



UNIVERSITÀ
DI PARMA

Programmazione di Applicazioni Software

Alberto Ferrari

- Programmazione di applicazioni software (PASW)
- **crediti:**
 - **9 CFU** (corrispondenti a circa 63 ore di lezione)
- **orario:**
 - Lunedì 16.30-18.30 – aula N o lab. Informatica 2,3
 - Martedì 14.30-16.30 – aula informatica E
 - Giovedì 14.30-16.30 – aula N
- **docente:** Prof. Alberto Ferrari (alberto.ferrari@unipr.it)
 - Phone: +39 0521 90 5708
 - ricev.: Ingegneria Sede Scientifica – Palazzina 1 – giovedì 11:00-12:00
- **materiale:** <http://elly.dia.unipr.it/2017/course/view.php?id=417>

- architettura degli elaboratori
- sistemi operativi
- programmazione orientata agli oggetti
- strutture dati e algoritmi notevoli
- sviluppo di applicazioni in ambiente locale e applicazioni distribuite

- lezioni in aula
- soluzione guidata di esercizi in aula
- esercizi di programmazione in laboratorio
 - le esercitazioni in laboratorio sono centrali per il corso
 - gli esercizi proposti vertono sugli stessi argomenti generali delle lezioni in aula
- come trama continua del corso, verrà proposto lo sviluppo incrementale di un'applicazione e, facendo leva sui suoi requisiti, sul suo sviluppo e sulla sua ottimizzazione, verranno recuperati o introdotti concetti fondamentali della programmazione object oriented, della multiprogrammazione e dell'interazione tra processi

- l'esame consiste in
 - una prova sui concetti teorici (brevi esercizi e quiz)
 - una prova di programmazione (lo sviluppo o il completamento di un'applicazione software da sviluppare in laboratorio)
 - eventualmente seguita da una prova orale integrativa
 - è previsto l'utilizzo di prove di valutazione in itinere

- ogni squadra (max due studenti) svilupperà nel corso delle attività laboratoriali un'applicazione che sarà in grado di risolvere di volta in volta una serie di problemi di varia natura e complessità
- al termine del corso un'applicazione server sottoporrà alle applicazioni client una serie di problemi selezionati fra quelli proposti durante il corso e in base alla correttezza o meno delle soluzioni e alle singole performance assegnerà i punteggi che definiranno una classifica intermedia
- ogni squadra invierà all'applicazione server anche il codice sorgente che verrà valutato in base alla sua chiarezza e alla presenza di documentazione appropriata
- queste valutazioni porteranno alla stesura di una classifica finale
- le migliori squadre otterranno un bonus di credito che contribuirà alla valutazione finale del corso



Programmazione di applicazioni software

PROGRAMMA ESTESO

- CPU
- registri
- memorie
- I/O
- canali di comunicazione

- gestione delle risorse
- gestione della memoria
- multiprogrammazione
 - processi
 - cooperazione
 - competizione
- thread

- incapsulamento
- composizione
- ereditarietà
- polimorfismo

- strutture dati lineari
- liste, pile, code
- alberi
- grafi

- gestione dei processi
- thread
- client/server
- peer to peer

- non sono obiettivi del corso ma strumenti per rendere effettivi i concetti
- Windows (perché ?)



- C++ (perché ?)

statcounter.com

Feb 2018	Feb 2017	Change	Programming Language	Ratings	Change
1	1		Java	14.988%	-1.69%
2	2		C	11.857%	+3.41%
3	3		C++	5.726%	+0.30%
4	5	▲	Python	5.168%	+1.12%
5	4	▼	C#	4.453%	-0.45%

tiobe.com

- Robert Sedgewick, “Algoritmi in C++”, terza ed., Addison-Wesley, 2003
- Carl Hamacher, Zvonko Vranesic, Safwat Zaky, “Introduzione all'architettura dei calcolatori”, seconda ed., McGraw-Hill, 2007
- Clifford A. Shaffer, Data Structures and Algorithm Analysis Edition 3.2 (C++ Version) [<https://people.cs.vt.edu/shaffer/Book/C++3elatest.pdf>]
- Anthony Williams, C++ Concurrency in Action: Practical Multithreading, Manning, 2012