

Master

Informatica e Linguaggi di Programmazione

Partnership



Association
for Computing
Machinery

tech global
university



Master

Informatica e Linguaggi di Programmazione

- » Modalità: online
- » Durata: 12 mesi
- » Titolo: TECH Global University
- » Accreditamento: 60 ECTS
- » Orario: a tua scelta
- » Esami: online

Accesso al sito web: www.techtitude.com/it/informatica/master/master-informatica-linguaggi-programmazione

Indice

01

Presentazione del programma

pag. 4

02

Perché studiare in TECH?

pag. 8

03

Piano di studi

pag. 12

04

Obiettivi didattici

pag. 24

05

Opportunità professionali

pag. 30

06

Licenze software incluse

pag. 34

07

Metodologia di studio

pag. 38

08

Personale docente

pag. 48

09

Titolo

pag. 52

01

Presentazione del programma

La continua evoluzione dell'intelligenza artificiale, del *machine learning* e dell'elaborazione del linguaggio ha trasformato il panorama dell'informatica a livello globale. Questi progressi richiedono professionisti in grado di comprendere sia i fondamenti teorici che le applicazioni pratiche dei linguaggi computazionali. Secondo l'UNESCO, il 90% dei posti di lavoro nel prossimo futuro richiederà competenze digitali avanzate. Questa realtà ha aumentato la necessità di programmi specializzati che integrino le conoscenze in algoritmi, strutture dati e semantica dei linguaggi. In questo contesto, TECH presenta una qualifica innovativa e completamente online in Informatica e Linguaggi di Programmazione, orientata a fornire una preparazione accademica solida e aggiornata in un ambiente digitale e flessibile.



“

Un programma completo e 100% online, esclusivo di TECH e con una prospettiva internazionale supportata dalla nostra partnership con l'Association for Computing Machinery"

In un ambiente tecnologico sempre più caratterizzato da automazione, intelligenza artificiale ed elaborazione del linguaggio naturale, capire come sono costruiti, interpretano e ottimizzano i linguaggi di calcolo è diventata una necessità chiave per avanzare in molteplici aree della conoscenza e dell'industria. L'interazione uomo-macchina, l'efficienza dei sistemi informatici e la creazione di nuovi strumenti digitali dipendono in larga misura dalla padronanza di questi linguaggi.

Questo programma offre una visione completa dei pilastri su cui si basano l'informatica e i linguaggi di programmazione, consentendo di acquisire competenze specialistiche che potenziano il pensiero logico, la risoluzione di problemi complessi e la progettazione di soluzioni innovative. Vengono affrontate tematiche come la semantica formale, la teoria dei linguaggi, gli automi, la compilazione e altri elementi fondamentali per comprendere il funzionamento interno dei sistemi informatici.

Inoltre, padroneggiare questi contenuti dà accesso a opportunità in settori altamente dinamici come l'intelligenza artificiale, lo sviluppo di linguaggi di programmazione, il calcolo quantistico o l'ingegneria del software. Questa specializzazione rappresenta un passo strategico per coloro che desiderano avanzare nella loro carriera in ambito accademico, tecnico o professionale, soprattutto nei contesti in cui sono valutati profili con una solida base teorica e capacità analitica. La profondità concettuale acquisita consente non solo di applicare le tecnologie esistenti, ma anche di progettare quelle future.

La modalità online di questo programma consente l'accesso a contenuti aggiornati e di alto livello accademico da qualsiasi luogo, senza compromettere la qualità o le esigenze. Questa flessibilità facilita la compatibilità con altri progetti professionali o accademici e promuove un apprendimento autonomo, rigoroso e orientato all'analisi critica. Attraverso una piattaforma interattiva, materiali multimediali e l'accompagnamento di esperti, è garantita un'esperienza arricchente che risponde alle attuali esigenze del settore digitale e accademico. Inoltre, un rinomato Direttore Ospite Internazionale offrirà 10 approfondite *Master class*.

Inoltre, grazie al fatto che TECH è membro dell'**Association for Computing Machinery (ACM)**, lo studente avrà accesso a risorse esclusive e aggiornate come pubblicazioni scientifiche, corsi specializzati e conferenze internazionali. Inoltre, avrà l'opportunità di ampliare la rete di contatti, connettendosi con esperti in tecnologia, intelligenza artificiale, data science e altre discipline chiave del settore.

Questo **Master in Informatica e Linguaggi di Programmazione** possiede il programma universitario più completo e aggiornato del mercato. Le sue caratteristiche principali sono:

- ♦ Sviluppo di casi di studio presentati da esperti in Informatica e Linguaggi di Programmazione
- ♦ Contenuti grafici, schematici ed eminentemente pratici che forniscono informazioni scientifiche e pratiche sulle discipline essenziali per l'esercizio della professione
- ♦ Esercizi pratici che offrono un processo di autovalutazione per migliorare l'apprendimento
- ♦ Particolare enfasi sulle metodologie innovative
- ♦ Lezioni teoriche, domande all'esperto, forum di discussione su questioni controverse e compiti di riflessione individuale
- ♦ Contenuti disponibili da qualsiasi dispositivo fisso o mobile dotato di connessione a internet



Un prestigioso Direttore Ospite Internazionale offrirà 10 esclusive Master class sulle ultime tendenze in Informatica e Linguaggi di Programmazione"

“

Approfondirai la teoria degli automi e dei linguaggi formali per capire come funzionano internamente i sistemi computazionali”

Il personale docente del programma comprende rinomati specialisti del settore e altre aree correlate, che forniscono agli studenti le competenze necessarie a intraprendere un percorso di studio eccellente.

I contenuti multimediali, sviluppati in base alle ultime tecnologie educative, forniranno al professionista un apprendimento coinvolgente e localizzato, ovvero inserito in un contesto reale.

La creazione di questo programma è incentrata sull'Apprendimento Basato su Problemi, mediante il quale lo specialista deve cercare di risolvere le diverse situazioni che gli si presentano durante il corso accademico. Lo studente potrà usufruire di un innovativo sistema di video interattivi creati da esperti di rinomata fama.

Gestirai strutture dati avanzate per sviluppare soluzioni efficienti, scalabili e orientate alle prestazioni.

Progetterai algoritmi sofisticati per risolvere problemi computazionali ad alta domanda in settori tecnologici chiave.



02

Perché studiare in TECH?

TECH è la più grande università digitale del mondo. Con un catalogo eccezionale di oltre 14.000 programmi accademici disponibili in 11 lingue, si posiziona come leader in termini di occupabilità, con un tasso di inserimento professionale del 99%. Inoltre, dispone di un enorme personale docente, composto da oltre 6.000 professori di altissimo prestigio internazionale.



“

Studia presso la più grande università digitale del mondo e assicurati il successo professionale. Il futuro inizia con TECH"

La migliore università online al mondo secondo FORBES

La prestigiosa rivista Forbes, specializzata in affari e finanza, ha definito TECH "la migliore università online del mondo". Lo hanno recentemente affermato in un articolo della loro edizione digitale, che riporta il caso di successo di questa istituzione: "grazie all'offerta accademica che offre, alla selezione del suo personale docente e a un metodo innovativo di apprendimento orientato alla formazione dei professionisti del futuro".

Forbes

La migliore università online del mondo

Il piano

di studi più completo

I piani di studio più completi del panorama universitario

TECH offre i piani di studio più completi del panorama universitario, con argomenti che coprono concetti fondamentali e, allo stesso tempo, i principali progressi scientifici nelle loro specifiche aree scientifiche. Inoltre, questi programmi sono continuamente aggiornati per garantire agli studenti l'avanguardia accademica e le competenze professionali più richieste. In questo modo, i titoli universitari forniscono agli studenti un vantaggio significativo per elevare le loro carriere verso il successo.

Il miglior personale docente internazionale top

Il personale docente di TECH è composto da oltre 6.000 docenti di massimo prestigio internazionale. Professori, ricercatori e dirigenti di multinazionali, tra cui Isaiah Covington, allenatore dei Boston Celtics; Magda Romanska, ricercatrice principale presso MetaLAB ad Harvard; Ignacio Wistuba, presidente del dipartimento di patologia molecolare traslazionale di MD Anderson Cancer Center; o D.W Pine, direttore creativo della rivista TIME, ecc.

Personale docente Internazionale
TOP

La più grande università digitale del mondo

TECH è la più grande università digitale del mondo. Siamo la più grande istituzione educativa, con il migliore e più ampio catalogo educativo digitale, cento per cento online e che copre la maggior parte delle aree di conoscenza. Offriamo il maggior numero di titoli di studio, diplomi e corsi post-laurea nel mondo. In totale, più di 14.000 corsi universitari, in undici lingue diverse, che ci rendono la più grande istituzione educativa del mondo.

**N°1
al Mondo**

La più grande università online del mondo



La metodologia più efficace

Un metodo di apprendimento unico

TECH è la prima università ad utilizzare il *Relearning* in tutte le sue qualifiche. Si tratta della migliore metodologia di apprendimento online, accreditata con certificazioni internazionali di qualità docente, disposte da agenzie educative prestigiose. Inoltre, questo modello accademico dirompente è integrato con il "Metodo Casistico", configurando così una strategia di insegnamento online unica. Vengono inoltre implementate risorse didattiche innovative tra cui video dettagliati, infografiche e riassunti interattivi.

L'università online ufficiale dell'NBA

TECH è l'università online ufficiale dell'NBA. Grazie ad un accordo con la più grande lega di basket, offre ai suoi studenti programmi universitari esclusivi, nonché una vasta gamma di risorse educative incentrate sul business della lega e su altre aree dell'industria sportiva. Ogni programma presenta un piano di studi con un design unico e relatori ospiti eccezionali: professionisti con una distinta carriera sportiva che offriranno la loro esperienza nelle materie più rilevanti.

Leader nell'occupabilità

TECH è riuscita a diventare l'università leader nell'occupabilità. Il 99% dei suoi studenti ottiene un lavoro nel campo accademico che hanno studiato, prima di completare un anno dopo aver terminato uno qualsiasi dei programmi universitari. Una cifra simile riesce a migliorare la propria carriera professionale immediatamente. Tutto questo grazie ad una metodologia di studio che basa la sua efficacia sull'acquisizione di competenze pratiche, assolutamente necessarie per lo sviluppo professionale.



Google Partner Premier

Il gigante americano della tecnologia ha conferito a TECH il logo Google Partner Premier. Questo premio, accessibile solo al 3% delle aziende del mondo, conferisce valore all'esperienza efficace, flessibile e adattata che questa università offre agli studenti. Il riconoscimento non solo attesta il massimo rigore, rendimento e investimento nelle infrastrutture digitali di TECH, ma fa anche di questa università una delle compagnie tecnologiche più all'avanguardia del mondo.



