

COURSE SYLLABUS

Algorithms and Programming

2526-2-E3501Q067

Obiettivi

Scopo di questo insegnamento è rendere lo studente in grado di progettare un sistema software a oggetti, integrando diversi tipi di strategie di problem solving.

Coerentemente con gli obiettivi formativi del Corso di Studio, l'insegnamento si propone di fornire allo studente le *conoscenze di base* riguardanti la progettazione e implementazione di sistemi software secondo il paradigma della programmazione a oggetti. Verranno altresì fornite le *competenze* necessarie a modellare semplici domini attraverso il linguaggio UML e codificarli in linguaggio di programmazione Java, secondo il paradigma della programmazione a oggetti.

Al termine dell'insegnamento, lo studente sarà in grado di:

- Conoscenza e capacità di comprensione
Acquisire le conoscenze di base relative alla progettazione e implementazione di sistemi software secondo il paradigma della programmazione a oggetti.
- Conoscenza e capacità di comprensione applicate
Applicare tali conoscenze alla modellazione di semplici domini utilizzando il linguaggio UML e alla loro implementazione in linguaggio Java, seguendo i principi della programmazione a oggetti.
- Autonomia di giudizio
Integrare diverse strategie di problem solving per affrontare criticamente scelte progettuali e implementative in un contesto software orientato agli oggetti.
- Abilità comunicative
Comunicare in modo efficace le soluzioni progettuali, utilizzando correttamente il lessico tecnico della programmazione a oggetti e degli strumenti di modellazione.
- Capacità di apprendimento
Sviluppare la capacità di apprendere in autonomia ulteriori tecniche e strumenti relativi alla progettazione e sviluppo di software object oriented

Contenuti sintetici

Il corso si prefigge l'obiettivo di insegnare la programmazione ad oggetti e cenni di progettazione del software. Alla fine del corso lo studente dovrà essere in grado di modellare un problema secondo il paradigma a oggetti e di tradurlo in un programma scritto in un linguaggio object-oriented. Il linguaggio di riferimento è Java.

Programma esteso

- Introduzione ai concetti fondamentali del paradigma a oggetti (incapsulamento, ereditarietà, polimorfismo) e al linguaggio UML (Unified Modeling Language)
- Cenni al ciclo di vita del software.
- Java come linguaggio e come piattaforma.
- Il paradigma a oggetti base nel contesto Java: classi e oggetti, attributi e metodi.
- Il paradigma a oggetti avanzato nel contesto Java: ereditarietà e polimorfismo.
- Eccezioni, ArrayList, Generics e Collection Framework

Prerequisiti

Programmazione strutturata (corso di Laboratorio di Matematica e Informatica)

Modalità didattica

- Lezione frontale, 4 cfu
- Esercitazione frontale, 1 cfu
- Laboratorio frontale, 1 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 sia tramite domande e richieste durante le esercitazioni pratiche in presenza che 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).

Le esercitazioni si svolgono in presenza e sono tenute in italiano, per 12 ore (1 CFU).

I Laboratori sono a distanza e sono tenuti in italiano, per 12 ore (1 CFU).

In caso di necessità logistiche, alcune ore potrebbero essere erogate da remoto, in modalità online o offline.

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,

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 e lode. 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.

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 a oggetti; nella seconda si valuta, attraverso l'implementazione di un semplice sistema 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 a oggetti presentati a lezione, senza generare errori (di compilazione, a runtime o logici).

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 5 appelli d'esame nei seguenti periodi: uno nel mese di giugno, uno a luglio, uno a settembre, uno a gennaio e uno 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

su appuntamento.

Sustainable Development Goals
