

# Informatica - Mod. Programmazione

## Lezione 01

Prof. Giuseppe Psaila

Laurea Triennale in Ingegneria Informatica  
Università di Bergamo

## Corso: **Informatica**

- **Modulo di Programmazione**

Questo semestre

Prof. Giuseppe Psaila (il sottoscritto)

6 CFU

- **Modulo di Calcolatori Elettronici**

Secondo semestre, Prof. Giuseppe Coldani

6 CFU

L'intero corso di **Informatica** vale 12 CFU ed è considerato un unico corso, con un **unico** voto.

## Esami

- Gli esami vengono suddivisi nei due moduli, quindi svolti separatamente.
- L'intero corso è superato se per entrambi i moduli si è ottenuta una valutazione positiva.
- Voto per il singolo modulo in sedicesimi (quindi, voto massimo 16/16, sufficienza 9/16).
- Voto complessivo del corso da 12 CFU: somma dei voti ottenuti per i due moduli (oltre il 30, si ha la lode).

## Appelli

Sullo Sportello Internet trovate tre appelli diversi:

- Uno per lo scritto del Mod. di Programmazione (non utile per la verbalizzazione)
- Uno per lo scritto del Mod. di Calcolatori Elettronici (non utile per la verbalizzazione)
- Uno specifico per la verbalizzazione del voto complessivo

## Verbalizzazione

- Se prevedete di essere prossimi ad aver superato entrambi i moduli, prenotatevi all'appello per verbalizzare (non è fondamentale)
- Una volta superati entrambi i moduli, scrivetemi un e-mail,. chiedendomi di verbalizzare il voto complessivo.
- **ATTENZIONE**  
Non effettuo verbalizzazioni in automatico; siete voi a dover accettare il voto in modo esplicito (via e-mail)

- Cremonesi, Psaila  
Introduzione Ragionata al C/C++  
Esculapio, Progetto Leonardo
- Psaila  
Esercizi Ragionata in C/C++  
Esculapio, Progetto Leonardo
- Moriggia, Psaila  
Concetti Fondamentali di Informatica  
Esculapio, Progetto Leonardo

## Lezioni/Esercitazioni

- 32 ore di lezione
- 16 ore di esercitazioni,  
suddivisi in DUE squadre (vi daremo i criteri di  
suddivisione non appena avremo l'elenco degli  
immatricolati)
- 12 ore di tutorato tra gennaio e febbraio, per aiutarvi  
a preparare l'esame

## Esame

- **Modalità scritta**
- Orale facoltativo per migliorare il voto dello scritto (solo se lo scritto è sufficiente)
- Talvolta, propongo io l'orale nei casi vicini alla sufficienza

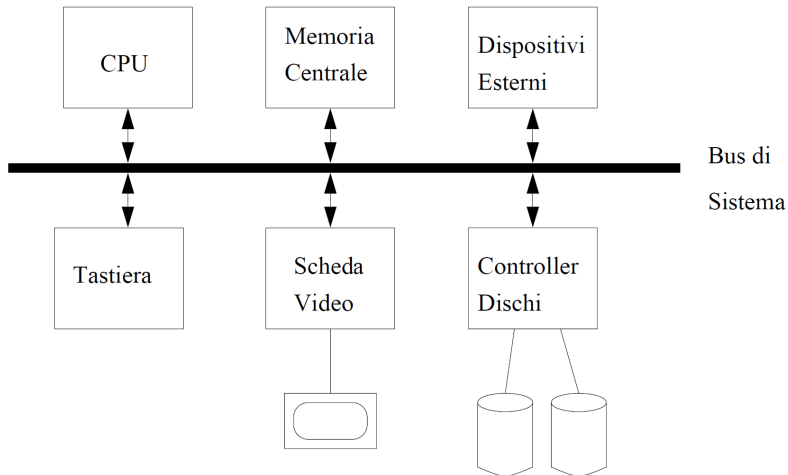


## Argomenti

- Nozioni di Base sui sistemi di elaborazione
- Programmazione Procedurale con il linguaggio C++
- Nozioni generali di teoria
  - Sistemi di numerazione
  - Architettura dei calcolatori e della CPU
  - Organizzazione dei sistemi operativi
  - Protocolli di rete

# Nozioni di Base

# Architettura di Von Neumann



- **CPU**

Central Processing Unit

Unità Centrale di Processo

Processore

- **Bus di Sistema**

- **Memoria Centrale**

RAM: Random Access Memory, volatile

ROM: Read-Only Memory, non modificabile

- **Memoria di Massa**

Dischi Magnetici, Dischi a Stato Solido

Persistente

- **Tastiera**

- **Schermo**

Scheda Video e Monitor

- **Dispositivi Esterni**

Controller USB (Universal Serial Bus)

Scheda di Rete

Altro

- Sistema posizionale

Ogni cifra può assumere due soli valori: 0 o 1 (bit, contrazione di **binary digit**)

- Valore del numero binario: si moltiplica ogni cifra per potenze crescenti del 2 da destra, partendo da  $2^0$
- Esempio:

$$\begin{aligned} 10011_2 &= 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = \\ &= 1 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = \\ &= 19_{10} \end{aligned}$$

Quanti bit vengono gestiti dal computer?