L'università meglio valutata dai suoi studenti

Gli studenti hanno posizionato TECH come l'università più valutata al mondo nei principali portali di opinione, evidenziando il suo punteggio più alto di 4,9 su 5, ottenuto da oltre 1.000 recensioni. Questi risultati consolidano TECH come l'istituzione universitaria di riferimento a livello internazionale, riflettendo l'eccellenza e l'impatto positivo del suo modello educativo.



03

Piano di studi

In un contesto in cui i progressi nei linguaggi di programmazione, nella teoria degli automi e nella semantica formale stanno guidando lo sviluppo di nuove tecnologie, l'approfondimento di questi fondamenti è la chiave per innovare in aree come il calcolo quantistico, l'intelligenza artificiale e la sicurezza informatica. Pertanto, questo piano di studi integra contenuti aggiornati e strategicamente strutturati che consentono di comprendere la logica interna dei sistemi informatici. In questo modo si favorisce una visione critica e avanzata del funzionamento computazionale, in sintonia con le trasformazioni digitali che stanno ridefinendo gli standard dell'informatica a livello globale.



“

*Progetterai e implementerai linguaggi
di dominio specifico, adattati a
specifiche esigenze tecniche"*

Modulo 1. Fondamenti di programmazione

- 1.1. Introduzione alla programmazione
 - 1.1.1. Struttura di base di un computer
 - 1.1.2. Software
 - 1.1.3. Linguaggio di programmazione
 - 1.1.4. Ciclo di vita un'applicazione informatica
- 1.2. Progettazione di algoritmi
 - 1.2.1. Risoluzione dei problemi
 - 1.2.2. Tecniche descrittive
 - 1.2.3. Elementi e struttura di un algoritmo
- 1.3. Elementi di un programma
 - 1.3.1. Origini e caratteristiche del linguaggio C++
 - 1.3.2. L'ambiente di sviluppo
 - 1.3.3. Concetto di programma
 - 1.3.4. Tipi di dati fondamentali
 - 1.3.5. Operatori
 - 1.3.6. Espressioni
 - 1.3.7. Frasi
 - 1.3.8. Input e output di dati
- 1.4. Dichiarazioni di controllo
 - 1.4.1. Frasi
 - 1.4.2. Direzioni
 - 1.4.3. Loop
- 1.5. Astrazione e modularità: funzioni
 - 1.5.1. Progettazione modulare
 - 1.5.2. Concetto di funzione e utilità
 - 1.5.3. Definizione di una funzione
 - 1.5.4. Flusso di esecuzione in una chiamata di funzione
 - 1.5.5. Prototipo di una funzione
 - 1.5.6. Restituzione dei risultati
 - 1.5.7. Chiamata di una funzione: parametri
 - 1.5.8. Passaggio di parametri per riferimento e per valore
 - 1.5.9. Ambito identificatore
- 1.6. Strutture dati statiche
 - 1.6.1. Array
 - 1.6.2. Matrici: Poliedri
 - 1.6.3. Ricerca e ordinamento
 - 1.6.4. Stringhe: Funzioni di I/O per le stringhe
 - 1.6.5. Strutture: Unioni
 - 1.6.6. Nuovi tipi di dati
- 1.7. Strutture dati dinamiche: puntatori
 - 1.7.1. Concetto e definizione di puntatore
 - 1.7.2. Operatori e operazioni con i puntatori
 - 1.7.3. Array di puntatori
 - 1.7.4. Puntatori e array
 - 1.7.5. Puntatori a stringhe
 - 1.7.6. Puntatori a strutture
 - 1.7.7. Indirizzi multipli
 - 1.7.8. Puntatori a funzioni
 - 1.7.9. Passaggio di funzioni, strutture e array come parametri di funzione
- 1.8. File
 - 1.8.1. Concetti di base
 - 1.8.2. Operazioni con i file
 - 1.8.3. Tipi di file
 - 1.8.4. Organizzazione dei file
 - 1.8.5. Introduzione ai file C++
 - 1.8.6. Gestione dei file
- 1.9. Risorse
 - 1.9.1. Definizione di risorse
 - 1.9.2. Tipi di risorse
 - 1.9.3. Vantaggi e svantaggi
 - 1.9.4. Considerazioni
 - 1.9.5. Conversione ricorsiva-iterativa
 - 1.9.6. Lo stack di ricorsione

- 1.10. Test e documentazione
 - 1.10.1. Test dei programmi
 - 1.10.2. Test della scatola bianca
 - 1.10.3. Test della scatola nera
 - 1.10.4. Strumenti per i test
 - 1.10.5. Documentazione dei programmi

Modulo 2. Struttura dei dati

- 2.1. Introduzione alla programmazione in C++
 - 2.1.1. Classi, costruttori, metodi e attributi
 - 2.1.2. Variabili
 - 2.1.3. Espressioni condizionali e loop
 - 2.1.4. Obiettivi
- 2.2. Tipi di dati astratti (ADT)
 - 2.2.1. Tipi di dati
 - 2.2.2. Strutture di base e ADT
 - 2.2.3. Vettori e *array*
- 2.3. Strutture di dati lineari
 - 2.3.1. ADT lista: definizione
 - 2.3.2. Elenchi collegati e doppiamente collegati
 - 2.3.3. Elenchi ordinati
 - 2.3.4. Elenchi in C++
 - 2.3.5. ADT pila
 - 2.3.6. ADT coda
 - 2.3.7. Pila e coda in C++
- 2.4. Strutture di dati gerarchiche
 - 2.4.1. ADT albero
 - 2.4.2. Percorsi
 - 2.4.3. Alberi n-ari
 - 2.4.4. Alberi binari
 - 2.4.5. Alberi binari di ricerca

- 2.5. Strutture dati gerarchiche: alberi complessi
 - 2.5.1. Alberi perfettamente bilanciati o di altezza minima
 - 2.5.2. Alberi multipercorso
 - 2.5.3. Riferimenti bibliografici
- 2.6. Insieme e coda di priorità
 - 2.6.1. ADT heap
 - 2.6.2. ADT coda di priorità
- 2.7. Tabelle *hash*
 - 2.7.1. ADT tabella *hash*
 - 2.7.2. Funzioni *hash*
 - 2.7.3. Funzione *hash* nelle tabelle *hash*
 - 2.7.4. Ridispersione
 - 2.7.5. Tabelle *hash* aperte
- 2.8. Grafi
 - 2.8.1. Grafi ADT
 - 2.8.2. Tipi di grafi
 - 2.8.3. Rappresentazione grafica e operazioni di base
 - 2.8.4. Progettazione dei grafi
- 2.9. Algoritmi e concetti grafici avanzati
 - 2.9.1. Problemi grafici
 - 2.9.2. Algoritmi di percorso
 - 2.9.3. Algoritmi di percorso o di ricerca
 - 2.9.4. Altri algoritmi
- 2.10. Altre strutture di dati
 - 2.10.1. Insiemi
 - 2.10.2. *Array* paralleli
 - 2.10.3. Tabelle dei simboli
 - 2.10.4. *Tries*

Modulo 3. Algoritmo e complessità

- 3.1. Introduzione alle strategie di progettazione di algoritmi
 - 3.1.1. Risorse
 - 3.1.2. Dividi e conquista
 - 3.1.3. Altre strategie
- 3.2. Efficienza e analisi degli algoritmi
 - 3.2.1. Misure di efficienza
 - 3.2.2. Misurare l'ingresso di input
 - 3.2.3. Misurare il tempo di esecuzione
 - 3.2.4. Caso peggiore, migliore e medio
 - 3.2.5. Notazione asintotica
 - 3.2.6. Criteri di analisi matematica per algoritmi non ricorsivi
 - 3.2.7. Analisi matematica per algoritmi ricorsivi
 - 3.2.8. Analisi empirica degli algoritmi
- 3.3. Algoritmi di ordinamento
 - 3.3.1. Concetto di ordinamento
 - 3.3.2. Ordinamento delle bolle
 - 3.3.3. Ordinamento per selezione
 - 3.3.4. Ordinamento per inserimento
 - 3.3.5. Ordinamento per mix (*Merge Sort*)
 - 3.3.6. Ordinamento rapido (*QuickSort*)
- 3.4. Algoritmi con alberi
 - 3.4.1. Concetto di albero
 - 3.4.2. Alberi binari
 - 3.4.3. Percorsi degli alberi
 - 3.4.4. Rappresentare le espressioni
 - 3.4.5. Alberi binari ordinati
 - 3.4.6. Alberi binari bilanciati
- 3.5. Algoritmi con *heaps*
 - 3.5.1. Gli *heaps*
 - 3.5.2. L'algoritmo *heap sort*
 - 3.5.3. Code prioritarie

- 3.6. Algoritmi con grafi
 - 3.6.1. Rappresentazione
 - 3.6.2. Percorso in larghezza
 - 3.6.3. Percorso in profondità
 - 3.6.4. Ordinamento topologico
- 3.7. Algoritmi *greedy*
 - 3.7.1. La strategia *greedy*
 - 3.7.2. Elementi della strategia *greedy*
 - 3.7.3. Cambio valuta
 - 3.7.4. Il problema del viaggiatore
 - 3.7.5. Problema dello zaino
- 3.8. Ricerca del percorso minimo
 - 3.8.1. Il problema del percorso minimo
 - 3.8.2. Archi e cicli negativi
 - 3.8.3. Algoritmo di Dijkstra
- 3.9. Algoritmi *greedy* sui grafi
 - 3.9.1. L'albero a sovrapposizione minima
 - 3.9.2. Algoritmo di Prim
 - 3.9.3. Algoritmo di Kruskal
 - 3.9.4. Analisi della complessità
- 3.10. *Backtracking*
 - 3.10.1. Il *backtracking*
 - 3.10.2. Tecniche alternative

