



# UNIVERSITÀ DI PISA

---

## PARADIGMI DI PROGRAMMAZIONE

**PAOLO MILAZZO**

Anno accademico 2021/22  
CdS INFORMATICA  
Codice 733AA  
CFU 9

Moduli	Settore/i	Tipo	Ore	Docente/i
PARADIGMI DI PROGRAMMAZIONE	INF/01	LEZIONI	72	PAOLO MILAZZO

### Obiettivi di apprendimento

#### *Conoscenze*

Imparare i principi alla base dei principali paradigmi di programmazione (funzionale, object-oriented e concorrente) e del funzionamento dei linguaggi di programmazione moderni.

#### *Modalità di verifica delle conoscenze*

Esame finale, scritto e orale

#### *Capacità*

Imparare a ragionare sul funzionamento dei linguaggi di programmazione in astratto, tramite modelli formali del loro comportamento. Imparare ad apprendere nuovi linguaggi di programmazione. Imparare a programmare nel linguaggio OCaml.

#### *Modalità di verifica delle capacità*

Esercitazioni durante il corso ed esame finale

#### *Comportamenti*

Programmare in modo consapevole, avendo cognizione di cosa accade al programma in fase di compilazione ed esecuzione.

#### *Modalità di verifica dei comportamenti*

Esercitazioni durante il corso ed esame finale

#### **Prerequisiti (conoscenze iniziali)**

- Elementi di linguaggi formali, semantica e programmazione visti nei corsi del primo anno
- Linguaggio JavaScript

#### **Corequisiti**

Nessuno

#### **Prerequisiti per studi successivi**

Nessuno

#### **Indicazioni metodologiche**

Lezioni frontali con slides e uso dei jupyter notebooks in OCaml

#### **Programma (contenuti dell'insegnamento)**

PROGRAMMA DI MASSIMA:

- Paradigma di programmazione funzionale

L'essenza: il  $\lambda$ -calcolo; Il linguaggio OCaml; Formalizzazione di un linguaggio funzionale; Implementazione in OCaml di un interprete del linguaggio funzionale; Sistemi di tipi nei linguaggi funzionali; Il linguaggio Haskell (peculiarità)



## UNIVERSITÀ DI PISA

---

- Paradigma di programmazione object-oriented

L'astrazione dei dati e gli oggetti; Modularità, incapsulamento, ereditarietà, principio di sostituzione, e class-based vs object-based; Formalizzazione di linguaggi object-oriented; Sistemi di tipi nei linguaggi object-oriented; Tecniche di implementazione di linguaggi class-based (es. Java Virtual Machine); Esempi di implementazione in OCaml

- Paradigma di programmazione concorrente

Concetti di base: esecuzione non sequenziale; Formalizzazione di un modello di concorrenza ed esempi di implementazione in OCaml; Costrutti di programmazione concorrente nei linguaggi moderni

### Bibliografia e materiale didattico

Materiale didattico:

- Slides e dispense fornite dal docente
- Jupyter notebooks in OCaml
- Registrazione delle lezioni

Libri di testo consigliati (per approfondimenti):

- Real World OCaml ( <https://dev.realworldocaml.org/> )
- Peter Sestoft, Programming Language Concepts
- Maurizio Gabrielli e Simone Martini, Linguaggi di programmazione

### Indicazioni per non frequentanti

Nessuna

### Modalità d'esame

Scritto e orale

### Stage e tirocini

Nessuno

### Pagina web del corso

[https://teams.microsoft.com/team/19%3a9h-Fsn8lijospdPolmNv\\_tpv4P7PLjVZ4JAVzL8l33Q1%40thread.tacy2/conversations?groupId=66deb541-11be-4992-bf56-c6ff5be951a3&tenantId=c7456b31-a220-47f5-be52-473828670aa1](https://teams.microsoft.com/team/19%3a9h-Fsn8lijospdPolmNv_tpv4P7PLjVZ4JAVzL8l33Q1%40thread.tacy2/conversations?groupId=66deb541-11be-4992-bf56-c6ff5be951a3&tenantId=c7456b31-a220-47f5-be52-473828670aa1)

### Altri riferimenti web

Tutto il materiale sarà messo a disposizione nel team del corso su Microsoft Teams

### Note

Nessuna

*Ultimo aggiornamento 23/11/2021 12:00*