

# Linguaggi di Programmazione e compilatori

6 CFU - II° Anno

## Informazioni generali

- Home Page:  
<http://www.di.unipi.it/~tesei/unicam/LPC20032004.html>
  - Sigla: LPC
  - Orario:
    - Lun 11-13 [AB2] 17-19 [AB2]
    - Mar 9-11 [AA1] 17-19\*\* [AA1]
- \*\* Modificato

## Informazioni Generali

- Ricevimento  
Mar 14-15 + 16-17 (buco nel mezzo perché c'è lezione)
- Email  
[luca.tesei@unicam.it](mailto:luca.tesei@unicam.it)
- Ufficio  
Polo Informatico, 1° piano
- Telefono  
0737 402572

## Programma

- Introduzione: linguaggi di programmazione, paradigmi, storia.
- Macchine astratte: definizione, implementazione, compilazione, interpretazione
- Struttura generale di un compilatore: fasi di compilazione, primi concetti

## Programma

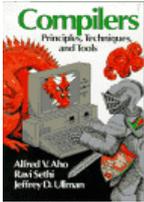
- Analisi lessicale: riconoscimento di tokens, espressioni regolari, automi, algoritmi
- Analisi sintattica: grammatiche libere, automi a pila, analisi discendente, analisi ascendente, costruzione di tabelle LR(1), LL(1), LALR(1), SLR(1)

## Programma

- Analisi semantica: attributi sintetizzati ed ereditati, schemi di traduzione
- Analisi statiche tramite attributi e regole della grammatica: type checking, ...
- Generazione del codice a triple (tipo assembly)
- Cenni all'ottimizzazione

## Materiale

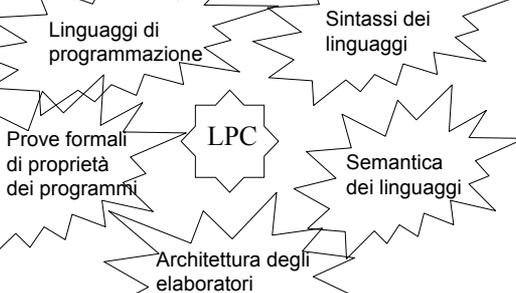
- Dispense, lucidi ppt distribuiti durante il corso
- Esercizi con soluzione e non (links sulla pagina web)
- Libro di riferimento: "Compilers, principles, techniques and tools" di Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman Addison-Wesley Pub



## Esame

- Prova scritta e possibilità di prova orale integrativa
- Appelli 8: 2 giugno/luglio, 2 settembre/ottobre, 2 dicembre/gennaio, 2 marzo/aprile
- Per i non frequentanti: preparare l'esame su dispense + lucidi + esercizi con soluzione + ricevimenti dedicati

## Discipline principali e collegate



## Possibili perplessità

- Nella vita non scriverò mai un compilatore
- *Non si sa mai, e comunque almeno un parser, se fai il programmatore, lo dovrai prima o poi implementare*
- A cosa mi serve studiare tutti questi algoritmi?
- *Sono quelli che hanno stimolato gli informatici fin dagli inizi, sono ottimi esempi di formalizzazione e risoluzione di problemi, nella cultura di base di un informatico ci devono stare ☺*

## Altre perplessità?

- Noi non lo faremo, ma se volete vi posso dare indicazioni su come implementare le varie fasi (soprattutto i parser e gli analizzatori lessicali con i relativi generatori automatici)
- Linguaggi adatti di implementazione: C, C++, Java (of course)