Modulo 4. Progettazione avanzata degli algoritmi

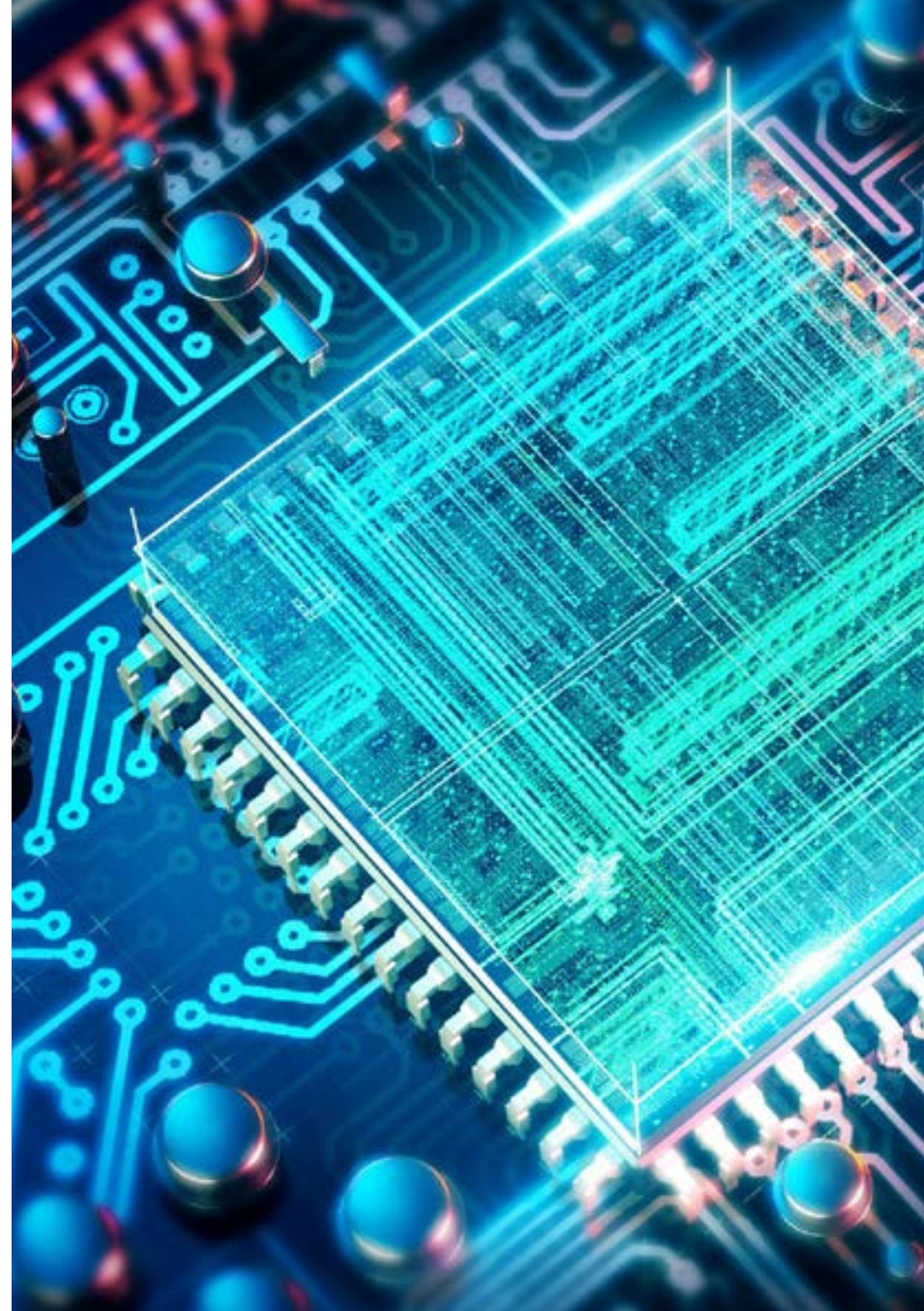
- 4.1. Analisi di algoritmi ricorsivi e divide et impera
 - 4.1.1. Approccio e risoluzione di equazioni di ricorrenza omogenee e non omogenee
 - 4.1.2. Panoramica della strategia divide et impera
- 4.2. Analisi ammortizzata
 - 4.2.1. Analisi aggregata
 - 4.2.2. Il metodo di contabilizzazione
 - 4.2.3. Il metodo del potenziale
- 4.3. Programmazione dinamica e algoritmi per problemi NP

- 4.3.1. Caratteristiche della programmazione dinamica
- 4.3.2. Indietro nel tempo: *backtracking*
- 4.3.3. Ramificazione e potatura
- 4.4. Ottimizzazione combinatoria
 - 4.4.1. Rappresentazione del problema
 - 4.4.2. Ottimizzazione 1D
- 4.5. Algoritmi di randomizzazione
 - 4.5.1. Esempi di algoritmi di randomizzazione
 - 4.5.2. Il teorema di di Buffon
 - 4.5.3. Algoritmo di Monte Carlo
 - 4.5.4. Algoritmo di Las Vegas
- 4.6. Ricerca locale e di candidati
 - 4.6.1. *Gradient Ascent*
 - 4.6.2. *Hill Climbing*
 - 4.6.3. *Simulated Annealing*
 - 4.6.4. *Tabu Search*
 - 4.6.5. Ricerca di candidati
- 4.7. Verifica formale dei programmi
 - 4.7.1. Specifica delle astrazioni funzionali
 - 4.7.2. Il linguaggio della logica del primo ordine
 - 4.7.3. Sistema formale di Hoare
- 4.8. Verifica di programmi iterativi
 - 4.8.1. Regole del sistema formale di Hoare
 - 4.8.2. Concetto di iterazioni invarianti
- 4.9. Metodi numerici
 - 4.9.1. Il metodo della bisezione
 - 4.9.2. Il metodo Newton-Raphson
 - 4.9.3. Il metodo della secante
- 4.10. Algoritmi paralleli
 - 4.10.1. Operazioni binarie parallele
 - 4.10.2. Operazioni in parallelo con i grafi
 - 4.10.3. Parallelismo nel divide et impera
 - 4.10.4. Parallelismo nella programmazione dinamica

Modulo 5. Programmazione avanzata

- 5.1. Introduzione alla programmazione orientata agli oggetti
 - 5.1.1. Introduzione alla programmazione orientata agli oggetti
 - 5.1.2. Progettazione delle lezioni
 - 5.1.3. Introduzione a UML per la modellazione dei problemi
- 5.2. Relazioni tra classi
 - 5.2.1. Astrazione ed ereditarietà
 - 5.2.2. Concetti avanzati di ereditarietà
 - 5.2.3. Polimorfismo
 - 5.2.4. Composizione e aggregazione
- 5.3. Introduzione ai design pattern per i problemi orientati agli oggetti
 - 5.3.1. Cosa sono i design pattern
 - 5.3.2. *Pattern factory*
 - 5.3.3. *Pattern singleton*
 - 5.3.4. *Pattern observer*
 - 5.3.5. *Pattern composite*
- 5.4. Eccezioni
 - 5.4.1. Quali sono le eccezioni?
 - 5.4.2. Gestione e acquisizione delle eccezioni
 - 5.4.3. Avvio delle eccezioni
 - 5.4.4. Creazione di eccezioni
- 5.5. Interfaccia utente
 - 5.5.1. Introduzione a Qt
 - 5.5.2. Posizionamento
 - 5.5.3. Cosa sono gli eventi?
 - 5.5.4. Eventi: definizione e acquisizione
 - 5.5.5. Sviluppo di interfacce utente
- 5.6. Introduzione alla programmazione concorrente
 - 5.6.1. Introduzione alla programmazione concorrente
 - 5.6.2. Il concetto di processo e di thread
 - 5.6.3. Interazione tra processi o thread
 - 5.6.4. Thread in C++
 - 5.6.5. Vantaggi e svantaggi della programmazione concorrente

- 5.7. Gestione e sincronizzazione dei thread
 - 5.7.1. Ciclo di vita di un thread
 - 5.7.2. La classe *Thread*
 - 5.7.3. Pianificazione del thread
 - 5.7.4. Gruppi di thread
 - 5.7.5. Thread di tipo *daemon*
 - 5.7.6. Sincronizzazione
 - 5.7.7. Meccanismi di bloccaggio
 - 5.7.8. Meccanismi di comunicazione
 - 5.7.9. Monitor
- 5.8. Problemi comuni nella programmazione concorrente
 - 5.8.1. Il problema dei produttori-consumatori
 - 5.8.2. Il problema dei lettori e degli scrittori
 - 5.8.3. Il problema della cena dei filosofi
- 5.9. Documentazione e test del software
 - 5.9.1. Perché è importante documentare il software?
 - 5.9.2. Documentazione di progettazione
 - 5.9.3. Utilizzo di strumenti per la documentazione
- 5.10. Test di software
 - 5.10.1. Introduzione al test del software
 - 5.10.2. Tipi di test
 - 5.10.3. Test dell'unità
 - 5.10.4. Test di integrità
 - 5.10.5. Test di convalida
 - 5.10.6. Test del sistema



Modulo 6. Informatica teorica

- 6.1. Concetti matematici utilizzati
 - 6.1.1. Introduzione alla logica proposizionale
 - 6.1.2. Teoria delle relazioni
 - 6.1.3. Set numerabili e non numerabili
- 6.2. Linguaggi formali e grammatiche e introduzione alle macchine di Turing
 - 6.2.1. Linguaggi e grammatiche formali
 - 6.2.2. Problema decisionale
 - 6.2.3. La macchina di Turing
- 6.3. Estensioni per macchine di Turing, macchine di Turing vincolate e computer
 - 6.3.1. Tecniche di programmazione per macchine di Turing
 - 6.3.2. Estensioni per macchine di Turing
 - 6.3.3. Macchine di Turing vincolate
 - 6.3.4. Macchine di Turing e computer
- 6.4. Indecibilità
 - 6.4.1. Linguaggio non ricorsivo enumerabile
 - 6.4.2. Un problema indecidibile ricorsivamente enumerabile
- 6.5. Altri problemi indicibili
 - 6.5.1. Problemi indecidibili per le macchine di Turing
 - 6.5.2. Problema di post-corrispondenza (PCP)
- 6.6. Problemi intrattabili
 - 6.6.1. Le classi P e NP
 - 6.6.2. Un problema NP completo
 - 6.6.3. Problema di soddisfacibilità ristretta
 - 6.6.4. Altri problemi NP completi
- 6.7. Problemi co - NP e PSPACE
 - 6.7.1. Complemento ai linguaggi NP
 - 6.7.2. Problemi risolvibili nello spazio polinomiale
 - 6.7.3. Problemi PSPACE completi

- 6.8. Classi di linguaggi basati sulla randomizzazione
 - 6.8.1. Modello TM con casualità
 - 6.8.2. Le classi RP e ZPP
 - 6.8.3. Test di primalità
 - 6.8.4. Complessità del test di primalità
- 6.9. Altre classi e grammatiche
 - 6.9.1. Automi finiti probabilistici
 - 6.9.2. Automi cellulari
 - 6.9.3. Celle McCulloch e Pitts
 - 6.9.4. Grammatiche di Lindenmayer
- 6.10. Sistemi informatici avanzati
 - 6.10.1. Informatica di membrana: sistemi P
 - 6.10.2. Informatica a DNA
 - 6.10.3. Informatica quantistica

Modulo 7. Teoria degli automi e linguaggi formali

- 7.1. Introduzione alla teoria degli automi
 - 7.1.1. Perché studiare la teoria degli automi?
 - 7.1.2. Introduzione alle dimostrazioni formali
 - 7.1.3. Altre forme di dimostrazione
 - 7.1.4. Induzione matematica
 - 7.1.5. Alfabeti, stringhe e lingue
- 7.2. Automi finiti deterministici
 - 7.2.1. Introduzione agli automi finiti
 - 7.2.2. Automi finiti deterministici
- 7.3. Automi finiti non deterministici
 - 7.3.1. Automi finiti non deterministici
 - 7.3.2. Equivalenza tra AFD e AFN
 - 7.3.3. Automi finiti con transizioni