- 1 Byte = 8 bit
- L'intera macchina è organizzata per lavorare su singoli Byte o multipli
- Il Byte è l'unità minima di memorizzazione

- 1 KByte = 1024 Byte (Chilo-Byte)
- 1 MByte = 1024 KByte (Mega-Byte)
- 1 GByte = 1024 MByte (Giga-Byte)
- 1 TByte = 1024 GByte (Tera-Byte)
- 1 PByte = 1024 TByte (Peta-Byte)

Perché 1024?  $2^{10} = 1024$ , sono i valori rappresentabili con 10 bit



**Programma Applicativo**

**Sistema Operativo**

**Macchina Fisica**

- Microsoft Windows
- Linux (Unix)
- Apple MacOSX (basato su Linux)
- Google Android
- Apple iOS

- L'insieme di istruzioni riconosciute ed eseguite dal processore.
- Sono in formato binario, perché devono essere immediatamente riconosciute dal processore.
- Ovviamente, per un essere umano è molto scomodo programmare in questo modo.

- Detto anche Assembly o Linguaggio Assemblatore.
- Introduce dei simboli mnemonici che corrispondono alle istruzioni del linguaggio macchina
  - LOAD
  - STORE
  - SUM
- Occorre **TRADURRE** il programma scritto in assembler in linguaggio macchina.

- Sono linguaggi di programmazione che si allontanano dalla macchina e si avvicinano agli esseri umani.
- Linguaggi di programmazione **Procedurale**  
Consentono di specificare le procedure che devono essere eseguite
- Linguaggi di programmazione **a Oggetti**  
Consentono di organizzare il codice (programma) in modo più sofisticato.
- In questo corso, useremo la parte procedurale del C++.

- Il Linguaggio C nasce all'inizio degli anni '70.
- Si chiama C perchè:
  - I progettisti di Unix avevano bisogno di un linguaggio di programmazione per fare Unix.
  - Prima versione: Linguaggio A, con prima versione di Unix.
  - Seconda versione: Linguaggio B, con seconda versione di Unix.
  - Terza Versione: Linguaggio C, con versione finale di Unix
- Progettato per **correre**.
- Nel 1986 è stata proposta l'estensione a oggetti, chiamata C++.

- **Editore di Testi**

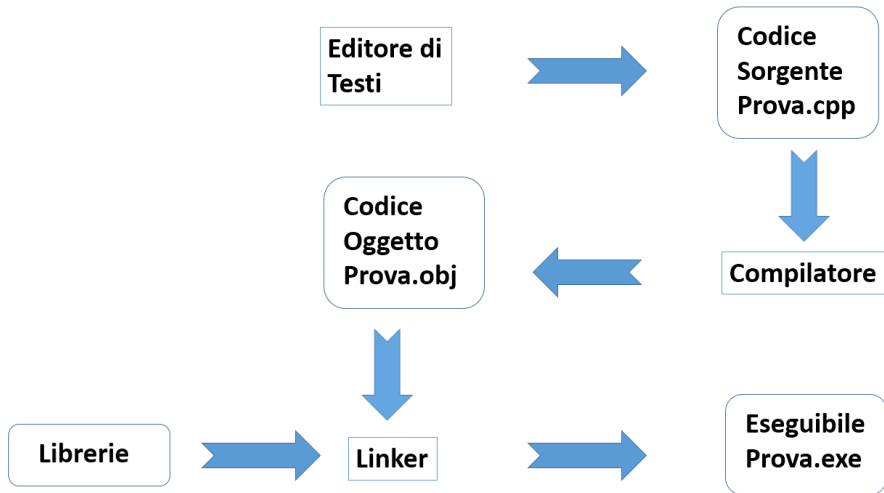
Consente di scrivere un file con il codice sorgente (nel linguaggio di programmazione)

- **Compilatore**

Traduce il codice sorgente nel linguaggio macchina (codice oggetto).

- **Linker (Collegatore)**

Completa il codice oggetto con i pezzi mancanti (detti Librerie) e prepara il **Codice Eseguitibile**.





- **IDE**

Integrated Development Environment

- Racchiude editor di testi, compilatore e linker in un unico strumento integrato.
- Dev-C++ (per Windows, usato nel corso).
- CodeBlock (per Windows).
- Microsoft Visual Studio (per Windows).
- Mac XCode (per chi ha il Mac).

- **BUG**

Il termine usato per indicare gli errori di funzionamento

- Ma **Bug** significa **Cimice**:

Il primo malfunzionamento della storia dei computer è stato causato da una cimice che metteva in corto i circuiti

- **Debugging**: l'attività volta a rimuovere i malfunzionamenti (nel codice, gli errori di programmazione)

- **Debugger**: lo strumento che supporta l'attività di debugging, normalmente integrato negli IDE.

# Microsoft Teams

- **Microsoft Account**

Se non lo avete già, create un *Microsoft Account* dal sito

`www.office.com`

usando l'account studenti di UniBG

- Teams è disponibile nella suite Office online, ma se preferite potete scaricarlo sul PC. Sono disponibili versioni per smartphone e tablet.

- **Il team del corso**

Si chiama

**[2023/2024] Mod. Programmazione**

- Se non lo trovate, potete usare il codice di ricerca o0g064s
- Se continuate a non trovarlo, vi sono due possibilità:
  - 1 Aspettate il giorno successivo e riprovate.
  - 2 Se ancora non lo trovate, scrivetemi dall'account Studenti UniBG (lo stesso usato per creare il Microsoft Account) e vi inserisco io.

Il mio indirizzo email:

`giuseppe.psaila@unibg.it`

- **I Canali**

Nel team troverete due canali:

- **Generale**, nel quale troverete tutto quello che riguarda le lezioni;
- **Esercitazioni**, nel quale troverete tutto quello che riguarda le esercitazioni.

- **Sezione FILE**

Ogni canale ha una sezione FILE, nella quale troverete il materiale del corso.

Per il canale **Generale**:

- 1 Cartella **Materiale del Corso**
- 2 Che a sua volta contiene
  - la cartella **Slide**,
  - la cartella **Esercizi**
  - la cartella **Compilatore**, che contiene il compilatore Dev-C++ per Windows.