

PROGRAMMAZIONE AVANZATA

0 Organizzazione

- Cosa si studia e perché
- Struttura del corso
- Libri e materiale didattico
- Strumenti per programmare in C++
- Avvertimenti
- Esame = progetto + discussione
- Come e quando contattare il docente



Documento distribuito con licenza [CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/). Generato il 22/02/2022.

Cosa si studia e perché

Corso “Basi di informatica”: paradigma **imperativo**

- programma = sequenza di **comandi** che alterano lo **stato** del calcolatore

Corso “Programmazione avanzata”: paradigma **orientato agli oggetti**

- programma = collezione di **oggetti** che **comunicano** tra loro
- oggetto = rappresentazione (privata) + operazioni (pubbliche)
- comunicazione = uso delle operazioni sugli oggetti

Non basta il paradigma imperativo?

- In teoria, sì.
- In pratica, è utile adottare un approccio più organizzato che agevoli la scomposizione di programmi complessi in **moduli indipendenti**, il **riuso** e la **modifca** del codice.
- Il paradigma di programmazione a oggetti è uno di questi approcci.

Struttura del corso

Lezioni

- **8 lezioni** in laboratorio (mercoledì 13:30 – 16:30)
- ogni lezione è divisa in **due parti**
 1. introduzione di concetti (docente attivo)
 2. risoluzione degli esercizi (docente passivo)

Attenzione

- L'unico modo di imparare un linguaggio (di programmazione) è **usarlo**
- Leggere libri o codice scritto da altri **non basta**
- La parte di risoluzione degli esercizi è fondamentale

Libri e materiale didattico

Testi di riferimento

1. Allen B. Downey, Think C++ (capitoli 11-15), disponibile gratuitamente [qui](#).
2. Bjarne Stroustrup, Programming: Principles and Practice Using C++, Addison Wesley, 2014.
3. Bjarne Stroustrup, The C++ Programming Language (fourth edition), Addison Wesley, 2013.

Note

- Nel corso vedremo **pochi concetti** e **tanti esempi**
- Non c'è un libro che seguo fedelmente o che è fondamentale per l'esame
- [3] è riferimento ufficiale ma **non** una introduzione alla programmazione

Altro materiale

- Slide (HTML e PDF) disponibili su Moodle
- Possono subire aggiornamenti man mano che il corso procede, se volete stamparle fatelo di settimana in settimana in base agli argomenti trattati

Strumenti per programmare in C++

Linux

- Installare GCC con il package manager della distribuzione

MacOS

- Installare XCode (anche solo linea di comando)

```
$ xcode-select --install
```

Windows

- Installare [MinGW](#) oppure [Visual Studio C++](#)

Ambienti di sviluppo

- Ambienti integrati: XCode, Visual Studio, ... non li uso e non li conosco
- Editor “intelligenti”: [Visual Studio Code](#), [Atom](#), [Sublime](#), ...
- Scegliete gli strumenti più comodi per voi, editor/compiler già usati per il corso di Basi di Informatica vanno bene

Avvertimenti

- Il linguaggio C++ è in continua evoluzione da circa 3 decenni
- A seconda del compilatore usato alcune caratteristiche del linguaggio (e della sua libreria) possono non essere disponibili
- È possibile che alcune caratteristiche che illustro a lezione non siano disponibili con il vostro compilatore
- In caso di problemi, consultare il documento con [domande & risposte](#) e/o contattare il docente

Esame = progetto + discussione

Progetto

- individuale o di gruppo (max 2 persone)
- se fatto in gruppo, i contributi dei singoli devono essere **equi** e **facilmente individuabili** (indicativamente si chiede lo sviluppo di una classe per ogni membro del gruppo, vedremo a breve cosa si intende per classe)
- tema del progetto da concordare con il docente (farò proposte)
- poche cose fatte bene (una classe + un contenitore)

Discussione

- sempre **individuale**
- dimostrare di conoscere e saper spiegare **tutto** il codice (non solo la propria classe)
- saper apportare semplici **modifiche** al volo (non solo nella propria classe)

Come e quando contattare il docente

In aula

- Richiamando la mia attenzione quando volete

Via email

- Scrivendo a luca.padovani@unito.it dal vostro **indirizzo istituzionale**

A ricevimento

- Online nella [mia stanza Webex](#) previo appuntamento (inviare email)

Precisazioni

- Rispondo via email solo a domande **semplici** e tipicamente **chiuse: no** correzione esercizi, **no** “non ho capito l’argomento X”, **no** “questo codice non funziona”, ecc.
- **Non** rispondo via email né effettuo ricevimento nei **tre giorni lavorativi** precedenti un appello d’esame