



UNIVERSITÀ
degli STUDI
di CATANIA

Introduzione al linguaggio C++

Corso di programmazione I AA 2020/21

Corso di Laurea Triennale in Informatica

Prof. Giovanni Maria Farinella

Web: <http://www.dmi.unict.it/farinella>

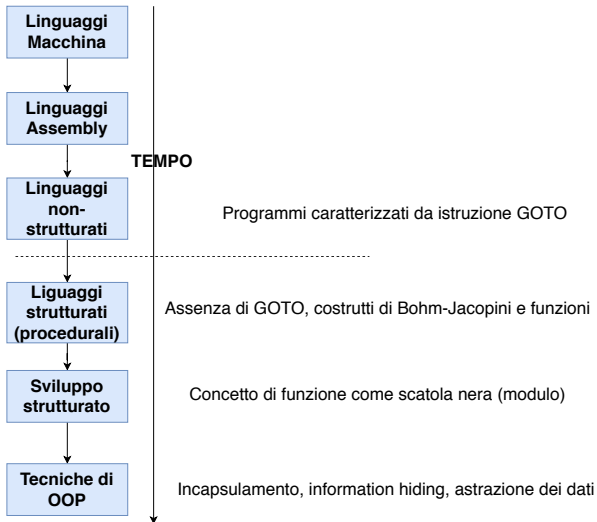
Email: gfarinella@dm.unict.it

Dipartimento di Matematica e Informatica

C++ (OOP) estensione del linguaggio C basato sul paradigma OOP (Object Oriented Programming):

- (+) Semplicità ed efficienza (dal C).
- (+) Introduzione di caratteristiche tipiche dei linguaggi orientati agli oggetti (classi, costruttori, template, etc).
- (+) Facile lettura, manutenzione, modifica espansione del codice.
- (+) Prototipazione veloce del software.

Il linguaggio C++



Assembly

Sostituiscono i numeri con le parole.

Procedurali

Nascondono la complessità delle operazioni sui dati.

OOP

Nascondono i dati e la complessità del programma.

Modello di computazione

Assembly

Programmatore deve conoscere **l'architettura** della macchina (quali registri, operazioni supportate etc).

Procedurali

Si modella un **diagramma di flusso** che rappresenta il movimento dei **dati** tra svariate **funzioni o procedure**.

OOP

Il sistema è costituito da **oggetti** che si scambiano **messaggi**.

Esecuzione di un programma

Assembly

Operazioni aritmetico logiche, si **spostano dati** tra la memoria e i **registri** della CPU.

Procedurali

Invocazione di funzioni su blocchi di dati.

OOP

Scambio di **messaggi tra oggetti**.

- Con C++ (ed in generale un linguaggio OOP) la **progettazione di interfacce grafiche** diviene più agevole (polimorfismo, information hiding, etc).
- **In C non esiste reale incapsulamento**, quindi la modifica della rappresentazione dei dati influenza anche altre parti del programma, introducendo **maggiore probabilità di errore**.
- C++ favorisce la **modularità**
- C++ permette di scrivere programmi maggiormente **mantenibili**

Incapsulamento

Membri della classe (dati e metodi) possono essere **pubblici o privati**.

In tal modo dati e metodi utili al funzionamento interno (implementazione) possono essere **nascosti all'esterno**.

Astrazione

Classi astratte in C++ e Interfacce in Java.

Concetti base della OOP

ADT (Abstract Data Type)

Definizione di dati astratti.

Information Hiding

Nascondere dettagli implementativi ed esporre solo l'interfaccia.

Incapsulamento, astrazione e ADT permettono di realizzare Information hiding.

Ereditarietà

Definizione di **gerarchie di classi**, aiuta ad implementare il modello corrispondente al problema da risolvere.

Motore (classe base) \leftarrow {MotoreDiesel, MotoreBenzina}
(classi derivate)

Interfaccia di classe motore **comune** alle due classi **derivate**.

Ma comportamento/informazioni differenti.

Polimorfismo

Polimorfismo: *assumere forme, aspetti, modi di essere diversi secondo le varie circostanze.*

Il chiamante opera su un'unica interfaccia (ES: tipo/classe Motore, metodo info())

Il chiamante, può ottenere risultati differenti a seconda del tipo di motore (diesel o benzina).

Ereditarietà e astrazione consentono di realizzare polimorfismo.