



UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

SYLLABUS DEL CORSO

Laboratorio di Matematica e Informatica

2526-1-E3502Q005

Obiettivi

Conoscenza e capacità di comprensione

Al termine del corso, lo studente avrà acquisito le conoscenze fondamentali relative all'architettura dei sistemi informatici e delle reti di calcolatori, nonché una comprensione critica dei principali paradigmi di programmazione, con particolare focus sul paradigma imperativo.

Capacità di applicare conoscenza e comprensione

Lo studente sarà in grado di analizzare problemi computazionali di complessità elementare, progettare algoritmi risolutivi appropriati e implementarli efficacemente utilizzando il linguaggio di programmazione Java, dimostrando padronanza delle tecniche di programmazione imperativa.

Autonomia di giudizio

Lo studente svilupperà la capacità di valutare criticamente l'appropriatezza di diverse soluzioni algoritmiche, confrontare l'efficienza di approcci alternativi e selezionare le strategie di programmazione più adeguate in relazione al contesto del problema da risolvere.

Abilità comunicative

Lo studente sarà in grado di documentare il proprio codice in modo chiaro e professionale, spiegare le scelte implementative adottate e comunicare efficacemente i risultati del proprio lavoro di programmazione utilizzando il linguaggio tecnico appropriato.

Capacità di apprendimento

Lo studente acquisirà le competenze metodologiche necessarie per approfondire autonomamente nuovi linguaggi di programmazione, tecnologie informatiche e paradigmi di sviluppo software, costruendo le basi per un apprendimento continuo nel campo dell'informatica applicata alla matematica.

Contenuti sintetici

Architettura di Von Neumann. Componenti e funzionalità di un sistema operativo. Cenni alle reti di calcolatori. Linguaggi di Programmazione. Programmazione strutturata in Java.

Programma esteso

Programma: Architettura dei sistemi informatici

- Cenni di architettura dei calcolatori e codifica dell'informazione
- Cenni di sistemi operativi
- Cenni di reti di telecomunicazioni

Programmazione strutturata in Java

- Gerarchia dei linguaggi di programmazione, compilatori e interpreti
- La Java Virtual Machine
- Algoritmi e programmi
- Tipi di dati primitivi.
- Strutture di controllo selettive e iterative
- Array di tipi primitivi
- Metodi, definizione ed invocazione
- Ricorsione

Prerequisiti

Nessuno

Modalità didattica

- Lezione frontale, 4 cfu
- Laboratorio a distanza 2 cfu

Si utilizza un approccio didattico ibrido che combina didattica frontale (DE) e didattica interattiva (DI). La DE include la presentazione e spiegazione dettagliata dei contenuti teorici. La DI prevede interventi attivi degli studenti tramite esercizi e problemi svolti durante i laboratori pratici, supportati da un tutor/esercitatore.

Le lezioni si svolgono in presenza e sono tenute in italiano, per 32 ore (4 CFU).

I Laboratori sono a distanza e sono tenuti in italiano, per 24 ore (24 CFU).

Materiale didattico

Tutte le informazioni sul corso, le slide presentate a lezione e le esercitazioni da svolgere in laboratorio verranno fornite tramite la piattaforma elearning di ateneo all'indirizzo elearning.unimib.it.

Testo di riferimento:

W. Savitch: "Programmazione di base e avanzata con Java", a cura di Daniela Micucci, 3° edizione, Pearson

Periodo di erogazione dell'insegnamento

Secondo semestre

Modalità di verifica del profitto e valutazione

Modalità dell'esame

Prova scritta e prova orale facoltativa (a richiesta dello studente che abbia ottenuto valutazione dello scritto almeno pari a 26/30). Valutazione con voto in trentesimi 18-30/30. La prova orale, che può riguardare sia domande teoriche che svolgimento di esercizi pratici di programmazione, ha peso inferiore alla prova scritta, determinando un incremento massimo di 4 punti rispetto al risultato della prova scritta ed è necessaria per l'ottenimento della lode.

La prova scritta è divisa in due parti: nella prima si valutano, attraverso una serie di domande a risposta chiusa, le conoscenze dei fondamenti teorici della programmazione strutturata; nella seconda, si valuta, attraverso l'implementazione di un semplice programma software, la capacità di realizzare in pratica un programma in grado di risolvere correttamente un semplice problema applicativo, rispondendo alle specifiche del problema da risolvere e rispettando i principi della programmazione presentati a lezione, senza generare errori (di compilazione, a runtime o logici).

La media aritmetica (eventualmente pesata) dei voti conseguiti nelle due parti definisce il voto proposto allo studente, che, se sufficiente, ha la facoltà di accettarlo o modificarlo attraverso lo svolgimento di un orale integrativo (eventualmente peggiorandolo nel caso la prova orale non sia soddisfacente). La prova orale può essere sostenuta solo in caso di voto allo scritto pari ad almeno 26/30. Il docente si riserva la facoltà di richiedere un supplemento di indagine, attraverso una prova orale obbligatoria, nei casi in cui la prova scritta, pur essendo valutata sufficiente, presenti delle criticità: ad esempio, insufficienza nella parte teorica e parte pratica molto soddisfacente, o viceversa.

Nel corso dell'anno sono previsti 6 appelli d'esame nei seguenti periodi: uno nel mese di giugno, uno a luglio, uno a settembre, uno a gennaio e due a febbraio. Durante il periodo del corso si tengono due prove scritte parziali che, in caso di esito complessivo positivo, permetteranno di verbalizzare il voto al termine del corso (6 prova d'esame).

Orario di ricevimento

Giovedì, dalle 11 alle 12 o su appuntamento

Sustainable Development Goals
