



Corso di  
Linguaggi di Programmazione + Lab.  
A.A. 2012-2013  
Docente: Marco de Gemmis

Presentazione dell'insegnamento



# Informazioni generali

- Corso: **Linguaggi di Programmazione + Laboratorio**
- Docente: **Marco de Gemmis - RICERCATORE**
- Lezioni: **2 Ottobre 2011 - 21 Dicembre 2011**
  - **Mercoledì e Venerdì** dalle ore 9.30 alle ore 14.30
- Ricevimento: **per appuntamento, in genere dopo le lezioni**
- Web
  - *Informazioni sul corso, esami, tracce e materiale didattico*  
<http://www.di.uniba.it/~swap/degemmis.html>  
*seguendo il link "courses"*
  - avvisi generici: [http://informatica.di.uniba.it/info\\_comuni/bacheca.php](http://informatica.di.uniba.it/info_comuni/bacheca.php)  
*e/o sulla pagina del corso*
- Contatti
  - Telefono: **080 5442276**
  - Email: [degemmis@di.uniba.it](mailto:degemmis@di.uniba.it)



# Programma del corso

## ■ Tempi

- 72 ore di lezioni frontali equivalenti a 9 crediti T1
- 45 ore di laboratorio equivalenti a 3 crediti T2
- 183 ore di studio individuale
- Totale crediti 12 (9 T1 + 3 T2)

## ■ Prerequisiti

- Matematica discreta
- Programmazione imperativa
- Conoscenza dei linguaggi C / C++



# Programma del corso

- **Obiettivi formativi**
  - Fornire gli strumenti fondamentali, formali e pratici, per la definizione di linguaggi di programmazione e la progettazione dei loro traduttori.
- **Strutturazione del corso**
  - **Prima parte:**
    - Si illustrano gli aspetti più significativi dei linguaggi di programmazione, la loro evoluzione ed i concetti che stanno alla base della traduzione dei linguaggi di alto livello
    - Si forniscono i concetti più significativi della teoria dei linguaggi formali, enfatizzando gli aspetti generativi e riconoscitivi dei linguaggi formali
  - **Seconda parte:**
    - Si presentano/sviluppano in laboratorio alcuni esempi di progettazione e realizzazione di programmi per la manipolazione di grammatiche



# Programma del corso

## ■ Obiettivi professionalizzanti

- Comprensione profonda dei linguaggi di programmazione e dei principali paradigmi di programmazione
- Conoscenza delle tecniche di analisi e traduzione dei linguaggi di programmazione
- Conoscenza delle espressioni regolari e delle grammatiche



# Programma del corso

## ■ Contenuto

### □ Parte I: Linguaggi di Programmazione e Compilatori

- Linguaggi di programmazione. Gerarchia di linguaggi di programmazione e di macchine astratte: linguaggi macchina, linguaggi assembler, linguaggi di alto livello. Linguaggi di alto livello: interpretazione e compilazione, sintassi, semantica, analisi di programmi. L'evoluzione dei linguaggi di programmazione
- Linguaggi formali. Inquadramento della teoria dei linguaggi formali nell'informatica teorica. Classificazione di Chomsky. Operazioni sui linguaggi e proprietà di chiusura delle classi di linguaggi rispetto alle operazioni. Generazione di linguaggi: un'introduzione alle grammatiche. Carte sintattiche e BNF, grammatiche generative, derivazione, linguaggio generato da una grammatica, equivalenza tra grammatiche, relazione tra grammatiche e linguaggi, non determinismo, correttezza di una grammatica: indecidibilità del problema



# Programma del corso

## ■ Contenuto

### □ Parte I: Linguaggi di Programmazione e Compilatori

- I compilatori. Il modello di un compilatore: analizzatore lessicale, analizzatore sintattico, analizzatore semantico, generazione e ottimizzazione del codice
- Analisi lessicale. Linguaggi regolari, espressioni regolari, automi e linguaggi a stati finiti. Teoremi di equivalenza
- Analisi sintattica. Linguaggi liberi da contesto



# Programma del corso

## ■ Contenuto

- Parte I: Linguaggi di Programmazione e Compilatori
  - Cenni sulla Tabella dei simboli (TS). Costruzione ed interazione, contenuti, operazioni, organizzazione di TS per linguaggi non a blocchi, rappresentazione di TS. Tabelle hash. Linguaggi a blocchi: organizzazione di TS. TS a stack
  - Gestione della memoria (statica, a stack, a heap)
  - Cenni su analisi semantica, formato intermedio del programma sorgente, espressioni e priorità degli operatori, generazione del codice oggetto, gestione degli errori





# Programma del corso

- **Contenuto**

- **Parte II: Laboratorio**

- Programmazione orientata agli oggetti in C++
    - Realizzazione di programmi per manipolare le grammatiche



# Prove d'esame

## ■ Prova scritta

- Esercizi sulla teoria dei linguaggi formali
- Domande sulla teoria dei linguaggi formali (definizioni, dimostrazioni di teoremi, etc.) e sui linguaggi di programmazione

## ■ Prova orale

- Discussione esercitazioni di laboratorio
- Domande sulla teoria dei linguaggi formali



## Organizzazione e valutazione prove d'esame

### ■ Prova scritta

- Si supera con una valutazione minima pari a 18

### ■ Prova orale

- Va sostenuta nello stesso appello della prova scritta
- Ammissione vincolata al superamento della prova scritta
- In caso di esito negativo, si ripete anche la prova scritta

### ■ Valutazione finale

- media delle valutazioni ottenute nelle due prove

### ■ Prove in itinere

- Due prove sulla teoria dei linguaggi formali
- esonero dalla prova scritta fino alla sessione di recupero (settembre 2013)



## Testi di riferimento

- **Semeraro, G., Appunti di teoria dei linguaggi formali, Adriatica, Bari, 1996.**
- Maurizio Gabbrielli, Simone Martini, Linguaggi di Programmazione, Principi e paradigmi. McGraw-Hill
- Libera scelta riguardo a manuali di C / C++
- Dispense del corso
- Altri
  - Hopcroft, Motwani, Ullman, Automi, linguaggi e calcolabilità. Addison-Wesley
  - Aho, A. V., Sethi, R., and Ullman, J. D., Compilers, Principles, Techniques and Tools, Addison-Wesley, 1988.
  - Zarrella, J., Concetti dei Linguaggi di Programmazione, Scienze Tecnologiche Informatiche, Franco Angeli, 1989.
  - Sethi, R., Linguaggi di Programmazione, Zanichelli, 1994.