

# Linguaggi di Programmazione con Laboratorio

-CdS Matematica

Marco Bellia, Dip. Informatica, Università di Pisa

February 24, 2016

- Corso 2016
  - Programma
  - Obiettivi del corso
  - Organizzazione
- Linguaggi di Programmazione (LP):  
Cosa sono? Perché più di uno? Perché studiarli?
- Due classi di base: Prescriptive vs. Descriptive
- Paradigmi di Linguaggi di Programmazione
- 50 anni of Linguaggi di Programmazione: 1955-2005

- **Tema del Corso**

Linguaggi di Programmazione: Principi e Applicazioni

- **Attività del Laboratorio**

Esperienze di implementazione di LP

- **Prerequisiti**

- Fondamenti di Programmazione e Laboratorio
- Programmazione di Algoritmi di base in almeno 1 Linguaggio

- **Materiale**

- Gabrielli M., S. Martini, Linguaggi di Programmazione: Principi e Paradigmi (seconda edizione), McGraw-Hill, Milano, 2011, ISBN 9788838665738,
- Materiale (Slides, Listings, Articoli, Esercizi) distribuito o indicato durante il corso

# Materiale: Come si usa

- **Libro:** Gabrielli M., S. Martini, Linguaggi di ....
  - Fornisce una lettura organica degli argomenti trattati nel corso
  - ma *non esaustiva*
- **Materiale Vario:** Slides, Listings, Articoli, Esercizi
  - Fornisce una completa traccia:
    - degli argomenti trattati
    - delle relazioni tra essi evidenziati
    - degli approfondimenti discussi e
    - del materiale aggiuntivo da conoscere
  - ma *non è autonomo*
- **Integrati tra loro** nello studio
  - Q. Dopo ogni lezione va *prima* guardato il materiale e *dopo* le pagine del libro indicate, o il *contrario*, o...?
  - A. Inessenziale ordine e numero di ri-letture per avere una *chiara comprensione*
- **Usare il ricevimento** quando questa integrazione non riesce

- LP, Macchine Astratte (MA), Esecutori di Programmi
- Metodi di definizione di LP
- Struttura dei Linguaggi LP e influenza su:
  - Metodologie di Programmazione
  - Spettro delle applicazioni: Web, Mobile, Enterprise, Embedded
- Meccanismi di supporto (della struttura):
  - proprietà e principali realizzazioni
  - influenza sull'uso dei costrutti
- Classificazione degli LP. Studio
  - dei Paradigmi:
    - Linguaggi Imperativi: Struttura, Applicazioni, C (Pascal)
    - Linguaggi Funzionali: Struttura, Applicazioni, OCaml
    - Linguaggi Object Oriented: Struttura, Applicazioni, Java
  - delle Relazioni tra programmi sviluppati nei paradigmi

- Strumenti Usati:
  - formalismi per Sintassi, e per Semantica,
  - linguaggio OCaml
- Attività:
  - Definizione di sotto/frammenti di linguaggi dei paradigmi trattati nel corso
  - Realizzazione di esecutori di programmi per tali linguaggi

## Alla fine del corso:

- Maggiore *consapevolezza dei limiti e delle possibilità* dello LP che si conosce, si sta usando, si deve usare
- Migliore *conoscenza dell'uso* di uno LP e del costo dei costrutti impiegati nei programmi scritti in esso
- Esperienze di programmazione di *una stessa applicazione in differenti paradigmi* di programmazione
- Esperienza di programmazione nella *realizzazione di esecutori di Linguaggi* o sue parti.

- **Docenti:** Marco Bellia, Vincenzo Ciancia
- **Orario Corso:** In Aula L  
Lunedì 14-16;  
Martedì 11-13;  
Mercoledì 09-11.
- **Orario Laboratorio:** In Aula M  
Venerdì 11-13.
- **Contatti:** [bellia@di.unipi.it](mailto:bellia@di.unipi.it) (usare argomento: "LPL ....")
- **Pagina del corso:** [www.di.unipi.it/~bellia](http://www.di.unipi.it/~bellia)
- **Orario di ricevimento:**  
Lunedì al termine della Lezione o Esercitazione  
su appuntamento



- **Modalità di Esame e Valutazione:**  
Media pesata della Valutazione Corso (80/100) + Valutazione Laboratorio (20/100)
- **Valutazione Corso:** Media ottenuta su due prove scritte  
Prova Preliminare:  
Esercizi sugli argomenti trattati  
Prova Finale:  
Test a risposta multipla e/o aperta sugli argomenti trattati  
Prova Orale:  
su richiesta
- **Valutazione Laboratorio:** Contestuale alle attività svolte da ciascun studente durante il laboratorio.

# Linguaggi di Programmazione: Cosa sono?

- Rispondiamo in 5 slides mostrando che sono lo strumento di congiunzione tra alcune nozioni correlate.  
In particolare, sono:
  - Gli **Strumenti** Formali per
    - **Esprimere** le *Computer Applications*
    - **Definire** le *Funzioni Calcolabili*
    - **Implementare** gli *Algoritmi* e renderli *Processi Automatici*

Gli **strumenti fondamentali** per:

- **Esprimere Tutte** le *Computer Applications* che sono:
  - Tutte le applicazioni che:
    - *sono state, sono e saranno* realizzabili in Processi Automatici (eseguibili oggi, su computers *isolati e/o interconnessi*)
    - *pervadono ogni comparto e attività* della nostra esistenza: Information, Production, Education, Research, Culture, Health...
- **Definire Tutte** le *Funzioni Calcolabili*,  $\mathcal{F}$ .