- 7.4. Linguaggi ed espressioni regolari (I)
 - 7.4.1. Linguaggi ed espressioni regolari
 - 7.4.2. Automi finiti ed espressioni regolari
- 7.5. Linguaggi ed espressioni regolari (II)
 - 7.5.1. Conversione di espressioni regolari in automi
 - 7.5.2. Applicazioni delle espressioni regolari
 - 7.5.3. Algebra delle espressioni regolari
- 7.6. Pumping lemma e chiusura dei linguaggi regolari
 - 7.6.1. Pumping lemma
 - 7.6.2. Proprietà di chiusura dei linguaggi regolari
- 7.7. Equivalenza e minimizzazione degli automi
 - 7.7.1. Equivalenza di AF
 - 7.7.2. Minimizzazione di AF
- 7.8. Grammatiche indipendenti dal contesto (CIG)
 - 7.8.1. Grammatiche indipendenti dal contesto
 - 7.8.2. Alberi di derivazione
 - 7.8.3. Applicazioni delle GIC
 - 7.8.4. Ambiguità nelle grammatiche e nelle lingue
- 7.9. Automi a Pila e GIC
 - 7.9.1. Definizione di automi a Pila
 - 7.9.2. Lingue accettate da un automa a Pila
 - 7.9.3. Equivalenza tra automi a Pila e automi GIC
 - 7.9.4. Automatismo deterministico a Pila
- 7.10. Forme normali, schema di pumping GIC e proprietà delle LIC
 - 7.10.1. Forme normali di GIC
 - 7.10.2. Pumping lemma
 - 7.10.3. Proprietà di chiusura dei linguaggi
 - 7.10.4. Proprietà di decisioni dei LIC

Modulo 8. Processori linguistici

- 8.1. Introduzione al processo di compilazione
 - 8.1.1. Compilazione e interpretazione
 - 8.1.2. Ambiente di esecuzione del compilatore
 - 8.1.3. Processo di analisi
 - 8.1.4. Processo di sintesi
- 8.2. Analizzatore lessicale
 - 8.2.1. Che cos'è un analizzatore lessicale?
 - 8.2.2. Implementazione dell'analizzatore lessicale
 - 8.2.3. Azioni semantiche
 - 8.2.4. Recupero degli errori
 - 8.2.5. Problemi di implementazione
- 8.3. Analisi sintattica
 - 8.3.1. Che cos'è un parser?
 - 8.3.2. Concetti preliminari
 - 8.3.3. Parser top-down
 - 8.3.4. Parser bottom-up
- 8.4. Parsing top-down e parsing bottom-up
 - 8.4.1. Parser LL (1)
 - 8.4.2. Parser LR (0)
 - 8.4.3. Esempio di parser
- 8.5. Parsing avanzato bottom-up
 - 8.5.1. Parser SLR
 - 8.5.2. Parser LR (1)
 - 8.5.3. Parser LR (k)
 - 8.5.4. Parser LALR
- 8.6. Analisi semantica (I)
 - 8.6.1. Traduzione guidata dalla sintassi
 - 8.6.2. Tabella dei simboli

- 8.7. Analisi semantica (II)
 - 8.7.1. Controllo del tipo
 - 8.7.2. Il sottosistema dei tipi
 - 8.7.3. Equivalenza dei tipi e conversioni
- 8.8. Generazione del codice e ambiente di esecuzione
 - 8.8.1. Aspetti progettuali
 - 8.8.2. Ambiente di esecuzione
 - 8.8.3. Organizzazione della memoria
 - 8.8.4. Allocazione della memoria
- 8.9. Generazione di codice intermedio
 - 8.9.1. Traduzione diretta da *sintassi*
 - 8.9.2. Rappresentazioni intermedie
 - 8.9.3. Esempi di traduzioni
- 8.10. Ottimizzazione del codice
 - 8.10.1. Allocazione dei registri
 - 8.10.2. Eliminazione delle allocazioni morte
 - 8.10.3. Esecuzione in tempo di compilazione
 - 8.10.4. Riordino delle espressioni
 - 8.10.5. Ottimizzazione del loop

Modulo 9. Computer grafica e visualizzazione

- 9.1. Teoria del colore
 - 9.1.1. Proprietà della luce
 - 9.1.2. Modelli a colori
 - 9.1.3. Lo standard CIE
 - 9.1.4. *Profiling*
- 9.2. Primitive di output
 - 9.2.1. Il controller video
 - 9.2.2. Algoritmi per il disegno di linee
 - 9.2.3. Algoritmi per il disegno di cerchi
 - 9.2.4. Algoritmi di riempimento

- 9.3. Trasformazioni 2D, sistemi di coordinate 2D e ritaglio 2D
 - 9.3.1. Trasformazioni geometriche di base
 - 9.3.2. Coordinate omogenee
 - 9.3.3. Trasformazione inversa
 - 9.3.4. Composizione delle trasformazioni
 - 9.3.5. Altre trasformazioni
 - 9.3.6. Cambio di coordinate
 - 9.3.7. Sistemi di coordinate 2D
 - 9.3.8. Cambio di coordinate
 - 9.3.9. Standardizzazione
 - 09.3.10. Algoritmi di ritaglio
- 9.4. Trasformazioni 3D
 - 9.4.1. Traslazione
 - 9.4.2. Rotazione
 - 9.4.3. Scala
 - 9.4.4. Riflessione
 - 9.4.5. Cesoia
- 9.5. Visualizzazione e modifica delle coordinate 3D
 - 9.5.1. Sistemi di coordinate 3D
 - 9.5.2. Visualizzazione
 - 9.5.3. Cambio di coordinate
 - 9.5.4. Proiezione e standardizzazione
- 9.6. Proiezione e taglio 3D
 - 9.6.1. Proiezione ortogonale
 - 9.6.2. Proiezione parallela obliqua
 - 9.6.3. Proiezione prospettica
 - 9.6.4. Algoritmi di ritaglio 3D
- 9.7. Rimozione di superfici nascoste
 - 9.7.1. *Back - face removal*
 - 9.7.2. *Z - buffer*
 - 9.7.3. Algoritmo del pittore
 - 9.7.4. Algoritmo di Warnock
 - 9.7.5. Rilevamento delle linee nascoste



- 9.8. Interpolazione e curve parametriche
 - 9.8.1. Interpolazione e approssimazione polinomiale
 - 9.8.2. Rappresentazione parametrica
 - 9.8.3. Polinomio di Lagrange
 - 9.8.4. *Spline* cubiche naturali
 - 9.8.5. Funzioni base
 - 9.8.6. Rappresentazione della matrice
- 9.9. Curve di Bézier
 - 9.9.1. Costruzione algebrica
 - 9.9.2. Forma a matrice
 - 9.9.3. Composizione
 - 9.9.4. Costruzione geometrica
 - 9.9.5. Algoritmo di disegno
- 9.10. *B - splines*
 - 9.10.1. Il problema del controllo locale
 - 9.10.2. *B - splines* cubiche uniformi
 - 9.10.3. Funzioni di base e punti di controllo
 - 9.10.4. Deriva all'origine e molteplicità
 - 9.10.5. Rappresentazione della matrice
 - 9.10.6. *B - splines* non uniformi

Modulo 10. Computazione bio-ispirata

- 10.1. Introduzione alla computazione bio-ispirata
 - 10.1.1. Introduzione alla computazione bio-ispirata
- 10.2. Algoritmi di adattamento sociale
 - 10.2.1. Computazione bio-ispirata basata su colonie di formiche
 - 10.2.2. Varianti degli algoritmi di colonia di formiche
 - 10.2.3. Elaborazione particellare basata su cloud
- 10.3. Algoritmi genetici
 - 10.3.1. Struttura generale
 - 10.3.2. Implementazioni dei principali operatori
- 10.4. Strategie spaziali di esplorazione e sfruttamento per algoritmi genetici
 - 10.4.1. Algoritmo CHC
 - 10.4.2. Problemi multimodali

- 10.5. Modelli di computazione evolutiva (I)
 - 10.5.1. Strategie evolutive
 - 10.5.2. Programmazione evolutiva
 - 10.5.3. Algoritmi basati sull'evoluzione differenziale
- 10.6. Modelli di computazione evolutiva (II)
 - 10.6.1. Modelli evolutivi basati sulla stima delle distribuzioni (EDA)
 - 10.6.2. Programmazione genetica
- 10.7. Programmazione evolutiva applicata ai problemi di apprendimento
 - 10.7.1. Apprendimento basato sulle regole
 - 10.7.2. Metodi evolutivi nei problemi di selezione delle istanze
- 10.8. Problemi multi-obiettivo
 - 10.8.1. Concetto di dominanza
 - 10.8.2. Applicazione degli algoritmi evolutivi ai problemi multi-obiettivo
- 10.9. Reti neurali (I)
 - 10.9.1. Introduzione alle reti neurali
 - 10.9.2. Esempio pratico con le reti neurali
- 10.10. Reti neurali (II)
 - 10.10.1. Casi di utilizzo delle reti neurali nella ricerca medica
 - 10.10.2. Casi di utilizzo delle reti neurali in economia
 - 10.10.3. Casi di utilizzo delle reti neurali nella visione artificiale



Svilupperai competenze per l'analisi formale e semantica dei programmi, migliorando significativamente l'efficienza del software"

04

Obiettivi didattici

Questo Master si propone di approfondire i fondamenti teorici e pratici che sostengono la progettazione, l'analisi e l'ottimizzazione dei linguaggi computazionali. Per fare questo, combina lo studio di strutture grammaticali, semantiche e modelli computazionali con strumenti applicati allo sviluppo di sistemi intelligenti. Promuove anche il pensiero astratto e la risoluzione rigorosa di problemi complessi, chiave in ambienti ad alta domanda tecnologica. Questa prospettiva globale favorisce la capacità di innovare in aree emergenti come l'elaborazione del linguaggio naturale, i linguaggi formali e l'architettura dei compilatori.



