- ▶ programmare in Python
- ▶ interrogare basi di dati in SQL

# Astrazione e sotto-problemi

## ***SUBMIT by EMAIL SYSTEM***



- ▶ gerarchia organizzativa
- ▶ sotto-problemi e sotto-programmi
- ▶ nascondere dettagli
- ▶ interfacce
- ▶ facile da analizzare
- ▶ divisione del lavoro

# Astrazioni e organizzazione del pensiero



Le astrazioni sono dei **pezzi logici** che modellano elementi del problema analizzato.

Sono gradini per costruire astrazioni di livello più alto.

# Livelli di astrazione (esempio)

- ▶ Circuiti
- ▶ Componenti hardware
- ▶ Sistema operativo
- ▶ Linguaggi di programmazione
- ▶ Applicazioni

# Esempi di astrazioni in Python

- ▶ Moduli
- ▶ Funzioni
- ▶ Variabili
- ▶ Operatori

# Il modello relazionale

Il modello relazionale è di fatto una grande astrazione di come i dati vengono organizzati su disco.

SQL è un linguaggio che sfrutta questa astrazione. Chi lo usa descrive quali dati vuole ottenere ma non deve descrivere **come** ottenerli.

# Programma del corso



# Introduzione all'informatica

Abbiamo dato alcuni cenni storici riguardanti l'evoluzione della disciplina dell'informatica. Poi abbiamo discusso di

- ▶ rappresentazione dei dati
- ▶ codifica di testi e numeri
- ▶ rappresentazione binaria, decimale, esadecimale
- ▶ ASCII, UTF-8, Latin1
- ▶ cosa sono gli algoritmi
- ▶ concetto di astrazione

# Programmazione Python

*Fondamenti di Programmazione in Python* di F. Pellacini.

Capitoli da 1 a 11.

# Algoritmi ed efficienza computazionale

- ▶ Ordine di crescita della complessità
- ▶ Notazioni  $O$ ,  $\Omega$ ,  $\Theta$
- ▶ Algoritmi di ricerca
- ▶ Algoritmi di ordinamenti
- ▶ Struttura a pila e programmi ricorsivi
- ▶ Risoluzione di equazioni di ricorrenza

# Introduzione alle basi di dati e SQL

*Manuale SQL*

Capitoli da 1 a 7.

# Altre letture

Tutte le dispense e le slide associate a tutte le lezioni.

# Esercizi

Mi aspetto che sappiate fare/capire tutti gli esercizi

- ▶ assegnati durante le lezioni di laboratorio;
- ▶ proposti nel materiale del corso (slide e dispense).

# Modalità di esame

# Disclaimer

Tutto quello che dico ha valore indicativo. Mi riservo il diritto di cambiare idea su **tutto** da qui a Gennaio e tra una sessione e l'altra.



# Modalità di esame

Ogni appello sarà costituito da

- appello scritto
- appello orale (+ consegna programma python)

Tipicamente le iscrizioni su infostud ad un appello chiudono il **giovedì** precedente all'appello stesso.

**Attenzione** per il primo appello di Gennaio chiudono un giorno prima per esigenze personali.

# Appelli scritti

Aula 2 CU002 - 9.00–14.00

- ▶ Primo appello scritto: 14/1/2019
- ▶ Secondo appello scritto: 11/2/2019
- ▶ Terzo appello scritto: 3/6/2019
- ▶ Quarto appello scritto: 1/7/2019
- ▶ Quinto appello scritto: 9/9/2019

# Appello orale

Gli orali si svolgeranno nei giorni successivi allo scritto corrispondente. Compatibilmente con

- ▶ il tempo di correggere gli scritti
- ▶ la disponibilità di aule
- ▶ altri impegni scientifici

La speranza è di poter fare tutto nella settimana dello scritto (eventualmente usando anche il Sabato).

In passato iniziavo gli orali il mercoledì seguente al compito, e cercherò di fare lo stesso anche quest'anno. Tendenzialmente cercherò di comunicare il giorno esatto entro lunedì sera.

# Sessione di Gennaio

Priorità agli studenti degli anni precedenti che hanno bisogno di verbalizzare entro la sessione di gennaio.

# Esercizi dell'esame scritto

## Contenuti

- Domande di teoria
- Esercizi sulla programmazione python
- Scrittura di una query SQL

## Tipi di esercizi

- Quiz a risposta multipla
- Esercizi con risposta aperta

# Domande di sbarramento

Il compito scritto prevede una o due domande di **sbarramento**. Se le risposte a queste domande non sono corrette, il compito è considerato fallito e la correzione del resto non avrà luogo.

