

Scheda 03NMVMQ - Programmazione e calcolo scientifico

Versione del 05/06/2024

Presentazione

Course description

L'insegnamento si propone di approfondire le conoscenze su linguaggi e tecniche di programmazione, algoritmi e strutture dati utili per il calcolo scientifico.

L'insegnamento si propone di far acquisire agli studenti:

- una conoscenza approfondita di programmazione C++;
- una conoscenza delle problematiche connesse all'utilizzo del calcolatore per eseguire simulazioni accurate ed efficienti;
- una conoscenza dei principali algoritmi e delle fondamentali strutture di dati dell'informatica;
- una conoscenza delle problematiche legate alla geometria computazionale applicata alla computer grafica, all'integrazione numerica e alla risoluzione di equazioni differenziali.

Risultati attesi

Expected Learning Outcomes

I risultati di apprendimento attesi consistono in:

- una conoscenza approfondita di programmazione C++;
- una conoscenza delle problematiche connesse all'utilizzo del calcolatore per eseguire simulazioni accurate ed efficienti (complessità computazionale degli algoritmi, stabilità, condizionamento della formulazione del problema);
- una conoscenza dei principali algoritmi e delle fondamentali strutture di dati dell'informatica;
- una conoscenza delle problematiche principali legate alla geometria computazionale.

Le abilità che si vogliono trasmettere sono:

- scrivere un codice complesso individuale e di gruppo;
- collaborare in un team per la risoluzione di un problema computazionale complesso;
- sviluppare algoritmi valutando gli aspetti positivi e negativi delle varie soluzioni possibili;
- prendere decisioni tra le diverse possibili soluzioni implementative sulla base di valutazioni oggettive in termini di complessità computazionale, robustezza e semplicità

Le competenze ottenute alla fine dell'insegnamento sono:

- analizzare un problema complesso;
- valutare le possibili soluzioni implementative;
- progettare un codice di simulazione per il calcolo scientifico funzionante, modulare ed efficiente.

Prerequisiti

Pre-requirements

Buona conoscenza di un linguaggio di programmazione e del funzionamento del calcolatore.

Buona conoscenza degli argomenti dei corsi di base di analisi matematica, geometria e algebra lineare.

Programma

Course topics

Introduzione al Calcolo Scientifico

Ripasso su condizionamento di un problema e stabilità di un algoritmo numerico.

Cenni sulla Teoria della complessità computazionale

Introduzione all'Ingegneria del Software

Cenni sull'Esecuzione e sulle Performance di un Codice

- Introduzione al linguaggio C++
- Compiler vs Interpreter
- Preprocessore
- Puntatori, allocazione statica e dinamica della memoria
- Cenni sui Paradigmi di programmazione
- Focus sulla Programmazione orientata agli oggetti (OOP)
- OOP nel linguaggio C++

Introduzione al Version Control

- Introduzione a Git

Applicazioni del Calcolo Scientifico

- Strutture dati avanzate (liste, code, pile, alberi, grafi)
- Complessità computazionale, Ricorsione
- Algoritmi elementari (ordinamento, attraversamento di alberi e grafi, ricerca di percorsi tra nodi nei grafi)
- Strutture dati in C++
- C++ (STL)

Geometria Computazionale

- Introduzione alle classi per il trattamento di oggetti geometrici
- Intersezioni
- Matrici di Rotazione
- Trasformazioni affini, sistemi di riferimento 2D/3D
- Quadratura numerica 2D/3D
- Geometria computazionale in C++
- C++ - Eigen

Organizzazione dell'insegnamento

Course structure

L'insegnamento copre la durata di un semestre.

Si compone di lezioni teoriche, esercitazioni in aula ed esercitazioni al calcolatore.

Una parte consistente dell'insegnamento è dedicata allo sviluppo di un progetto di gruppo.

Il progetto a tema vincolato consiste nello sviluppo di un codice da parte di gruppi di 2 o 3 studenti per la risoluzione del problema proposto dal docente utilizzando gli strumenti, le strutture dati e i linguaggi presentati nell'insegnamento.

Bibliografia

Reading materials

Crescenzi Pierluigi - Gambosi Giorgio - Grossi Roberto - Rossi Gianluca,
STRUTTURE DI DATI E ALGORITMI, Progettazione, analisi e programmazione, Editore: PEARSON
EDUCATION ITALIA.

Monegato Giovanni, METODI E ALGORITMI PER IL CALCOLO NUMERICO, Editore: CLUT.

Bjarne Stroustrup, A TOUR OF C++ (Second edition), Editore: Addison-Wesley.

Materiale elettronico fornito dai docenti

Materiale di supporto allo studio

Study materials

Lucidi delle lezioni; Libro di testo; Esercizi; Esercitazioni di laboratorio; Materiale multimediale ;

Criteri, regole e procedure per l'esame esclusivamente IN PRESENZA

Assessment and grading criteria for ONSITE exam

Prova orale obbligatoria; Elaborato progettuale individuale; Elaborato progettuale in gruppo;

La durata dell'esame orale è di circa 30-45 minuti e consta di 2/3 domande con peso nella valutazione finale circa uguale.

Le domande saranno prevalentemente indirizzate alla discussione con il docente:
delle esercitazioni proposte durante l'anno e implementate indipendentemente dallo studente;
del progetto a tema vincolato scelto, facendo particolare riferimento agli aspetti di complessità
computazionale delle scelte fatte nell'implementazione.

Durante l'esame orale si approfondiranno inoltre gli argomenti proposti nell'insegnamento.

Le valutazioni sono espresse in trentesimi e l'esame è superato se la votazione riportata è di almeno 18/30. La valutazione massima è 30 e lode.

La consegna del progetto entro le scadenze indicate permette di raggiungere la votazione massima, altrimenti il voto massimo della prova orale è 26/30.

Il progetto è considerato valido al fine della prova orale se consegnato nella sezione Elaborati della pagina dell'insegnamento (e sulla propria pagina GitHub) 3 giorni prima dell'appello in cui si vuole sostenere l'esame e non oltre la fine della sessione immediatamente successiva alla conclusione delle lezioni dell'insegnamento.

L'esame orale può essere sostenuto da ciascuno studente anche se i colleghi del gruppo per il progetto non lo sostengono in quell'appello. In ogni caso sarebbe auspicabile che tutti i componenti del gruppo di un progetto sostengano l'esame insieme.

Durante il semestre verranno presentate 6 esercitazioni da svolgere a casa. Si richiede il caricamento delle soluzioni delle esercitazioni su GitHub entro due settimane dalla presentazione.

Per la prova orale è richiesto che ciascun gruppo/studente si presenti con un portatile con gli esercizi delle esercitazioni e il progetto correttamente compilabile e funzionante.

Per il progetto è prevista una presentazione di 10 minuti. E' opportuno preparare delle slide con la descrizione delle parti significative del progetto (è importante il rispetto dei tempi e la capacità di focalizzarsi sulle parti più rilevanti omettendo i dettagli).

Nella valutazione dell'esame si terrà conto in modo rilevante delle soluzioni delle esercitazioni e del loro caricamento su GitHub entro la scadenza.

Ci saranno 6 esercitazioni con consegna che inizieranno la settimana prossima (13/03) e termineranno il 24/04. Il 30 aprile ci sarà la presentazione del progetto, seguita da una lezione preparativa con introduzione al CMake e alla progettazione UML, per poi cominciare dal 08/05 con le consulenze di supporto per lo sviluppo del progetto.