UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

COURSE SYLLABUS

Programming Languages

2223-2-E3101Q108

Obiettivi

Gli studenti apprenderanno vari paradigmi programmazione, in particolare il paradigma logico e quello funzionale. Apprenderanno inoltre a utilizzare ambienti per programmare nei principali linguaggi presentati. Gli studenti saranno in grado di sviluppare progetti di piccole e medie dimensioni in (Common) Lisp e Prolog utilizzando gli ambienti di programmazione presentati.

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 e Julia.
4. Il paradigma di programmazione imperativo. Introduzione al linguaggio di programmazione C/C++.
5. Utilizzo dei vari paradigmi in situazioni e contesti diversi.

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.

Materiale didattico

- 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);
- Peter Seibel, Practical Common Lisp (PCL);
- Brian W. Kernigham & Dennis M. Ritchie, C Programming Language (2?? Edition), Prentice Hall, 1988;
- Robert Sedgewick, Algorithms in C, Parts 1-5 (Bundle): Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms (3rd Edition), Addison Wesley, 2001

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 e Lisp; Programmazione Imperativa e C/C++), e uno o due progetti in itinere, postati a fine Ottobre, da consegnare in forma elettronica a Gennaio/Febbraio (Common Lisp, Prolog e/o C++). 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 posso essere svolti in gruppo (massimo 3 persone), ma sono da consegnare individualmente. 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 cioè 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: giorno da decidersi o su appuntamento via email.

Prof.ssa Gabriella Pasi: su appuntamento via email.

Prof. Fabio Sartori: su appuntamento via email.

Sustainable Development Goals