Gli **strumenti fondamentali** per:

- **Esprimere Tutte** le *Computer Applications* che sono: ...

- **Definire Tutte** le *Funzioni Calcolabili*,  $\mathcal{F}$ .

Sia  $\mathcal{D}$  insieme numerabile,

- $\mathcal{F} \equiv [\mathcal{D} \rightarrow \mathcal{D}] \subset \mathcal{D} \rightarrow \mathcal{D}$ <sup>1</sup>
- $\mathcal{D} \cong^2 [\mathcal{D} \rightarrow \mathcal{D}]$
- definite<sup>3</sup> mediante *specifici formalismi*  $\mathcal{L}$  che includono:  
Combinatory Logic, Lambda Calculus, Turing Machines,  
**Linguaggi di Programmazione (LP)**

---

<sup>1</sup>  $\mathcal{D} \rightarrow \mathcal{D}$  è l'insieme non numerabile contenente tutte le funzioni da  $\mathcal{D}$  in  $\mathcal{D}$

<sup>2</sup>  $\cong$  sta per *isomorfo*

<sup>3</sup> "definite", qui, significa *finitamente, completamente definite*.

Gli **strumenti fondamentali** per:

- **Esprimere Tutte** le *Computer Applications* che sono: ...
- **Definire Tutte** le *Funzioni Calcolabili*,  $\mathcal{F}$ .

Sia  $\mathcal{D}$  insieme numerabile,

- $\mathcal{F} \equiv [\mathcal{D} \rightarrow \mathcal{D}]$
- $\mathcal{D} \cong [\mathcal{D} \rightarrow \mathcal{D}]$

- Sia  $\mathcal{L} \in \mathbf{LP}$ . Sia  $\mathcal{P}$ <sup>4</sup> l'insieme (numerabile) dei programmi di  $\mathcal{L}$ . Sia  $\forall p \in \mathcal{P}, \bar{p} \in \mathcal{D}$  una iniezione di  $\mathcal{P}$  in  $\mathcal{D}$ .

- (Universale)  $\exists \mathcal{U} \in \mathcal{F}$ ,
- (Mapping)  $\exists [|-|] : \mathcal{P} \rightarrow \mathcal{F}$ ,
- $\forall p \in \mathcal{P}$ ,  
$$\mathcal{U}(\bar{p}) = g \in (\mathcal{D} \cong [\mathcal{D} \rightarrow \mathcal{D}] \equiv) \mathcal{F}$$
- $\forall g \in \mathcal{F}$ , sia  $g = [|p|]$ , per qualche  $p \in \mathcal{P}$   
$$\mathcal{U}(\bar{p})(x) = g(x) \quad \forall x \in \mathcal{D}$$

---

<sup>4</sup>  $\mathcal{P}$  dipende da  $\mathcal{L}$ , e scriveremo estesamente  $\mathcal{P}_{\mathcal{L}}$  quando più linguaggi  $\mathcal{L}$  sono considerati e potrebbe esserci ambiguità. Analogamente, è la funzione universale  $\mathcal{U}$  di  $\mathcal{L}$ , e il mapping suriettivo  $[|-|]$

- 1 Scrivere un programma  $C$  che calcoli la funzione *ordinamento di sequenza* implementando l'algoritmo di QuickSort. Allo scopo completare il testo esponendo eventuali vincoli che si ritenga utile assumere.
- 2 Se e dove, nella formulazione  $\mathcal{F}$ , vediamo che  $\mathcal{F}$  è numerabile.
- 3 Se e dove, nella formulazione  $\mathcal{F}$ , vediamo che la funzione  $[[\cdot]]$  è calcolabile. E se sì, cosa calcola.
- 4 Se e dove, nella formulazione  $\mathcal{F}$ , vediamo che  $\mathcal{P}$  contiene un programma che calcola  $[[\cdot]]$ .
- 5 Se e quale relazione, nella formulazione  $\mathcal{F}$ , vediamo tra l'iniettiva  $\bar{\cdot} : \mathcal{P} \rightarrow \mathcal{D}$  e la suriettiva  $[[\cdot]]$ .
- 6 Se e dove, nella formulazione  $\mathcal{F}$ , vediamo che per ogni funzione calcolabile  $c$ 'è un programma che la calcola.
- 7 Se e dove, nella formulazione  $\mathcal{F}$ , vediamo che per ogni programma  $c$ 'è una funzione calcolabile che è la funzione calcolata dal programma.
- 8 Se e dove, nella formulazione  $\mathcal{F}$ , vediamo che  $[[\cdot]]$  stabilisce una corrispondenza uno-uno tra  $\mathcal{P}$  e  $\mathcal{F}$ .
- 9 Sia  $u \in \mathcal{L}$  un programma di  $\mathcal{L}$  che calcola la funzione  $[[\cdot]]$  (vedi anche esercizio 4). Si dica cosa calcola  $U(\bar{u})$ .
- 10 Con riferimento alla formulazione  $\mathcal{F}$ . Sia  $U_{\mathcal{P}} \equiv \{p \in \mathcal{P} \mid [[p]] = U\}$ . Si discuta la cardinalità di  $U_{\mathcal{P}}$ .