



UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

SYLLABUS DEL CORSO

Linguaggi di Programmazione

2627-2-E3102Q109

Obiettivi

Conoscenza e capacità di comprensione

Gli studenti acquisiranno una solida comprensione dei principali paradigmi di programmazione, con particolare attenzione a quello logico e a quello funzionale. Comprenderanno le differenze concettuali tra i paradigmi, nonché le motivazioni storiche e teoriche alla base del loro sviluppo, inclusi aspetti legati alla tipizzazione, alla modellazione dei problemi e al controllo del flusso computazionale.

Conoscenza e capacità di comprensione applicate

Gli studenti saranno in grado di applicare le conoscenze acquisite nello sviluppo di programmi in linguaggi rappresentativi dei paradigmi studiati (es. Common Lisp per il paradigma funzionale, Prolog per quello logico). Utilizzeranno ambienti di sviluppo specifici per progettare e implementare soluzioni a problemi di complessità medio-bassa, anche in contesti applicativi diversi.

Autonomia di giudizio

Gli studenti svilupperanno la capacità di valutare criticamente i diversi paradigmi di programmazione in relazione al problema da risolvere, scegliendo consapevolmente linguaggi e tecniche in base ai vincoli del dominio applicativo. Saranno in grado di analizzare e confrontare approcci alternativi alla programmazione sulla base di criteri teorici e pratici.

Abilità comunicative

Gli studenti saranno in grado di descrivere in modo chiaro e rigoroso soluzioni software e scelte progettuali, sia oralmente sia per iscritto, utilizzando un linguaggio tecnico appropriato. Sapranno inoltre discutere e motivare le scelte di paradigma e linguaggio adottate in relazione al problema affrontato.

Capacità di apprendere

Il corso fornirà agli studenti strumenti teorici e pratici per approfondire autonomamente nuovi linguaggi e modelli di programmazione. In particolare, saranno messi in grado di affrontare con consapevolezza e spirito critico l'apprendimento di linguaggi non trattati direttamente nel corso ma appartenenti ai paradigmi studiati.

Contenuti sintetici

L'insegnamento ha l'obiettivo di mostrare il panorama dei paradigmi di programmazione secondo la tradizionale suddivisione tra paradigma imperativo, paradigma logico (dichiarativo) e paradigma funzionale.

Programma esteso

1. I paradigmi di programmazione: imperativo, logico (dichiarativo) e funzionale. Richiami delle nozioni di "run-time" e di esecuzione di un programma su un'architettura idealizzata a pila (stack). Nozioni base degli ambienti di programmazione per i diversi sistemi presentati. Cenni storici.
2. Il paradigma di programmazione logico. Introduzione al linguaggio di programmazione Prolog.
3. Il paradigma di programmazione funzionale. Introduzione al linguaggio di programmazione LISP (Common Lisp). Note su Haskell, Julia e Javascript.
4. Utilizzo dei vari paradigmi in situazioni e contesti diversi: tipizzazione, il concetto di astrazione e abstract data types, concorrenza ed event handling

Prerequisiti

Concetti di base di programmazione ricorsiva, logica matematica e di architetture hardware e software.

Modalità didattica

Il corso si svolgerà mediante lezioni frontali in presenza e con l'ausilio di sistemi di e-learning per la distribuzione del materiale didattico e di esercizi e progetti. Ci saranno inoltre delle lezioni in laboratorio per imparare a programmare nei linguaggi considerati a lezione.

Indicativamente, e a seconda delle necessità contingenti l'insegnamento sarà articolato su

- 48 ore di lezione frontale da 2-3 ore svolte in modalità erogativa in presenza;
- 8 attività di laboratorio da 3 ore svolte in modalità interattiva in presenza.

Materiale didattico

Libro di Testo:

Maurizio Gabbriellini, Simone Martini (2/ED): LINGUAGGI DI PROGRAMMAZIONE: Principi e Paradigmi", McGraw Hill, ISBN 8838665737 · 9788838665738

Testi consigliati:

- Robert W. Sebesta (12th Edition). Concepts of Programming Languages. Pearson, 2025
- Leon Sterling and Ehud Shapiro, The Art of Prolog Advanced Programming Techniques - 2nd Edition;
- Abelson, Sussman e Sussman, Structure and Interpretation of Computer Programs (SICP);
- Abelson, Sussman, Henz and Wrigstad, Structure and Interpretation of Computer Programs (SICP) - Javascript edition;
- Peter Seibel, Practical Common Lisp (PCL);

- Varie risorse online per gli altri linguaggi di programmazione presentati.

Periodo di erogazione dell'insegnamento

Primo semestre

Modalità di verifica del profitto e valutazione

L'insegnamento di Linguaggi di Programmazione prevede il superamento di una prova teorica (uno scritto) e di un progetto di programmazione. A discrezione del docente potrà essere richiesta anche una prova orale. La prova scritta ed il progetto corrispondente al primo appello sono strutturati in due prove "in itinere".

Le prove in itinere sono due scritti offerti in Novembre (Programmazione Logica e Prolog) e a Gennaio/Febbraio (Programmazione Funzionale; Programmazione Imperativa), e uno o due progetti in itinere, postati a fine Ottobre, da consegnare in forma elettronica a Gennaio/Febbraio. Le prove in itinere costituiscono, come già detto, il primo appello per il corso, e non sono recuperabili nei restanti appelli.

Dall'appello successivo (Febbraio/Marzo) lo scritto comprende tutte le parti del corso. Lo stesso vale per i progetti. Gli appelli seguenti saranno a Giugno, Luglio e Settembre (in data da precisarsi).

La parte teorica ed i progetti di laboratorio vanno superati entrambi nella stessa tornata; il voto complessivo è la media, ponderata a seconda dei casi, ovvero a discrezione del docente, dei voti parziali. In particolare, una media aritmetica almeno pari a 18 non è sufficiente per garantire il superamento dell'esame.

I progetti possono essere svolti in gruppo (massimo 3 persone). Chi svolge il progetto da solo od in coppia potrà avere un bonus.

Il voto è in trentesimi, ed esprime una valutazione complessiva di tutto ciò che concorre al raggiungimento degli obiettivi formativi dichiarati. Cioè, è frutto di una valutazione complessiva delle varie caratteristiche delle prove. Per esempio: chiarezza, rigore, autonomia di giudizio, capacità di scegliere esempi e di illustrare l'argomento in modo efficace.

Nota finale: tutti gli esiti saranno caricati nel sistema: voto, ritirato, assente, insufficiente etc.

Orario di ricevimento

Prof. Marco Antoniotti: su appuntamento via email.

Prof. Claudio Ferretti: su appuntamento via email.

Prof. Fabio Sartori: su appuntamento via email.

Sustainable Development Goals

