Fondamenti di Informatica

Docente: Gabriele Di Stefano

Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica
Università degli Studi dell'Aquila, Italy
gabriele.distefano@univaq.it

I Anno Corso di Laurea in Ingegneria dell'Informazione

Introduzione al corso

Cosa vedremo:

- Obiettivi del corso
- Programma del corso
- Materiale didattico e strumenti di sviluppo
- Modalità di esame

Tutti i dettagli relativi al corso possono essere trovati presso il sito:
http://gs.ing.univaq.it > Fondamenti di Informatica.

Obiettivi del corso

L'obiettivo principale del corso è prendere confidenza con il "pensiero algoritmico", attraverso un processo di astrazione e modellazione di problemi a cui trovare soluzioni mediante algoritmi.
Centrale è lo studio e lo sviluppo di algoritmi e il ragionare sulle loro proprietà, tra cui correttezza, efficienza, ottimalità.

Il corso quindi si propone di presentare i concetti di base relativi all'elaborazione automatica dell'informazione. Tali concetti sono sia analizzati da un punto di vista teorico che presentati in pratica.

Scopo del corso è anche l'apprendimento di un linguaggio di programmazione per la realizzazione degli algoritmi.
Il linguaggio utilizzato è il Python.

Programma

Alcune lezioni saranno dedicate ad una introduzione al corso ed in particolare ai seguenti argomenti:

- Informatica: definizione e cenni storici
- Le nozioni di problema, algoritmo e programma.
- L'organizzazione del calcolatore.
- Linguaggi di programmazione. Processi e processori.
- La macchina a registri illimitati URM come modello di macchina universale.
- Semplici algoritmi e programmi per la URM.

Il programma è logicamente diviso in tre parti, ma gli argomenti saranno trattati in ordine diverso.

Programma: I - ARCHITETTURA DEI CALCOLATORI

Obiettivo: Fornire i concetti base sulla rappresentazione dell'informazione, l'architettura dei calcolatori ed i linguaggi di programmazione.

- Codifica binaria dell'informazione:
  - Sistemi numerici posizionali a base fissa.
  - Conversioni di base.
  - Rappresentazione di numeri negativi.
  - Operazioni aritmetiche.
  - Rappresentazione di numeri reali in virgola mobile normalizzata.
  - Algebra di Boole e principali funzioni logiche.
  - Codifica caratteri e immagini.
- Architettura dei calcolatori:
  - Modello di von Neumann.
  - Linguaggio macchina. Il linguaggio assemblatore.
  - Linguaggi di alto livello.
  - Compilatori e interpreti.

Programma: II - PROGRAMMAZIONE

Obiettivo: introdurre i costrutti fondamentali del linguaggio Python.

- Ambiente di programmazione Python: compilatore e interprete. Struttura di un programma Python. Concetto di variabile. Operazioni elementari: lettura, scrittura, assegnazione e confronto. Tipi semplici, rappresentazione interna e operazioni. Espressioni numeriche e logiche. Diagrammi di flusso. Pseudo-codice. Struttura di controllo se-allora-altrimenti. Cicli con pre-condizione. Progettazione di algoritmi con controlli e cicli. Costrutti base del Python (if-else, if-elif-else, for, while).
- Tipi strutturati: stringhe, liste, tuple, pile, code, insiemi, dizionari, matrici. Algoritmi di base su tipi strutturati (lettura, scrittura, ricerca,inserimento e cancellazione di elementi). Riferimenti a tipi strutturati. Tipo file. Primitive per la gestione sequenziale dei files.
- Funzioni: concetto di sottoprogramma. Progettazione top-down. Dichiarazione di funzioni. Passaggio di parametri per valore e per riferimento; variabili globali e locali. Visibilità e ciclo di vita delle variabili. Mascheramento di variabili. Stack e record d'attivazione. Ricorsione.

Programma: III - ALGORITMI E COMPLESSITÀ

Obiettivo: introdurre la complessità computazionale, studiare alcuni algoritmi di base e valutare la loro complessità, introdurre le strutture dati complesse e valutazione di alcuni algoritmi di base relativi.

- Complessità dei programmi: funzioni calcolabili e non calcolabili. Efficienza dei programmi e modelli di costo. Caso migliore, caso peggiore, e caso medio. Complessità computazionale. Notazione asintotica. Delimitazioni alla complessità di un problema.
- Algoritmi fondamentali: Ricerca sequenziale. Ricerca binaria. Inserimento e cancellazione in un array ordinato. Algoritmi di ordinamento: analisi della complessità nel caso migliore e nel caso peggiore. Introduzione agli alberi e loro rappresentazione. Algoritmi di visita di alberi binari.

Materiale didattico

Materiale fornito dal docente:

- Lucidi del corso
- Codice programmi in Python
- Dispensa del docente sulla macchina URM

Testi

Testo di riferimento:

- "Concetti di Informatica e Fondamenti di Python". Cay S. Horstmann, Apogeo.

Si possono consultare:

- "Informatica arte e mestiere", D. Mandrioli et al., McGrawHill per approfondimenti sulla prima parte del corso
- "Programmazione in Python", K. A. Lambert, Apogeo per argomenti avanzati di programmazione

Strumenti di sviluppo

Uno dei vantaggi dell'interprete Python è che può essere utilizzato anche con un semplice terminale.
Si suggerisce comunque di fornirsi di un IDE (Integrated Development Environment) per lo sviluppo di programmi. I due seguenti sono semplici ed adeguati per gli studenti anche alle prime armi. Disponibili per i sistemi operativi Windows, Linux e Mac OS.

- Spyder: semplice IDE a finestra unica divisa in due parti principali: una per l'interprete Python e l'altra per la scrittura dei programmi
- Idle: è l'IDE che viene generalmente fornito al momento dell'istallazione di Python. Presenta due finestre separate per l'interprete e per la scrittura dei programmi.

Modalità di esame

Durante il corso saranno organizzati due prove di esonero: se entrambe sono superate lo studente può registrare l'esito in uno degli appelli dell'anno accademico corrente.

Ogni appello consiste di

- una prova scritta e una orale.
  - La prova scritta è relativa ai concetti fondamentali e avanzati dell'informatica. Consiste nel fornire risposte a domande di carattere generale e nella realizzazione di programmi in Python.
    La realizzazione di programmi in Python in sede di esame, può essere sostituita da una tesina implementativa.
  - L'orale prevede la discussione del compito e domande di approfondimento.