



---

# ADVANCED PROGRAMMING LANGUAGES

ING-INF/05 - 9 CFU - 1° semestre

## Docenti titolari dell'insegnamento

### VINCENZA CARCHIOLO

**Email:** vincenza.carchiolo@unict.it

**Edificio / Indirizzo:** Edificio 3 - piano V - stanza 11

**Telefono:** 095 7382359

**Orario ricevimento:** martedì e venerdì dalle ore 11 alle ore 12

### GIUSEPPE MANGIONI

**Email:** giuseppe.mangioni@dieei.unict.it

**Edificio / Indirizzo:** Viale A. Doria, 6 - Edificio 13

**Telefono:** 0957382380

**Orario ricevimento:** giovedì 9:00 - 11:00

---

## OBIETTIVI FORMATIVI

### Conoscenza e capacità di comprensione (Knowledge and understanding abilities)

- conoscenza dei diversi paradigmi di programmazione
- conoscenza delle diverse tecniche di gestione della memoria
- conoscenza delle diverse tecniche di traduzione
- conoscenza dei metodi per la gestione di comunicazione e concorrenza
- conoscenza dei metodi per la gestione degli errori

### Capacità di applicare conoscenza e comprensione (Applying knowledge and understanding abilities)

- capacità di scrivere programmi in C++
- capacità di sviluppo di programmi in C#
- capacità di sviluppo di programmi in Python
- capacità di risolvere in Go

### Autonomia di giudizio (Ability of making judgements)

- lo studente acquisisce la conoscenza di diversi linguaggi e paradigmi di programmazione ed è in

grado di scegliere il migliore in base al problema da risolvere

#### Abilità comunicative (Communication skills)

- lo studente è in grado di valutare il linguaggio più adatto per l'implementazione di un'applicazione che risolve un determinato problema

#### Capacità di apprendimento (learning skills):

- Introdurre i principali costrutti dei linguaggi di programmazione e di fornire gli strumenti necessari per la valutazione critica dei linguaggi di programmazione.
- Acquisire approfondite conoscenze sulle strutture dei linguaggi di programmazione che consentano di comprendere l'effettiva funzionalità nei diversi campi di applicazione.

### **MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO**

Il corso è organizzato in moduli..

Il corso prevede come metodo di insegnamento principale le lezioni frontali per acquisire le conoscenze teoriche di base e tutti gli elementi sintattici e lo svolgimento di esercitazioni proposte dal docente per acquisire la capacità di risolvere i problemi, applicare la conoscenza e utilizzare gli ambienti e le metodologie di sviluppo.

Il docente propone, inoltre, delle esercitazioni individuali che consistono nella soluzione di un problema, che lo studente deve affrontare in autonomia che vengono successivamente corrette o discusse in classe.

***Le lezioni potranno essere tenute in modalità mista e/o a distanza e potranno essere inserite delle modifiche in linea con il programma del presente syllabus.***

### **PREREQUISITI RICHIESTI**

Conoscenza delle architetture dei calcolatori base e le relative problematiche, i linguaggi assembly.

Conoscenze di base di Informatica e di programmazione ad oggetti.

Conoscenze di base di ingegneria del software.

---

### **FREQUENZA LEZIONI**

La frequenza non è obbligatoria ma è fortemente consigliata.

---

### **CONTENUTI DEL CORSO**

**Modulo 1: Caratteristiche dei linguaggi di programmazione e Linguaggi formali per la specifica e la traduzione dei linguaggi di programmazione**

Evoluzione dei principali linguaggi di programmazione

Tipi di dati - Espressioni e dichiarazioni di assegnazione - Strutture di controllo a livello di dichiarazione - Sottoprogrammi e loro implementazione - Gestione della memoria - Garbage collector - Exception e event handler

## **Modulo 2: IL linguaggio C++**

Introduzione al C++, utilizzo di classi predefinite, creazioni di classi in C++, puntatori e riferimenti, overloading delle funzioni e degli operatori, creazione di oggetti a tempo di esecuzione, reimpiego di codice in C++, scrittura di programmi estensibili, argomenti e valori di ritorno, classi contenitore e modelli in C++, gestione delle eccezioni, Standard C++ Library, STL