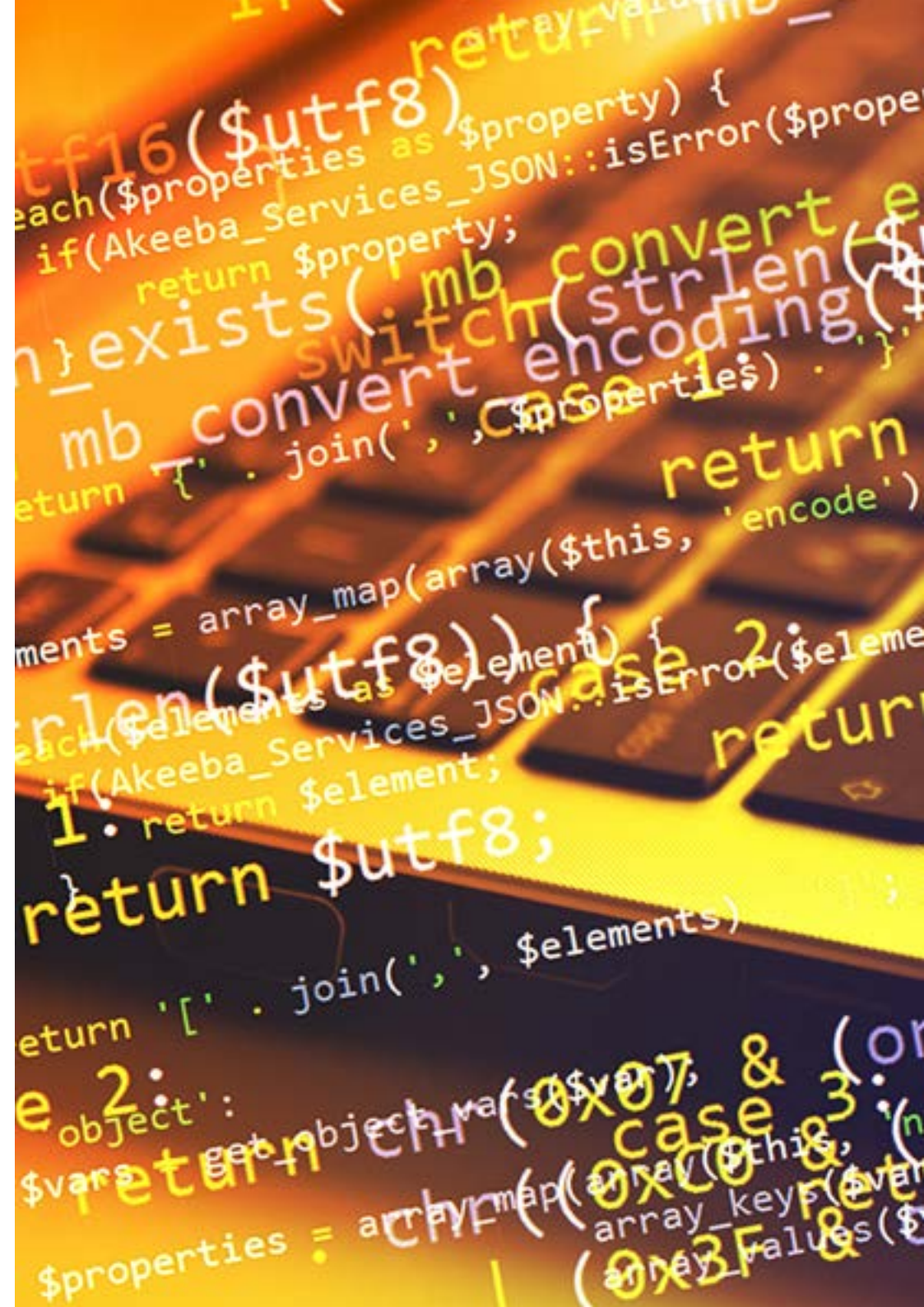
“

*Integrerai il pensiero computazionale
in progetti di innovazione tecnologica,
che ti permetteranno di offrire soluzioni
avanzate a diverse entità”*



Obiettivi generali

- Approfondire i fondamenti teorici dei linguaggi di programmazione e la loro applicazione in ambienti computazionali avanzati e specializzati
- Analizzare criticamente le strutture formali che sostengono la progettazione e l'evoluzione dei linguaggi computazionali in diversi contesti
- Applicare con solvibilità modelli di calcolo per la risoluzione efficiente di problemi complessi nel campo dell'informatica attuale
- Padroneggiare i principi essenziali della semantica formale e la loro relazione diretta con l'esecuzione precisa del software
- Studiare rigorosamente automati, grammatiche e linguaggi formali da una prospettiva computazionale e scientifica integrale
- Integrare le conoscenze fondamentali nella teoria del calcolo con gli sviluppi tecnologici attuali e futuri nel settore
- Valutare sistematicamente le prestazioni e l'efficienza di diversi linguaggi, compilatori e ambienti di sviluppo avanzati
- Progettare soluzioni computazionali innovative a partire dall'analisi dettagliata di strutture logiche, sintattiche e semantiche
- Sviluppare competenze specialistiche per la ricerca applicata nel campo dell'informatica e dei linguaggi formali
- Interpretare l'impatto dei linguaggi computazionali sulla continua evoluzione dei moderni sistemi intelligenti





Obiettivi specifici

Modulo 1. Fondamenti di programmazione

- Identificare gli elementi di base di un linguaggio di programmazione
- Applicare strutture di controllo nella risoluzione dei problemi
- Utilizzare variabili, operatori e funzioni in programmi semplici
- Sviluppare programmi strutturati seguendo le best practice

Modulo 2. Struttura dei dati

- Implementare strutture lineari come liste e pile
- Utilizzare alberi e grafi nelle rappresentazioni computazionali
- Valutare l'efficienza di diverse strutture di dati
- Selezionare strutture adeguate in base al problema posto

Modulo 3. Algoritmo e complessità

- Analizzare l'efficienza temporale e spaziale degli algoritmi
- Classificare gli algoritmi in base alla loro complessità computazionale
- Risolvere i problemi utilizzando paradigmi algoritmici di base
- Confrontare le alternative di soluzione in base alle prestazioni

Modulo 4. Progettazione avanzata degli algoritmi

- Applicare tecniche di programmazione dinamica e *greedy*
- Progettare algoritmi per problemi di ottimizzazione complessa
- Utilizzare strutture avanzate nella costruzione di algoritmi
- Valutare le prestazioni delle soluzioni algoritmiche avanzate

Modulo 5. Programmazione avanzata

- ♦ Sviluppare programmi utilizzando concetti di programmazione orientata agli oggetti
- ♦ Integrare la gestione degli errori e del debug nei progetti complessi
- ♦ Implementare modelli di progettazione in applicazioni scalabili
- ♦ Utilizzare librerie e *framework* per accelerare lo sviluppo

Modulo 6. Informatica teorica

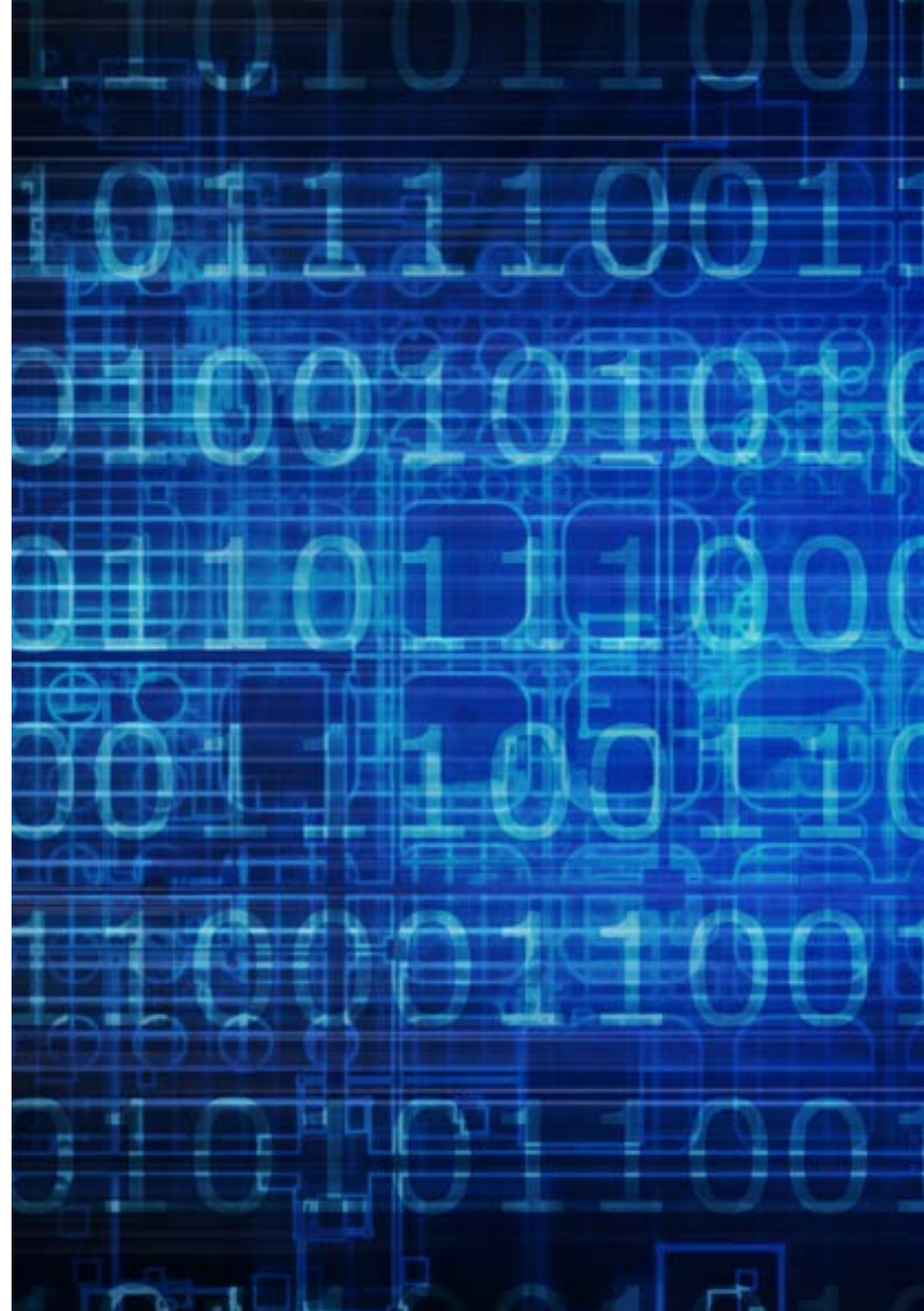
- ♦ Comprendere i limiti del calcolo attraverso modelli teorici
- ♦ Studiare i problemi non calcolabili e la loro classificazione
- ♦ Analizzare le classi P e NP in teoria del calcolo
- ♦ Applicare concetti teorici nella risoluzione di problemi astratti

Modulo 7. Teoria degli automi e linguaggi formali

- ♦ Progettare gli automi finiti deterministici e non
- ♦ Generare grammatiche regolari e senza contesto
- ♦ Analizzare i linguaggi formali utilizzando automati ed espressioni regolari
- ♦ Mettere in relazione i tipi di linguaggio con le gerarchie formali

Modulo 8. Processori linguistici

- ♦ Implementare analizzatori lessicali e sintattici di base
- ♦ Comprendere il processo di traduzione nei compilatori
- ♦ Progettare grammatiche per linguaggi di programmazione specifici
- ♦ Integrare le fasi del compilatore in un ambiente funzionale



Modulo 9. Computer grafica e visualizzazione

- ♦ Utilizzare librerie grafiche per rappresentare visivamente i dati
- ♦ Comprendere i fondamenti della modellazione grafica computazionale
- ♦ Implementare tecniche di rendering 2D e 3D
- ♦ Progettare sistemi interattivi di visualizzazione delle informazioni

Modulo 10. Computazione bio-ispirata

- ♦ Applicare algoritmi evolutivi nella risoluzione di problemi complessi
- ♦ Comprendere i fondamenti delle reti neurali e la loro formazione
- ♦ Utilizzare strategie bioispirate in contesti di intelligenza artificiale
- ♦ Confrontare i metodi bioispirati con gli algoritmi tradizionali



Lanciati in settori emergenti come elaborazione quantistica, NLP e sicurezza informatica avanzata. Renditi indispensabile!"

