



Introduzione al Corso

Luca Abeni

February 25, 2014



■ Contatti

- ◆ Luca Abeni `luca.abeni@unitn.it`
 - Per informazioni / ricevimento, email
- ◆ `http://www.disi.unitn.it/~abeni/IEP1`

■ Materiale per il corso

- ◆ Slide sul sito del corso
- ◆ Libri per chi è interessato:
 - Brian W. Kernighan e Dennis M. Ritchie, “Il Linguaggio C”, Pearson Prentice Hall

- 2 aspetti importanti: teoria e pratica
- **Aspetti Pratici**
 - ◆ Saper usare un computer
 - ◆ Saper scrivere ed eseguire programmi...
- **Aspetti “Teorici”**
 - ◆ Capire perché le cose funzionano in un certo modo...
 - ◆ ...per non rimanere sorpresi quando il computer “si comporta in modo strano”
 - Esempio: la somma di due numeri positivi... Da’ risultato negativo???

Organizzazione del Corso

- In generale, 2 lezioni di teoria alla settimana (più una di pratica dal 17 Marzo)
 - ◆ Teoria
 - Come funziona un computer
 - Come programmare un computer
 - ...
 - ◆ Pratica in laboratorio
 - Programmazione in C e octave

■ Quando:

- ◆ 2 appelli a Giugno/Luglio
- ◆ 2 appelli a fine Agosto/Settembre
- ◆ 2 appelli a Gennaio/Febraioio

■ Come:

- ◆ Prova pratica
 - Scrivere semplice programma
 - Se il programma funziona, prova passata
- ◆ Prova teorica
 - Solo dopo aver passato la prova pratica
 - Assegna il voto in trentesimi
 - Prova scritta, o usando la piattaforma Moodle

■ Struttura di un computer

- ◆ La macchina di Von Neumann
- ◆ Rappresentazione binaria dei dati
- ◆ L'algebra booleana

■ Concetti di base di programmazione

- ◆ Esempi di linguaggi
- ◆ Algoritmi, codifica e programmi

■ Il linguaggio C

- ◆ Costrutti di base, funzioni, puntatori, ...
- ◆ Implementazione, compilazione ed esecuzione di semplici algoritmi

- Struttura di un computer
- I linguaggi di programmazione
 - ◆ Linguaggi dichiarativi, imperativi, ...
 - ◆ Compilatori ed interpreti
 - ◆ Variabili e tipi di dati
 - ◆ Struttura di un programma
 - ◆ Cicli e costrutti di controllo del flusso
 - ◆ Funzioni / Procedure
 - ◆ Input / Output
- Stile di programmazione (programmazione modulare, riuso del codice, etc...)

- Struttura di un programma
- Tipi di dato
 - ◆ float, double, integer, char, ...
 - ◆ Rappresentazione dei dati nel computer
 - ◆ Regole di conversione
- Variabili
 - ◆ Dichiarazione, inizializzazione, accesso, ...
 - ◆ Visibilità e tempo di vita
- Funzioni di I/O

■ Operatori

- ◆ Operatori aritmetici e logici
- ◆ Operatori di confronto
- ◆ Regole di precedenza

■ Costrutti

- ◆ iterativi e condizionali

■ Tipi di dati strutturati

■ Concetti avanzati

- ◆ Puntatori, relazione fra puntatori ed array
- ◆ Allocazione dinamica della memoria

- Informatica:
 - ◆ **Scienza della rappresentazione e dell'elaborazione dell'informazione**
 - Studio degli algoritmi che descrivono e trasformano l'informazione
 - ◆ Oppure: **Scienza dei Computer** (Computer Science)
- Come funzionano i computer?
- Come si rappresenta l'informazione (in un computer)?
- Come istruire un computer per risolvere un determinato problema?

- Scienza della **rappresentazione dell'informazione**: come rappresenta l'informazione un computer?
 - ◆ Rappresentare numeri
 - ◆ Rappresentare testo
 - ◆ Rappresentare immagini
 - ◆ Rappresentare suoni
 - ◆ Rappresentare *algoritmi*
- Computer: insieme di circuiti elettronici
 - ◆ Tutto è 0 o 1...
 - ◆ ...numeri, testo, immagini, suoni, etc... Tutto va rappresentato come sequenza di 0 e 1!!!
 - ◆ Come operare su tali sequenze???

In Questo Corso...

■ Vedremo:

- ◆ Cos'è un computer
- ◆ Cos'è un algoritmo
- ◆ Cos'è un linguaggio di programmazione (esempio: C)

■ Nel dettaglio:

- ◆ Architettura (struttura di un computer, linguaggio macchina)
- ◆ Codifica dell'elaborazione
- ◆ Struttura di un algoritmo
- ◆ Dall'algoritmo al programma