## **Modulo 3: Programmazione funzionale orientata agli oggetti: il linguaggio C#:**

Sintassi C#: Classes and Objects, Basic Types and Operations, Functional Objects Built-in Control Structures, Functions and Closure, Astrazione, Ereditarietà e gerarchia delle classi, Lambda Calculus, Liste, Pattern Matching, Programmazione ad eventi, Concorrenza, GUI

## **Modulo 4: Il Linguaggio Python:**

Introduzione a Python, Strutture dati, Stringhe, Funzioni avanzate e OOP, Standard Library, Strumenti di sviluppo, Networking, Crawling and Scraping, Serializzazione e persistenza dei dati, Programmazione GUI, Distributing Python

## **Modulo 5: Il linguaggio GO**

Introduzione al linguaggio Go, Sintassi del linguaggio :Tipi dati e variabili e Costrutti di controllo e decisione, Strutture dati: array, slice e map, Funzioni e defer Struttura della memoria e puntatori Programmazione Object-Oriented, P, Gestione dell'Input e Output su terminale e file, Il compilatore e il garbage collector

---

## **TESTI DI RIFERIMENTO**

[T1] Materiale fornito del docente on line

[T2] Sebesta, Concepts of Programming Languages, 11th Edition - Pearson

[T3] Mark Michaelis, Essential C# 8.0, Addison-Wesley Microsoft Technology English

[T4] Alan A. Donovan and [Brian W. Kernighan](#), Go Programming Language, Addison-Wesley Professional Computing Series

[T5] Thinking in C++, Vol 1 Thinking in C++, Bruce Eckel

## **ALTRO MATERIALE DIDATTICO**

Microsoft team canale del corso

---

## PROGRAMMAZIONE DEL CORSO

Argomenti	Riferimenti testi
1 Valutazione dei linguaggi di programmazione	[T2] 1.1-1.4 (inclusive), 1.6
2 Storia dei linguaggi di programmazione	[T2] 2
3 Naming, Binding e Scope	[T2] capitolo 5
4 Data Type e Valutazione delle Espressioni	[T2] capitolo 6, 7
5 Gestione della memoria e Garbage Collection	Aho Ulmann, Compilers: Principles, Techniques, and Tools, 2nd Edition, cap 7.4 -7.8
6 Strutture di controllo e sottoprogrammi	[T2] capitolo 8,9,10
7 Tecniche avanzate di programmazione	[t2] capitolo 13,14
8 Il linguaggio C++	
9 Il linguaggio C#	
10 Il linguaggio Python	
11 Il linguaggio GO	

---

## VERIFICA DELL'APPRENDIMENTO

### MODALITÀ DI VERIFICA DELL'APPRENDIMENTO

E' previsto lo sviluppo di un elaborato assegnato dal docente e una prova orale successiva alla discussione dell'elaborato.

Lo studente dovrà compilare un form di richiesta di assegnazione dell'elaborato contenente alcune informazioni sulla base delle quali il docente procederà all'assegnazione dell'elaborato.

La richiesta dovrà essere effettuata durante il periodo delle lezioni e l'elaborato assegnato potrà essere presentato entro l'anno accademico.

La consegna dell'elaborato (codice e relazione) dovrà essere sottomessa attraverso il portale Microsoft Team entro la data concordata al momento dell'assegnazione dell'elaborato.

L'elaborato dovrà essere sviluppato da gruppi di 2 studenti. Solo in casi motivati sarà possibile sviluppare l'elaborato in modo individuale.

**L'esame sarà organizzato in due parti che si svolgeranno nello stesso giorno.**

**La prima riguarderà la discussione dell'elaborato svolto, la seconda preve una discussione su l'intero programma e verrà richiesta la capacità di sviluppare codice nei 4 linguaggi previsti.**

**La prima parte dell'esame avrà un valore del 25% mentre la seconda parte del restante 75%.**

**La rinuncia allo sviluppo dell'elaborato consentirà il raggiungimento di una votazione massima di 23.**

**ESEMPI DI DOMANDE E/O ESERCIZI FREQUENTI**

Disponibili su Microsoft Team

---