05

Opportunità professionali

Il costante progresso dell'intelligenza artificiale, dei linguaggi formali e dell'ingegneria del *software* ha notevolmente ampliato le opportunità di lavoro nel campo dell'informatica. Di conseguenza, avere una solida specializzazione in queste aree consente l'accesso a profili molto richiesti come designer di linguaggio, sviluppatore di compilatori, ricercatore in teoria del calcolo o esperto in analisi algoritmica. Inoltre, questo tipo di preparazione è fondamentale per l'inserimento in settori strategici come la sicurezza informatica, l'elaborazione del linguaggio naturale o il calcolo quantistico, dove si apprezza sempre più la padronanza profonda dei fondamenti computazionali.



“

*Modellerai processi computazionali
utilizzando sistemi formali e
matematici all'avanguardia"*

Profilo dello studente

Questo programma promuove lo sviluppo di un profilo tecnico e analitico, in grado di comprendere e applicare i principi alla base della moderna logica computazionale. Grazie a una visione integrata tra teoria e pratica, lo studente si caratterizza per la sua capacità di progettare soluzioni efficienti, modellare sistemi complessi e sviluppare strumenti innovativi in vari ambienti tecnologici. Inoltre, la sua padronanza dei linguaggi formali, degli algoritmi avanzati e delle strutture computazionali lo posiziona come un professionista altamente competente in ambienti accademici, scientifici o industriali in cui è richiesta una conoscenza approfondita del comportamento interno dei sistemi digitali.

*Incorporerai tecniche di calcolo bioispirate
in progetti innovativi che emulano processi
naturali ed evolutivi.*

- ♦ **Pensiero critico e risoluzione di problemi complessi:** Capacità di analizzare le situazioni da una prospettiva logica e proporre soluzioni fondate
- ♦ **Comunicazione tecnica efficace:** Capacità di esprimere concetti computazionali in modo chiaro e preciso, sia oralmente che per iscritto
- ♦ **Lavoro collaborativo in ambienti multidisciplinari:** Disponibilità ad integrarsi e contribuire a team con profili diversi nell'ambito tecnologico
- ♦ **Adattamento al cambiamento e apprendimento continuo:** Disponibilità ad aggiornare le conoscenze in un settore segnato dall'innovazione costante



Dopo aver completato il programma potrai utilizzare le tue conoscenze e competenze nei seguenti ruoli:

- 1. Ingegnere di software specializzato in compilatori:** Progetta e implementa traduttori di linguaggi di programmazione, ottimizzandone le prestazioni e la funzionalità.
- 2. Architetto dei linguaggi di programmazione:** Crea nuovi linguaggi o adatta quelli esistenti in base alle esigenze specifiche dei settori tecnologici.
- 3. Ricercatore in informatica teorica:** Studia modelli di calcolo, complessità algoritmica e linguaggi formali per risolvere problemi astratti applicabili ai sistemi reali.
- 4. Sviluppatore di sistemi intelligenti:** Costruisci soluzioni basate sulla logica computazionale e l'elaborazione del linguaggio naturale per applicazioni avanzate.
- 5. Analista di algoritmi:** Valuta e ottimizza gli algoritmi per migliorarne l'efficienza in contesti ad alte prestazioni computazionali.
- 6. Specialista nell'elaborazione del linguaggio naturale:** Progetta sistemi in grado di interpretare, generare o tradurre il linguaggio umano in ambienti digitali.
- 7. Ingegnere informatico grafico:** Sviluppa soluzioni visive interattive, simulazioni e ambienti grafici di alto livello per diversi settori.
- 8. Consulente in strutture computazionali:** Assiste nella selezione e implementazione di modelli e strutture per la progettazione software efficiente.

“

Implementerai soluzioni grafiche interattive e visualizzazioni utili nello sviluppo di software”

06

Licenze software incluse

TECH è un riferimento nel mondo universitario per la combinazione di tecnologie all'avanguardia con metodologie didattiche per il potenziale processo di insegnamento-apprendimento. A tal fine, ha creato una rete di partnership che le permette di accedere agli strumenti software più avanzati del mondo professionale.



“

Al momento dell'immatricolazione riceverai, in modo completamente gratuito, le credenziali per l'uso accademico delle seguenti applicazioni software professionali"

TECH ha stabilito una rete di partnership professionali in cui si trovano i principali fornitori di software applicato alle diverse aree professionali. Queste partnership permettono a TECH di avere accesso all'uso di centinaia di applicazioni e licenze software per metterle a disposizione dei suoi studenti.

Le licenze di software accademico consentiranno agli studenti di utilizzare le applicazioni informatiche più avanzate nel loro campo professionale, in modo da poterle conoscere e padroneggiarle senza dover sostenere costi aggiuntivi. TECH si occuperà della procedura di assunzione in modo che gli studenti possano utilizzarle illimitatamente durante il tempo in cui stanno studiando il programma Master in Sviluppo di Software, e inoltre potranno farlo completamente gratuitamente.

TECH ti darà accesso gratuito all'uso delle seguenti applicazioni software:



Google Career Launchpad

Google Career Launchpad è una soluzione per sviluppare competenze digitali in tecnologia e analisi dei dati. Con un valore stimato di **5.000\$**, è incluso **gratuitamente** nel programma universitario di TECH, fornendo accesso a laboratori interattivi e certificazioni riconosciute nel settore.

Questa piattaforma combina la formazione tecnica con casi pratici, utilizzando tecnologie come BigQuery e Google AI. Offre ambienti simulati per sperimentare con dati reali, insieme a una rete di esperti per l'orientamento personalizzato.

Funzioni in evidenza:

- ♦ **Corsi specializzati:** contenuti aggiornati su cloud computing, machine learning e analisi dei dati
- ♦ **Live lab:** esercizi pratici con gli strumenti reali di Google Cloud senza ulteriore configurazione
- ♦ **Certificazioni integrate:** preparazione per esami ufficiali con validità internazionale
- ♦ **Mentoring professionale:** sessioni con esperti di Google e partner tecnologici
- ♦ **Progetti collaborativi:** sfide basate su problemi reali di aziende leader

In conclusione, **Google Career Launchpad** collega gli utenti con le ultime tecnologie sul mercato, facilitando il loro inserimento in aree come intelligenza artificiale e data science con credenziali supportate dall'industria.



DBeaver Enterprise Edition

DBeaver Enterprise Edition è la versione professionale del rinomato gestore di database DBeaver, con un prezzo commerciale di circa **250 euro** all'anno. Il programma universitario di TECH è offerto **gratuitamente**, consentendo agli studenti di gestire, sviluppare e analizzare i dati in ambienti complessi in modo professionale e sicuro.

Questa piattaforma consente agli studenti di TECH di ottimizzare la gestione dei database relazionali e non relazionali, generare query SQL intelligenti, progettare schemi avanzati e visualizzare le informazioni con grafici interattivi. Inoltre, integra funzioni di analisi aziendale collegandosi a strumenti di *Business Intelligence*, trasformando i dati in informazioni strategiche per le decisioni.

Funzioni in evidenza:

- ♦ **Ampia compatibilità:** supporta Oracle, SQL Server, PostgreSQL, MongoDB, Cassandra e altro
- ♦ **Editor SQL avanzato:** completamento automatico, debug e assistente intelligente
- ♦ **Visualizzazione dei dati:** pannelli interattivi e grafici integrati
- ♦ **Integrazione con Tableau:** collegamento diretto agli strumenti di *Business Intelligence*
- ♦ **Progettazione di schemi:** editing ERD e reverse engineering
- ♦ **Amministrazione completa:** *backup*, ripristino, confronto e gestione degli utenti

In conclusione, **DBeaver Enterprise Edition** aiuta gli studenti di TECH a padroneggiare la gestione dei dati con precisione, efficienza e innovazione.

07

Metodologia di studio

TECH è la prima università al mondo che combina la metodologia dei **case studies** con il **Relearning**, un sistema di apprendimento 100% online basato sulla ripetizione diretta.

Questa strategia dirompente è stata concepita per offrire ai professionisti l'opportunità di aggiornare le conoscenze e sviluppare competenze in modo intensivo e rigoroso. Un modello di apprendimento che pone lo studente al centro del processo accademico e gli conferisce tutto il protagonismo, adattandosi alle sue esigenze e lasciando da parte le metodologie più convenzionali.



“

TECH ti prepara ad affrontare nuove sfide in ambienti incerti e a raggiungere il successo nella tua carriera"

Lo studente: la priorità di tutti i programmi di TECH

Nella metodologia di studio di TECH lo studente è il protagonista assoluto. Gli strumenti pedagogici di ogni programma sono stati selezionati tenendo conto delle esigenze di tempo, disponibilità e rigore accademico che, al giorno d'oggi, non solo gli studenti richiedono ma le posizioni più competitive del mercato.

Con il modello educativo asincrono di TECH, è lo studente che sceglie il tempo da dedicare allo studio, come decide di impostare le sue routine e tutto questo dalla comodità del dispositivo elettronico di sua scelta. Lo studente non deve frequentare lezioni presenziali, che spesso non può frequentare. Le attività di apprendimento saranno svolte quando si ritenga conveniente. È lo studente a decidere quando e da dove studiare.

“

*In TECH NON ci sono lezioni presenziali
(che poi non potrai mai frequentare)”*



I piani di studio più completi a livello internazionale

TECH si caratterizza per offrire i percorsi accademici più completi del panorama universitario. Questa completezza è raggiunta attraverso la creazione di piani di studio che non solo coprono le conoscenze essenziali, ma anche le più recenti innovazioni in ogni area.

Essendo in costante aggiornamento, questi programmi consentono agli studenti di stare al passo con i cambiamenti del mercato e acquisire le competenze più apprezzate dai datori di lavoro. In questo modo, coloro che completano gli studi presso TECH ricevono una preparazione completa che fornisce loro un notevole vantaggio competitivo per avanzare nelle loro carriere.

Inoltre, potranno farlo da qualsiasi dispositivo, pc, tablet o smartphone.

“

Il modello di TECH è asincrono, quindi ti permette di studiare con il tuo pc, tablet o smartphone dove, quando e per quanto tempo vuoi"

Case studies o Metodo Casistico

Il Metodo Casistico è stato il sistema di apprendimento più usato nelle migliori facoltà del mondo. Sviluppato nel 1912 per consentire agli studenti di Giurisprudenza non solo di imparare le leggi sulla base di contenuti teorici, ma anche di esaminare situazioni complesse reali. In questo modo, potevano prendere decisioni e formulare giudizi di valore fondati su come risolverle. Nel 1924 fu stabilito come metodo di insegnamento standard ad Harvard.

