

# Esercitazione di FdP-B

Lunedì, ore 15:30 - 17:30

# Short bio

Merenda Saverio Mattia



- Studente magistrale in Scienze Informatiche
- Altre info: [merendamattia.com](https://merendamattia.com)
- Contatti: [saveriomattia.merenda@studenti.unipr.it](mailto:saveriomattia.merenda@studenti.unipr.it)
- Elly 2024: <https://elly2024.didattica.unipr.it/course/view.php?id=2602>

# Qualche comando da sapere

- `ls -la`: elenca i file e le directory presenti nella directory corrente
- `mkdir <arg>`: crea una nuova directory
- `touch <arg>`: crea un nuovo file vuoto
- `rm -rf <args>`: elimina un file o una directory (**azione irreversibile**)
- `cp <arg> <arg>`: copia un file o una directory
- `mv <arg> <arg>`: sposta o rinomina un file o una directory
- `echo <arg>`: stampa il testo nella console
- Altri comandi: <https://www.freecodecamp.org/news/the-linux-commands-handbook/>

# Classe Prodotto in C++

## Gestione di un magazzino

- Implementare una classe Prodotto per un sistema di gestione magazzino applicando i concetti fondamentali della OOP in C++

# Classe Prodotto in C++

## Specifiche

La classe deve gestire:

- Codice identificativo univoco
- Nome/descrizione del prodotto
- Prezzo unitario
- Quantità disponibile in magazzino
- Stato di disponibilità (derivato dalla quantità)

# Classe Prodotto in C++

## Requisito 1 - Costruttori

- Un costruttore di default che inizializza a valori vuoti/zero
- Un costruttore parametrico che accetta solo il codice e il nome
- Un costruttore parametrico che accetta codice, nome, prezzo e quantità opzionale

# Classe Prodotto in C++

## Requisito 2 - Incapsulamento

- Tutti gli attributi devono essere privati
- Implementare metodi getter/setter appropriati
- I metodi getter devono essere dichiarati come const
- I setter devono includere controlli di validità (es. prezzo non negativo)

# Classe Prodotto in C++

## Requisito 3 - Overloading di funzione

Implementare:

- Due versioni del metodo `aggiungiQuantita`:
  - Una senza parametri (incrementa di 1)
  - Una con parametro intero (incrementa del valore specificato)
- Fare la stessa cosa con il metodo `rimuoviQuantita`



# Classe Prodotto in C++

## Requisito 4 - Overloading operatori

Implementare:

- Operatore di output (<<) che visualizza tutte le informazioni del prodotto
- Operatore di somma (+) che combina due prodotti che hanno lo stesso codice, somma le quantità disponibili e restituisce un nuovo oggetto Prodotto
- Operatori di confronto (==, !=)
- Operatore di incremento infisso e postfisso (++)