



UNIVERSITÀ DEGLI STUDI DELL'AQUILA

CORSI DI INGEGNERIA

A.A. 2016/2017

Fondamenti di Informatica (I3N, I3D)

- Clementini Eliseo -

(Aggiornato il 11-09-2017)

Contenuti del corso (abstract del programma):

Il corso è finalizzato all'acquisizione dei concetti fondamentali dell'informatica, senza trascurare gli aspetti sperimentali della disciplina e la sua applicazione immediata. Si inizia con una descrizione del sistema informatico in generale per poi concentrarsi sulle tecniche di programmazione ?ad alto livello? con lo scopo di far maturare negli studenti l'abilità a progettare programmi. Il linguaggio di riferimento sarà il C++, anche se i concetti presentati nel corso sono indipendenti dal linguaggio e facilmente applicabili in qualsiasi linguaggio imperativo come Pascal o Java. Nella seconda parte del corso si affronta lo studio degli algoritmi fondamentali e dei tipi di dato astratti. Vengono esaminati i principali algoritmi di ricerca e ordinamento con il calcolo del tempo di esecuzione. I tipi astratti introdotti sono le liste, le pile, le code, e gli alberi binari per i quali si considerano varie rappresentazioni concrete e le operazioni primitive.

Programma esteso:

1. Architettura dei sistemi informatici. Processi e processori. Problemi, algoritmi e programmi. Linguaggi di programmazione. Architettura del calcolatore. Modello di von Neumann. Sistemi numerici posizionali a base fissa. Conversioni di base. Rappresentazione di numeri negativi in modulo e segno e in complemento a due. Rappresentazione di numeri reali in virgola mobile normalizzata. Codice ASCII. Algebra di Boole e principali funzioni logiche. Ambiente di programmazione Dev-C++. 2. Elementi di programmazione. Sviluppo di algoritmi. Concetto di variabile. Operazioni elementari: lettura, scrittura, assegnazione e confronto. Diagrammi di flusso. Pseudo-codice. Struttura di controllo se-allora-altrimenti. Cicli con pre-condizione e post-condizione. Strutture di controllo finché-esegui e esegui-finché. Progettazione di algoritmi con cicli. Controllo sui dati di input. Traduzione in C++. Costrutti base del C++. Struttura di un programma. Parte dichiarazioni e parte istruzioni. Dichiarazione di variabili. Classificazione dei tipi. Tipi semplici, rappresentazione interna e operazioni. Espressioni numeriche e condizionali. Istruzione di assegnazione. Istruzione composta. Selezione binaria. Funzioni standard di ingresso/uscita. Istruzioni cicliche (WHILE, DO-WHILE, FOR). Costruzione di cicli annidati. Selezione n-aria. 3. Programmazione con tipi strutturati. Tipo array: definizione e rappresentazione in memoria. Dichiarazioni di tipo con typedef. Programmi su vettori. Array bidimensionali. Programmi su matrici. Tipo string. Tipo struct. Tipo file. Primitive per la gestione sequenziale dei files. 4. Programmazione con funzioni. Sottoprogrammi. Progettazione top-down. Parametricità. Dichiarazione di funzioni. Variabili globali. Variabili locali. Parametri formali e attuali. Passaggio di parametri per riferimento e per valore. Uso della memoria. Ciclo di vita delle variabili. Visibilità

delle variabili. Mascheramento di variabili. Ricorsione. 5. Efficienza dei programmi. La funzione $t(n)$. Modello di costo. Calcolo della $t(n)$ per programmi iterativi. Caso migliore, caso peggiore, e caso medio. Complessità computazionale. Notazione asintotica. Delimitazioni alla complessità di un problema. Calcolo della $t(n)$ per programmi ricorsivi. Soluzione dell'equazione di ricorrenza per il fattoriale e per i numeri di Fibonacci. 6. Algoritmi fondamentali. Ricerca sequenziale. Ricerca binaria. Inserimento e cancellazione in un array ordinato. SelectionSort. InsertionSort. Bubblesort. Mergesort. Quicksort: analisi della complessità nel caso migliore e nel caso peggiore. 7. Programmazione con puntatori. Gestione statica e gestione dinamica della memoria. Tipo puntatore. Operazioni con puntatori: assegnazione, new, delete, riferimento, dereferenziazione. Costante null. Uso dei puntatori. Creazione dinamica di array. Similitudine tra array e puntatori. Creazione dinamica di matrici. Strutture collegate in memoria dinamica. Liste collegate con puntatori. Programmi su liste collegate. 8. Tipi di dato astratti. Liste: definizioni, rappresentazione con array, rappresentazione con lista collegata, rappresentazione con lista collegata bidirezionale, operazioni primitive sulle liste. Pile: definizioni, rappresentazione con array, rappresentazione con lista collegata, operazioni primitive sulle pile. Code: definizioni, prima rappresentazione con array, seconda rappresentazione con array (gestione circolare), rappresentazione con lista collegata, operazioni primitive sulle code. Alberi: definizioni, alberi n-ari, alberi binari, algoritmi di visita in profondità e in larghezza. Alberi binari: rappresentazione con puntatori, codifica delle operazioni primitive. Calcolo della complessità in programmi che fanno uso di tipi astratti.

Modalità d'esame:

È prevista una prova scritta sugli argomenti 2, 3, 4, 7. Seguirà una prova orale su tutto il programma

Risultati d'apprendimento previsti:

Il corso ha lo scopo di far maturare negli studenti l'abilità a progettare programmi. Il linguaggio di riferimento sarà il C++. Inoltre, gli studenti apprenderanno gli algoritmi e le strutture dati fondamentali dell'informatica.

Link al materiale didattico:

<http://www.didattica.univaq.it/moodle/course/view.php?id=4875>

Testi di riferimento:

PARTI 1-4, 7: E. Clementini, Fondamenti di informatica: Programmazione strutturata in C++. Carocci Editore, 2006.

PARTI 5-6, 8: Dispensa del docente su algoritmi e strutture dati.