Con questo modello di insegnamento, è lo studente stesso che costruisce la sua competenza professionale attraverso strategie come il *Learning by doing* o il *Design Thinking*, utilizzate da altre istituzioni rinomate come Yale o Stanford.

Questo metodo, orientato all'azione, sarà applicato lungo tutto il percorso accademico che lo studente intraprende insieme a TECH. In questo modo, affronterà molteplici situazioni reali e dovrà integrare le conoscenze, ricercare, argomentare e difendere le sue idee e decisioni. Tutto ciò con la premessa di rispondere al dubbio di come agirebbe nel posizionarsi di fronte a specifici eventi di complessità nel suo lavoro quotidiano.



Metodo Relearning

In TECH i *case studies* vengono potenziati con il miglior metodo di insegnamento 100% online: il *Relearning*.

Questo metodo rompe con le tecniche di insegnamento tradizionali per posizionare lo studente al centro dell'equazione, fornendo il miglior contenuto in diversi formati. In questo modo, riesce a ripassare e ripete i concetti chiave di ogni materia e impara ad applicarli in un ambiente reale.

In questa stessa linea, e secondo molteplici ricerche scientifiche, la ripetizione è il modo migliore per imparare. Ecco perché TECH offre da 8 a 16 ripetizioni di ogni concetto chiave in una stessa lezione, presentata in modo diverso, con l'obiettivo di garantire che la conoscenza sia completamente consolidata durante il processo di studio.

Il Relearning ti consentirà di apprendere con meno sforzo e più rendimento, coinvolgendoti maggiormente nella specializzazione, sviluppando uno spirito critico, difendendo gli argomenti e contrastando opinioni: un'equazione diretta al successo.



Un Campus Virtuale 100% online con le migliori risorse didattiche

Per applicare efficacemente la sua metodologia, TECH si concentra sul fornire agli studenti materiali didattici in diversi formati: testi, video interattivi, illustrazioni, mappe della conoscenza, ecc. Tutto ciò progettato da insegnanti qualificati che concentrano il lavoro sulla combinazione di casi reali con la risoluzione di situazioni complesse attraverso la simulazione, lo studio dei contesti applicati a ogni carriera e l'apprendimento basato sulla ripetizione, attraverso audio, presentazioni, animazioni, immagini, ecc.

Le ultime prove scientifiche nel campo delle Neuroscienze indicano l'importanza di considerare il luogo e il contesto in cui si accede ai contenuti prima di iniziare un nuovo apprendimento. Poter regolare queste variabili in modo personalizzato favorisce che le persone possano ricordare e memorizzare nell'ippocampo le conoscenze per conservarle a lungo termine. Si tratta di un modello denominato *Neurocognitive context-dependent e-learning*, che viene applicato in modo consapevole in questa qualifica universitaria.

Inoltre, anche per favorire al massimo il contatto tra mentore e studente, viene fornita una vasta gamma di possibilità di comunicazione, sia in tempo reale che differita (messaggistica interna, forum di discussione, servizio di assistenza telefonica, e-mail di contatto con segreteria tecnica, chat e videoconferenza).

Inoltre, questo completo Campus Virtuale permetterà agli studenti di TECH di organizzare i loro orari di studio in base alla loro disponibilità personale o agli impegni lavorativi. In questo modo avranno un controllo globale dei contenuti accademici e dei loro strumenti didattici, il che attiva un rapido aggiornamento professionale.



La modalità di studio online di questo programma ti permetterà di organizzare il tuo tempo e il tuo ritmo di apprendimento, adattandolo ai tuoi orari"

L'efficacia del metodo è giustificata da quattro risultati chiave:

1. Gli studenti che seguono questo metodo non solo raggiungono l'assimilazione dei concetti, ma sviluppano anche la loro capacità mentale, attraverso esercizi che valutano situazioni reali e l'applicazione delle conoscenze.
2. L'apprendimento è solidamente fondato su competenze pratiche che permettono allo studente di integrarsi meglio nel mondo reale.
3. L'assimilazione di idee e concetti è resa più facile ed efficace, grazie all'uso di situazioni nate dalla realtà.
4. La sensazione di efficienza dello sforzo investito diventa uno stimolo molto importante per gli studenti, che si traduce in un maggiore interesse per l'apprendimento e in un aumento del tempo dedicato al corso.

La metodologia universitaria più apprezzata dagli studenti

I risultati di questo innovativo modello accademico sono riscontrabili nei livelli di soddisfazione globale degli studenti di TECH.

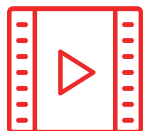
La valutazione degli studenti sulla qualità dell'insegnamento, la qualità dei materiali, la struttura del corso e i suoi obiettivi è eccellente. A questo proposito, l'università è considerata la migliore per i suoi studenti nella piattaforma di valutazione Global score, ottenendo un 4,9 su 5.

Accedi ai contenuti di studio da qualsiasi dispositivo con connessione a Internet (computer, tablet, smartphone) grazie al fatto che TECH è aggiornato sull'avanguardia tecnologica e pedagogica.

Potrai imparare dai vantaggi dell'accesso a ambienti di apprendimento simulati e dall'approccio di apprendimento per osservazione, ovvero Learning from an expert.



In questo modo, il miglior materiale didattico sarà disponibile, preparato con attenzione:



Materiale di studio

Tutti i contenuti didattici sono creati dagli specialisti che impartiranno il corso, appositamente per questo, in modo che lo sviluppo didattico sia realmente specifico e concreto.

Questi contenuti sono poi applicati al formato audiovisivo che supporterà la nostra modalità di lavoro online, impiegando le ultime tecnologie che ci permettono di offrirti una grande qualità per ogni elemento che metteremo al tuo servizio.



Capacità e competenze pratiche

I partecipanti svolgeranno attività per sviluppare competenze e abilità specifiche in ogni area tematica. Pratiche e dinamiche per acquisire e sviluppare le competenze e le abilità che uno specialista deve possedere nel mondo globalizzato in cui viviamo.



Riepiloghi interattivi

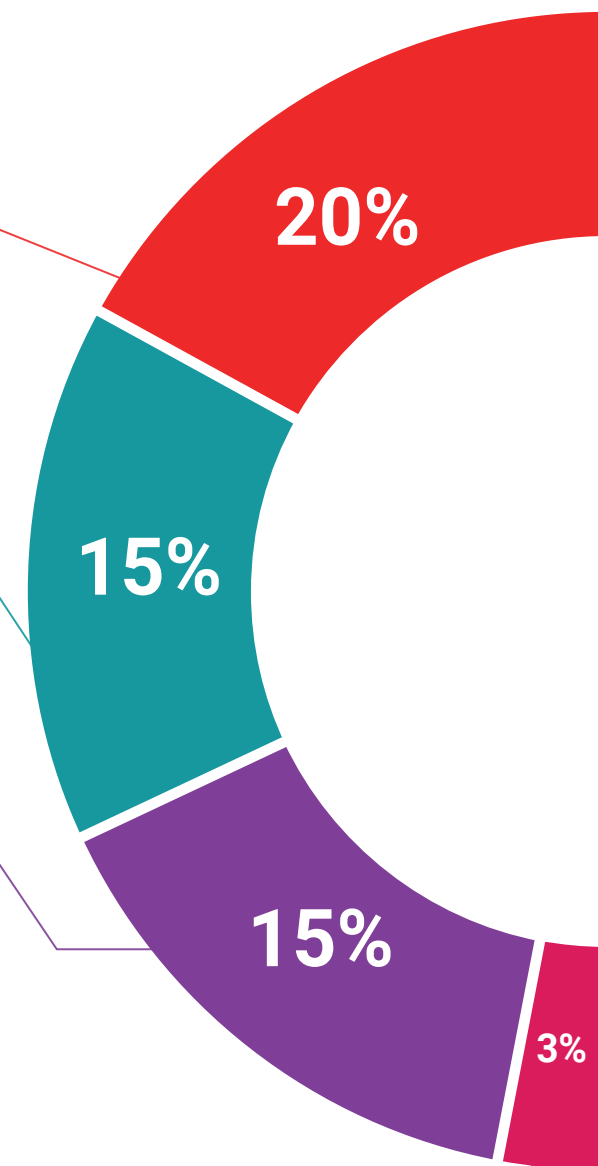
Presentiamo i contenuti in modo accattivante e dinamico tramite strumenti multimediali che includono audio, video, immagini, diagrammi e mappe concettuali per consolidare la conoscenza.

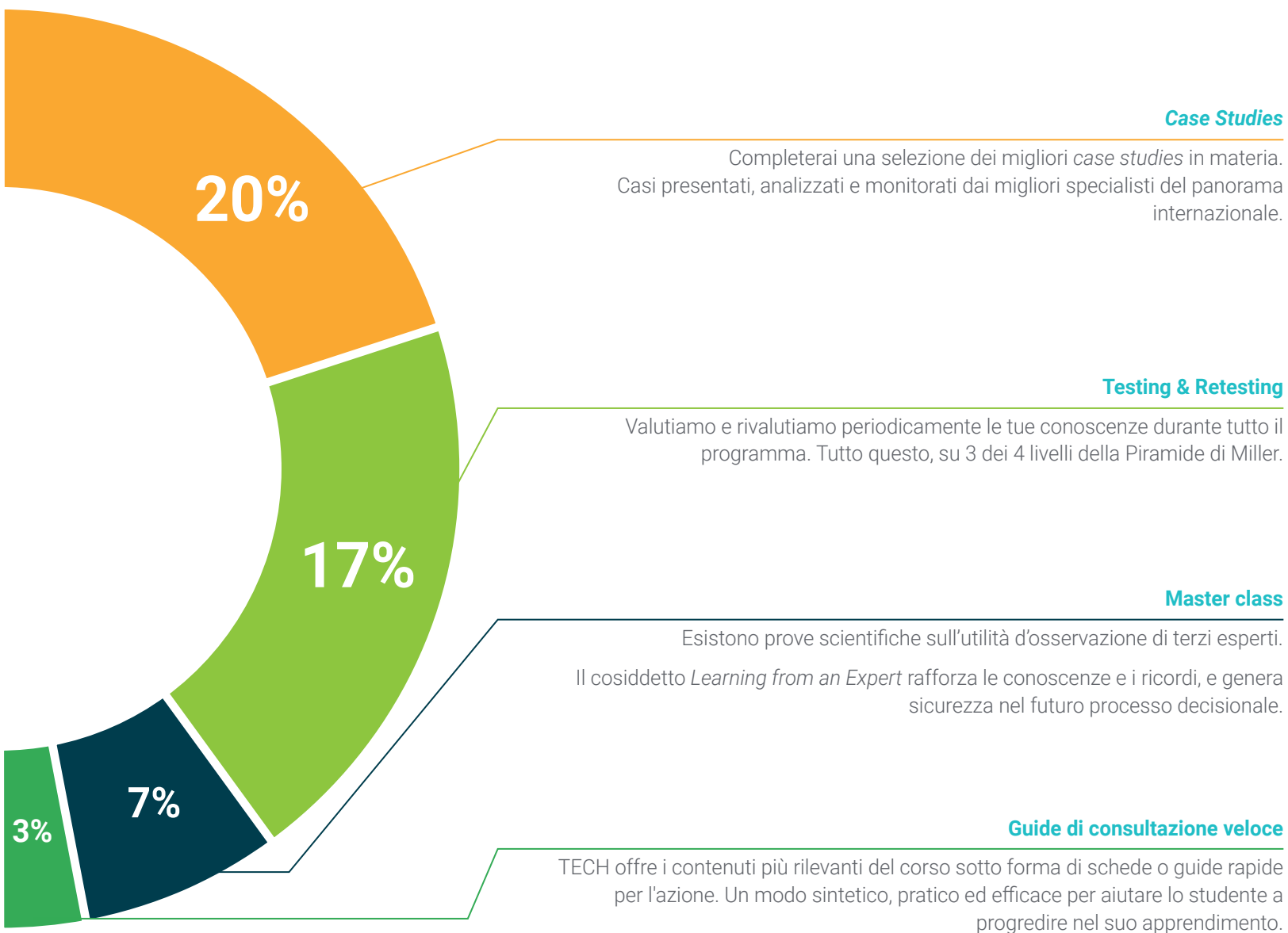
Questo esclusivo sistema di preparazione per la presentazione di contenuti multimediali è stato premiato da Microsoft come "Caso di successo in Europa".