**Es 1. (Domanda di sbarramento)** Quale di queste affermazioni è falsa?

- A ☐. un algoritmo termina sempre;
- B ☐. il risultato calcolato da un dato algoritmo dipende solo dal suo input;
- C ☐. la lunghezza di un algoritmo dipende dalla lunghezza dell'input;
- D ☐. la stessa funzione può essere calcolata da algoritmi diversi;

**Es 2. (Domanda di sbarramento)** Qual è la lista di indici generata dall'espressione python `range(3,8)`?

- A ☐. []      B ☐. [3,8]      C ☐. [3,4,5,6,7]      D ☐. [3,4,5,6,7,8]      E ☐. [3,7]

# Domande di programmazione (esempio)

- ▶ Scrivere una funzione che calcoli il minimo tra gli elementi di una sequenza `seq` che hanno posizione multipla di 7.
- ▶ Scrivere una funzione che data una stringa calcoli la sua inversa (cioè il primo carattere diventa l'ultimo, il secondo diventa il penultimo, e così via).
- ▶ Descrivere il programma seguente... (*e segue programma*)

# Domande di teoria

- ▶ codifiche di testo
- ▶ notazioni asintotiche
- ▶ algoritmi e complessità computazionale
- ▶ ricerca e ordinamento
- ▶ equazioni di ricorrenza
- ▶ ...



# Altre domande

- ▶ Query SQL
- ▶ Scrivere pezzi o varianti degli algoritmi fatti in classe

# Punteggio

Fatta eccezione per le domande di sbarramento, tutte le altre hanno un punteggio associato che varia a seconda della difficoltà **stimata** della domanda.

La somma dei punti di tutte domande di un compito è **superiore a 30**, così non è necessario rispondere a tutto per avere il voto massimo.

# Materiale che potete usare durante l'esame

## Fornito da me

- ▶ Traccia del compito (da riconsegnare)
- ▶ Fogli protocollo per brutte copie (da riconsegnare)

## Non fornito da me

- ▶ documento per l'identificazione
- ▶ penne, matite, cancelleria

Il materiale non può essere condiviso. Tutte le vostre cose devono essere messe in un angolo della stanza (compresi telefoni, che devono essere spenti).

# Disciplina durante lo scritto

Copiare o collaborare durante l'esame è **severamente vietato**, e violazioni di questa regola causeranno come minimo l'immediata espulsione dall'aula e l'annullamento del compito.

Più alta sarà la densità di studenti nell'aula, e più rigido e **sommario** sarà il controllo.

Qualunque richiesta deve essere fatta alzando la mano e comunicando **esclusivamente** con me.

Solo **una persona alla volta** può uscire dall'aula a meno che non consegni o rinunci.

# Validità dello scritto

Uno scritto con voto maggiore o uguale di 18 vale fino alla sessione di Settembre (quinto appello), ma

- ▶ se **consegnate** un nuovo scritto, il vecchio viene eliminato. Indipendentemente dall'esito del nuovo scritto.
- ▶ se all'orale mi rendo conto che lo scritto non è stato fatto dallo studente, potrei eliminarlo in ogni caso.

# Modalità dell'esame orale

- ▶ Il programma va consegnato via email **la sera prima** dell'appello orale

## Durante l'esame

- ▶ Test del programma python
- ▶ Domande sullo scritto
- ▶ Domande sul programma python
- ▶ Domande sui contenuti del corso

# Programma python da consegnare

Verrà pubblicata una serie di esercizi.

Dovrete consegnare un file python contenente le soluzioni di tutti gli esercizi. La traccia verrà pubblicata spero entro una settimana dopo la fine del corso (incrociate le dita).

La traccia sarà accompagnata da test **pubblici** per verificare/migliorare/provare le vostre soluzioni.

I test pubblici possono disambiguare la descrizione dell'esercizio.

# Valutazione del programma

- ▶ viene controllato con software anti-plagio
- ▶ deve passare **perfettamente tutti** i test pubblici o **non potrete** fare l'orale
- ▶ verranno eseguiti dei test **segreti** il cui esito influirà sulla valutazione
- ▶ verranno fatte domande sul codice della soluzione



# Variazioni

**Seguite le pagine del corso, in particolare**

`massimolauria.net/courses/infosefa2018/esami.html`

# Lavorate sodo e buone vacanze

