

# INFORMATION TECHNOLOGY ENGINEERING (LB08)

(Lecce - Università degli Studi)

## Teaching FOUNDATION OF COMPUTER SCIENCE

GenCod A000015

Owner professor Italo EPICOCO

**Teaching in italian** FONDAMENTI DI INFORMATICA

**Teaching** FOUNDATION OF COMPUTER SCIENCE

**SSD code** ING-INF/05

**Reference course** INFORMATION TECHNOLOGY ENGINEERING

**Course type** Laurea

**Credits** 9.0

**Teaching hours** Front activity hours: 81.0

**For enrolled in** 2022/2023

**Taught in** 2022/2023

**Course year** 1

**Language** ITALIAN

**Curriculum** PERCORSO COMUNE

**Location** Lecce

**Semester** First Semester

**Exam type** Oral

**Assessment** Final grade

**Course timetable**

<https://easyroom.unisalento.it/Orario>

## BRIEF COURSE DESCRIPTION

Il corso mira a fornire sia elementi di teoria dell'informatica che i principi e le basi per la programmazione. Partendo dal concetto di macchina programmabile in grado di svolgere operazioni seguendo una sequenza codificata di istruzioni elementari si passa alla strutturazione di un algoritmo e quindi alla competenze che permettono di scomporre un problema "complesso" per la macchina in una sequenza di operazioni "elementari". Durante il corso verranno inoltre studiati gli algoritmi di ordinamento e di ricerca più noti con l'obiettivo di illustrare come differenti scelte nella risoluzione di uno stesso problema possano incidere sull'efficienza di un algoritmo e sui tempi di esecuzione al calcolatore. Si affronteranno infine elementi sull'architettura dei calcolatori. Strutturare i dati in modo opportuno è uno dei principali fattori che influenzano l'efficienza di un algoritmo. Durante il corso si studieranno le strutture dati base quali le liste. Infine il corso fornirà gli strumenti per lo sviluppo di applicazioni attraverso ambienti di programmazione integrati IDE utilizzando il linguaggio C e alcuni cenni sull'analisi della complessità computazionale degli algoritmi.

## REQUIREMENTS

Non vi è alcuna propedeuticità.

---

## COURSE AIMS

### **Conoscenze e comprensione**

I risultati attesi di apprendimento prevedono che al termine del corso gli studenti siano in grado di:

- 1) illustrare le principali caratteristiche dell'architettura di un calcolatore;
- 2) comprendere le metodologie di rappresentazione e codifica dell'informazione;
- 3) progettare e implementare un algoritmo;
- 4) riconoscere ed applicare gli elementi caratterizzanti un linguaggio di programmazione procedurale strutturato (tipi dato, strutture di controllo, funzioni e procedure, gestione dei file e strutture dinamiche);
- 5) riconoscere le principali strutture dati dinamiche (liste, code, alberi) e i principali algoritmi di ordinamento e di ricerca;
- 6) analizzare, interpretare, comprendere e produrre listati di codice in linguaggio C utilizzando degli appositi ambienti di programmazione (IDE).
- 7) comprendere l'architettura hardware dei calcolatori

### **Capacità di applicare conoscenze e comprensione**

Gli studenti saranno in grado di applicare le conoscenze acquisite in diversi ambiti applicativi ed in generale per la risoluzione al computer di processi e sistemi.

### **Autonomia di giudizio**

Il corso favorisce l'autonomia di giudizio degli studenti attraverso l'analisi critica di problemi computazionali per i quali trovare soluzioni software in grado di risolverlo. Diverse soluzioni proposte dagli studenti sono poste a confronto e valutate criticamente dagli studenti stessi.

### **Abilità comunicative**

È fondamentale che gli studenti siano in grado di comunicare con un pubblico vario e composito, non omogeneo culturalmente, in modo chiaro, logico ed efficace, utilizzando gli strumenti metodologici acquisiti e le loro conoscenze scientifiche e, in particolar modo, il lessico di specialità. Il corso favorisce lo sviluppo delle abilità inerenti le capacità di esporre in termini precisi e formali le proprietà e le caratteristiche di algoritmi e la descrizione di possibili soluzioni algoritmiche a problemi reali.

