

Informatica e Laboratorio di Programmazione Alberto Ferrari





- o Informatica e laboratorio di programmazione
- o crediti:
 - o **9** CFU (corrispondenti a circa 72 ore di lezione)
- o orario:
 - o Lunedì 15.30-17.30 aula 7 o lab. Informatica 1,2,3
 - o Mercoledì 8.30-10.30 aula 7
 - Venerdì 13.30-15.30 aula B
- o **docente**: Prof. Alberto Ferrari (alberto.ferrari@unipr.it)
 - o Phone: +39 0521 90 5708
 - o ricev.: Ingegneria Sede Scientifica Palazzina 1 venerdì 11:30-12:30 (su appuntamento)
- o *materiale*: https://albertoferrari.github.io/



syllabus - CFU 9 – 72 ore lezione

o introduzione alla programmazione

- o algoritmo risolutore esecutore
- o programmazione strutturata
- o collezioni e flussi di dati
- o funzioni e ricorsione
- o oggetti e astrazioni
- o interfacce grafiche

o introduzione all'informatica

- o rappresentazione dei dati
- o concetto di computazione
- o sistemi di elaborazione
- o sviluppo del software

introduzione all'informatica

- o rappresentazione dei dati
 - o numeri testi immagini suoni
- o teoria della computazione
 - o linguaggi automi e calcolabilità complessità
- o sviluppo del software
 - o ciclo di vita qualità e collaudo
- o sistemi di elaborazione
 - o calcolatori sistemi operativi

introduzione alla programmazione

- o algoritmi in Python 3
- o collezioni e flussi di dati
- o funzioni
- o oggetti e interfacce
- o linguaggio C++11
- o interfacce grafiche





- o lezioni in aula
- o soluzione guidata di esercizi in aula
- o esercizi di programmazione in laboratorio
 - o le esercitazioni in laboratorio sono centrali per il corso
 - o gli esercizi proposti vertono sugli stessi argomenti generali delle lezioni in aula





- o esercizi di base, in *Python*
- o programmazione ad oggetti e semplici progetti
 - o prima in *Python* e poi in *C*++
- attenzione: non sono sufficienti le poche ore in laboratorio per imparare a programmare!
 - o ripassare i contenuti
 - o completate *tutti* gli esercizi, a casa

If you wish to learn swimming you have to go into the water and if you wish to become a problem solver you have to solve problems. (George Polya)



istruzioni per accesso in laboratorio

- o verificare in anticipo il proprio account per l'accesso ai lab
 - http://www.cedi.unipr.it/gestioneaccounts
- o possibile programmare (esercitazioni e progetti) da soli o in coppia
 - o tassativamente non più di due
- o verifica esercitazioni
- o al termine di ogni esercitazione, *consegna* dei programmi in una cartella denominata con la propria *matricola*
- o periodicamente, lavori consegnati sottoposti a *valutazioni intermedie*, non preannunciate





- o libri di testo e **slide** del corso
- o esercizi risolti
- o durante le esercitazioni, assistenza di *tutor*
- o a lezione
 - o esempi prima delle esercitazioni
 - o soluzioni di esercizi delle settimane precedenti



modalità di verifica dell'apprendimento

o l'esame consiste in

- o una prova sui concetti teorici (quiz)
 - o calendario appelli
- o una prova di programmazione (lo sviluppo di un'applicazione software da sviluppare in laboratorio)
- o è previsto l'utilizzo di prove di valutazione in itinere
- o prova in appelli usuali (~3 ore)
- o le due prove si possono svolgere in tempi diversi e ciascuna resta valida per l'intero anno (fino ad ottobre)

o il *voto finale* è determinato:

- o per 1/4 dalla teoria
- o per 3/4 dalla programmazione



- o **Fondamenti di informatica e lab**. (A.A. 2014-2015). McGraw-Hill custom publishing. ISBN 978-13-082-4813-4 (25€, nelle librerie universitarie)
- A.B. Downey et al.: How to Think Like a Computer Scientist:
 Learning with Python 3, 3rd Edition http://openbookproject.net/thinkcs/
- o M. Beri: *Python*, Apogeo Pocket, 2010, 978-8850329151 (~8€)
- o C. S. Horstmann: *Fondamenti di C*++, McGraw-Hill Education, 2003, 978-8838661051 (~43€, 768pp.)





- o C. S. Horstmann: Fondamenti di C++, McGraw-Hill 2003, 978-8838661051 (~43€, 768pp.)
- S.B. Lippman, J. Lajoie, B.E. Moo: C++ Primer 5e, Addison Wesley 2012, 978-0321714114 (~36€, 940pp.)
- o M. Dawson: Python Programming for the Absolute Beginner 3e, Course Tech. 2010, 978-1435455009 (~25€, 450pp.)
- M. Dawson: Beginning C++ Through Game Programming 4e, Course Tech. 2014, 978-1305109919
 (~28€, 390pp.)
- o A. Lorenzi, V. Moriggia: Programmazione ad oggetti e linguaggio C++, Atlas 2004, 978-8826811956 (~12€, 380pp.)
- Oppure altro testo per le superiori
- MIT: Intro to CS and Programming in Python Video delle lezioni (https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/lecture-videos/index.htm)



- o Strumenti open source, multi-piattaforma
- o **Python 3**, battery included: http://python.org/
 - PyGame: http://pygame.org/
 - o Brython: http://brython.info/
- o *C*++11: http://isocpp.org/
 - o documentazione
 - o http://cplusplus.com/reference/
 - o http://cppreference.com/
 - o ambiente di sviluppo (CodeBlocks)
 - o http://www.codeblocks.org/



