



UNIVERSITÀ  
degli STUDI  
di CATANIA

DIPARTIMENTO DI INGEGNERIA ELETTRICA ELETTRONICA E  
INFORMATICA

Corso di laurea magistrale in Ingegneria informatica

Anno accademico 2018/2019 - 2° anno

---

# 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

Gli obiettivi principali sono quello di introdurre i principali costrutti dei linguaggi di programmazione e di fornire gli strumenti necessari per la valutazione critica dei linguaggi di programmazione.

Un secondo obiettivo è quello di fornire approfondite conoscenze sulle strutture dei linguaggi di programmazione che consentano di comprendere l'effettiva funzionalità nei diversi campi di applicazione.

Il corso illustra i linguaggi di programmazione fornendo una tassonomia basata sulle applicazioni. Saranno studiati i linguaggi formali e alcuni linguaggi di programmazione orientati ad applicazioni specifiche quali cloud e Big Data. Saranno studiate le tecniche avanzate dei moderni linguaggi di programmazione con particolare attenzione ai linguaggi multi paradigma, all'uso dei tipi, al type checking dinamico, programmazione concorrente. Tali tecniche saranno analizzate nel contesto dei seguenti linguaggi: Scala, R, C++, GO.

## 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.

## **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

---

## **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 - Linguaggi formali per la descrizione del lessico della sintassi e della semantica dei 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; Programmazione funzionale orientata agli oggetti: il linguaggio SCALA**

SINTASSI SCALA: 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, Attori e concorrenza, GUI

### **Modulo 3: 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 4: 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

### **Modulo 5: il linguaggio R**

Introduzione al Linguaggio R, Sintassi del linguaggio R, arrays, matrici e data frames. Uso e definizione di procedure, funzioni e packages. Vectorisation, loops, control structures (if, while, for), Ottimizzazione non-lineare e convergenza. I/O e visualizzazione

---

## TESTI DI RIFERIMENTO

[T1] Materiale fornito del docente on line

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

[T3] Martin Odersky, Lex Spoon, and Bill Venners: Programming in Scala, Third Edition, Artima

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

[T5] Norman Matloff, The Art of R Programming, ISBN-13: 978-1-59327-384-2

[T6] Thinking in C++, Vol 1 Thinking in C++, Bruce Eckel

## ALTRO MATERIALE DIDATTICO

studium.unict.it

---

## 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 Descrizione della sintassi e della semantica di un linguaggio di programmazione	[T2]+ capitolo 3
4 Analisi lessicale e sintattica	[T2] capitolo 4
5 Naming, Binding e Scope	[T2] capitolo 5
6 Data Type e Valutazione delle Espressioni	[T2] capitolo 6, 7
7 Gestione della memoria e Garbage Collection	Aho Ulmann, Compilers: Principles, Techniques, and Tools, 2nd Edition, cap 7.4 -7.8
8 Strutture di controllo e sottoprogrammi	[T2] capitolo 8,9,10
9 Tecniche avanzate di programmazione	[t2] capitolo 13,14

---

10	SCALA: Sintassi Classes and Objects, Basic Types and Operations, Functional Objects Built-in Control Structures, Functions and Closures	[T1] CAP1, 2, 3, 4,5,6,7 e 8
11	SCAIA: Astrazione, Ereditarieta e gerarchia delle classi	[T3] CAP da 9 a 13
12	SCALA: Liste	[T3] cap 16
13	SCALA: Pattern Maching	[T3] cap 15
14	SCALA: Attori e concorrenza	[T3] cap 30
15	SCALA: GUI	[T3] cap 32
16	C++	
17	GO: Sintassi,	
18	Sintassi del Linguaggio R	

## 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 procedera all'assegnazione dell'elaborato.

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

La consegna dell'elaborato (codice e relazione) dovra essere sottomessa almeno 7 giorni prima dell'esame attraverso il portale Studium.

L'elaborato dovra essere sviluppato da gruppi di 2 studenti. Solo in casi motivati sara possibile sviluppare l'elaborato in modo individuale. La discussione dell'elaborato dovra essere effettuata in modo congiunto dai componenti del gruppo.

### ESEMPI DI DOMANDE E/O ESERCIZI FREQUENTI

Disponibili su Studium