Lecture complementari

Articoli recenti, documenti di consenso, guide internazionali... Nella nostra libreria virtuale avrai accesso a tutto ciò di cui hai bisogno per completare la tua formazione.





08

Personale docente

Il personale docente di questo programma è composto da esperti con una vasta esperienza accademica e professionale in aree chiave di informatica e linguaggi formali. Grazie alla loro esperienza nella ricerca applicata e nello sviluppo di soluzioni tecnologiche avanzate, offrono una visione aggiornata e rigorosa del settore. Inoltre, la loro partecipazione a progetti internazionali consente di trasferire in classe casi reali e metodologie innovative, che arricchiscono notevolmente il processo di apprendimento. Questa combinazione di conoscenza approfondita e approccio pratico garantisce un'esperienza accademica di alto livello, in linea con le esigenze dell'attuale ambiente scientifico e tecnologico.



“

Sarai consigliato in ogni momento dal personale docente, composto da rinomati specialisti in Informatica e Linguaggi di Programmazione”

Direttore Ospite Internazionale

Il Dott. Jeremy Gibbons è considerato un'eminenza internazionale per i suoi contributi nel campo della **Metodologia di Programmazione** e le sue applicazioni in **Ingegneria del Software**. Per più di due decenni, questo esperto associato al Dipartimento di Informatica dell'Università di Oxford ha guidato **diversi progetti di sviluppo** i cui risultati più tangibili sono applicati da informatici provenienti da diverse parti del mondo.

Il suo lavoro copre aree come la **programmazione generica**, i metodi formali, la biologia computazionale, la bioinformatica e la progettazione di algoritmi con Haskell. Quest'ultimo argomento è stato ampiamente sviluppato in collaborazione con il suo mentore, il Dottor Richard Bird.

Dal suo ruolo come **Direttore del Gruppo di Ricerca in Algebra di Programmazione**, Gibbons ha promosso progressi nei **Linguaggi di Programmazione Funzionale** e la **Teoria dei Modelli nella Programmazione**. Allo stesso tempo, le applicazioni delle sue innovazioni sono state legate al quadro sanitario, come dimostra la sua collaborazione con **CancerGrid** e **Datatype-Generic Programming**. Queste e altre iniziative riflettono il suo interesse per risolvere problemi pratici nella **ricerca sul Cancro e l'Informatica Clinica**.

Inoltre, Gibbons ha anche lasciato un'impronta significativa come **Redattore Capo di pubblicazioni accademiche** in *The Journal of Functional Programming* e *The Programming Journal: The Art, Science, and Engineering of Programming*. Attraverso queste responsabilità ha svolto un intenso lavoro di **divulgazione e diffusione della conoscenza**. Inoltre, ha guidato diversi corsi di studio collegati a rinomate istituzioni come l'Oxford Brookes University e l'Università di Auckland, in Nuova Zelanda.

Lo specialista è membro del Gruppo di Lavoro 2.1 sui Linguaggi Algoritmici e i Calcoli della **Federazione Internazionale per l'Elaborazione dell'Informazione (IFIP)**. Con questa organizzazione offre manutenzione ai linguaggi di programmazione ALGOL 60 e ALGOL 68.



Dott. Gibbons, Jeremy

- Direttore del Programma di Ingegneria del Software presso l'Università di Oxford, Regno Unito
- Vice Direttore del Laboratorio di Informatica e Dipartimento di Scienze Informatiche presso l'Università di Oxford
- Professore presso il Kellogg College, la Oxford Brookes University e l'Università di Auckland in Nuova Zelanda
- Direttore del Gruppo di Ricerca Algebra della Programmazione
- Redattore Capo delle riviste The Art, Science, and Engineering of Programming e Journal of Functional Programming
- Dottorato in Scienze Informatiche presso l'Università di Oxford
- Laurea in Informatica presso l'Università di Edimburgo
- Membro di: Gruppo di Lavoro 2.1 su Linguaggi Algoritmici e Calcoli della Federazione Internazionale per l'Elaborazione dell'Informazione (IFIP)



*Grazie a TECH potrai
apprendere dai migliori
professionisti del mondo"*

09 Titolo

Il Master in Informatica e Linguaggi di Programmazione garantisce, oltre alla preparazione più rigorosa e aggiornata, il conseguimento della qualifica di Master rilasciato da TECH Global University.



“

Porta a termine questo programma e ricevi la tua qualifica universitaria senza spostamenti o fastidiose formalità”

Questo programma permetterà di ottenere il titolo di studio privato di **Master in Informatica e Linguaggi di Programmazione** rilasciato da **TECH Global University**, la più grande Università digitale del mondo.

TECH Global University, è un'Università Ufficiale Europea riconosciuta pubblicamente dal Governo di Andorra ([bollettino ufficiale](#)). Andorra fa parte dello Spazio Europeo dell'Istruzione Superiore (EHEA) dal 2003. L'EHEA è un'iniziativa promossa dall'Unione Europea che mira a organizzare il quadro formativo internazionale e ad armonizzare i sistemi di istruzione superiore dei Paesi membri di questo spazio.

Il progetto promuove valori comuni, l'implementazione di strumenti congiunti e il rafforzamento dei meccanismi di garanzia della qualità per migliorare la collaborazione e la mobilità tra studenti, ricercatori e accademici.

Questo titolo privato di **TECH Global University**, è un programma europeo di formazione continua e aggiornamento professionale che garantisce l'acquisizione di competenze nella propria area di conoscenza, conferendo allo studente che supera il programma un elevato valore curriculare.

TECH è membro della **Association for Computing Machinery (ACM)**, la rete internazionale che raggruppa i principali referenti in informatica e scienze dell'informazione. Questo premio rafforza il suo impegno per l'eccellenza accademica, l'innovazione tecnologica e la formazione di professionisti nel settore digitale.

Partnership

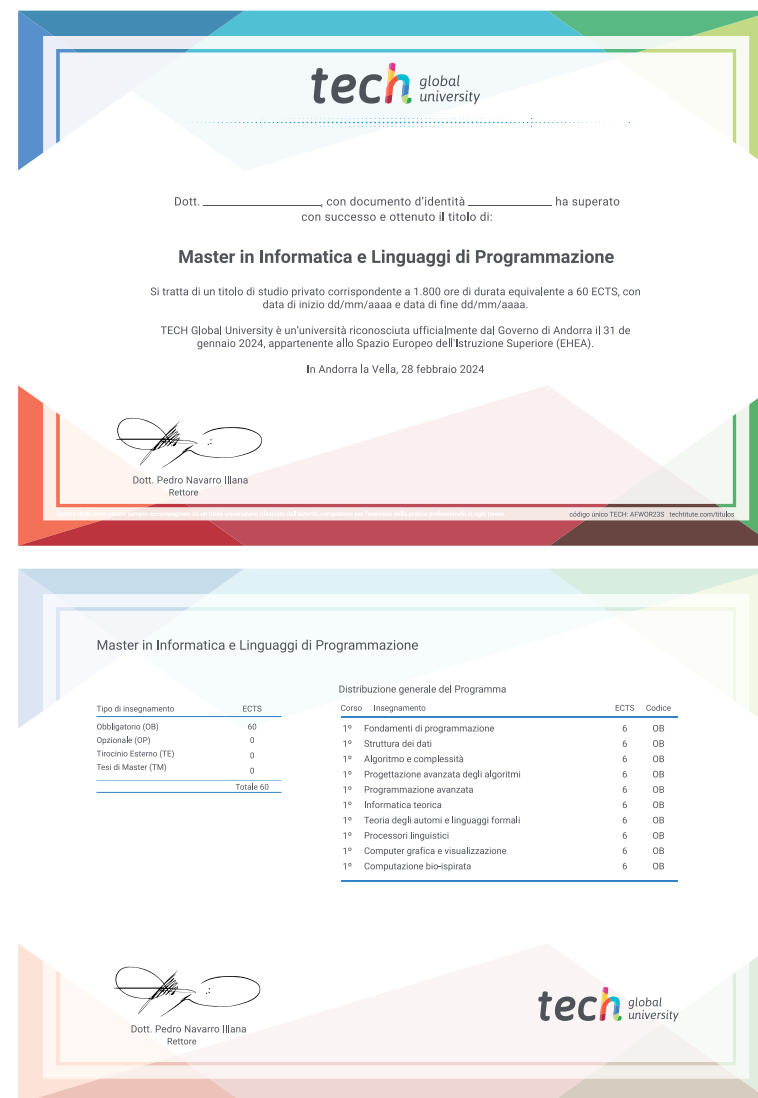


Titolo: **Master in Informatica e Linguaggi di Programmazione**

Modalità: **online**

Durata: **12 mesi**

Accreditamento: **60 ECTS**



futuro
salute fiducia persone
educazione informazione tutor
garanzia accreditamento insegnamento
istituzioni tecnologia apprendimento
comunità impegno
attenzione personalizzata inn
conoscenza presente qualità
formazione online
sviluppo istituzioni
classe virtuale lingu

tech global
university

Master
Informatica e Linguaggi
di Programmazione

- » Modalità: online
- » Durata: 12 mesi
- » Titolo: TECH Global University
- » Accreditamento: 60 ECTS
- » Orario: a tua scelta
- » Esami: online

Master

Informatica e Linguaggi di Programmazione

Partnership



Association
for Computing
Machinery



tech global
university