---

## TEACHING METHODOLOGY

Le lezioni teoriche si svolgeranno in aula utilizzando slide con esempi alla lavagna. Le esercitazioni si svolgeranno in laboratorio informatico con esercizi svolti direttamente al computer inerenti la programmazione in linguaggio C.

Le lezioni saranno inoltre registrate sulla piattaforma Microsoft Teams

Le slide del corso saranno disponibili sulla piattaforma formazioneonline all'indirizzo: <https://formazioneonline.unisalento.it/course/view.php?id=717>

Utilizzare le proprie credenziali di ateneo per accedere alla piattaforma formazioneonline; successivamente si usi la password "fondainfo1" per accedere ai contenuti del corso di Fondamenti di Informatica I

---

## ASSESSMENT TYPE

L'esame è composto da una prova scritta durante la quale si verificheranno le capacità acquisite nella progettazione di un algoritmo, nel riconoscimento delle principali strutture dati e degli algoritmi di base visti a lezione e nell'analisi della complessità computazionale di un algoritmo oltre agli aspetti teorici affrontati nel corso. Durante la prova scritta non sarà consentito l'uso di manuali né altro materiale di supporto.

Oltre alla prova scritta ci sarà una prova orale/programmazione alla quale si accede solo dopo aver superato la prova scritta. Durante la prova di programmazione verrà valutata la capacità pratica nello sviluppare un semplice programma eseguibile al calcolatore in linguaggio C. La prova di programmazione dura 1 ora, il programma realizzato durante la prova di programmazione verrà quindi discusso appena terminata la prova. Durante la prova di programmazione si può usare il proprio computer e tutti i manuali e le fonti online utili a risolvere l'esercizio assegnato.

---

## ASSESSMENT SESSIONS

Le date degli appelli d'esame sono disponibili al seguente link: [calendario esami di profitto](#)

---

## FULL SYLLABUS

- Architettura di Von Neumann, concetto di algoritmo, macchina astratta (4 ore)
  - Rappresentazione delle informazioni (10 ore)
  - Porte logiche e algebra booleana (2 ore)
  - Reti combinatorie (4 ore)
  - Bistabili e reti sequenziali (4 ore)
  - Processore e unità aritmetico-logica (4 ore)
  - Gerarchia della memoria (4 ore)
  - Pipeline e parallelismo a livello di istruzione (4 ore)
  - Linguaggio assembler (5 ore)
  - Strutturare un algoritmo (6 ore)
  - Costrutti del linguaggio C, variabili e tipi di dato, funzioni, array puntatori e gestione dinamica della memoria (12 ore)
  - Funzioni ricorsive (4 ore)
  - Tipi di dato astratto: Pila, Coda, Lista, Heap Tree (8 ore)
  - Cenni su analisi della complessità computazionale (2 ore)
  - Algoritmi di ordinamento: insertion sort, selection sort, bubble sort, merge sort, quick sort, heap sort (8 ore)
- 

## REFERENCE TEXT BOOKS

- [1] B. W. Kernighan, D. M. Ritchie, "Il linguaggio C. Principi di programmazione e manuale di riferimento. Ediz. MyLab. Con Contenuto digitale per download e accesso on line", ISBN-13: 978-8891908230
- [2] Dino Mandrioli, Stefano Ceri, Licia Sbattella, Paolo Cremonesi, Gianpaolo Cugola, "Informatica: Arte e Mestiere 4° Ed.", McGraw-Hill,, 2014. ISBN: 9788838668487.
- [3] Dispense fornite dal docente presenti nella piattaforma formazioneonline (si rimanda alla sezione "metodi didattici" per l'indicazione del link alla piattaforma)