

**L'etimologia, definizioni e terminologia tecnica.**

A): Etimologia. Si introducono i termini tecnici e la loro origine etimologica in itinere.

B): Le licenze. Aspetti etici e legali legati ai programmi.

---

**PARTE I: Sistema operativo MS\_Windows.**

Conoscenze generali e tecniche d'uso del SO MS-Windows

---

**PARTE II: Programmi portatili.**

Conoscenze e tecniche di utilizzo di applicazioni portatili su MS\_Windows. Differenza tra programmi portatili e non.

**PARTE III: Il sistema operativo GNU/Linux.**

Fondamenti sull'utilizzo del sistema operativo GNU/Linux, facendo uso della distribuzione LinuxMint (anche in macchina virtuale) e SLAX su macchine virtuali;

**Relazioni fondamentali tra macchine, problemi, informazioni e linguaggi.**

**PARTE I: Il calcolatore come macchina per programmare.**

Introduzione. Etimologia ed epistemologia dell'informatica. Struttura di von Neumann del calcolatore. Cenni alla struttura di Harvard. Ruolo della memoria fisica nell'uso del calcolatore. Input e output. Definizione di variabile e relazione con quella della matematica di incognita. Primi cenni a programmi e processi.

---

**PARTE II: Definizioni fondamentali.**

Introduzione. Il dato. L'informazione. Relazione con i codici e con i sistemi di numerazione. Il problema. L'analisi. L'algoritmo.

L'implementazione. Sui linguaggi: alfabeto, lessico, sintassi, semantica, pragmatica. Primi cenni a variabili e strutture dati. Relazione con la RAM.

---

## Linguaggi e macchine a vari livelli di astrazione.

### **PARTE I: Classificazione generale.**

1. Introduzione.
  2. Linguaggi di progetto degli algoritmi.
  3. Linguaggi di programmazione.
  4. Linguaggi di formattazione e di presentazione (HTML).
- 

### **PARTE II: Progetto di algoritmi.**

1. Introduzione.
  2. Algoritmi strutturati e non strutturati.
  3. Teorema di Bhom-Jacopini.
  4. Pseudo codice.
  5. Diagrammi di flusso.
  6. Struttogrammi (schemi di Nassi-Schneidermann).
  7. Cenni ai diagrammi a blocchi funzionali.
  8. Tabella di traccia di un algoritmo.
- 

### **PARTE III: Traduttori di algoritmi.**

1. Introduzione.
  2. Interpreti.
  3. Compilatori.
  4. Semicompilatori (bytecode).
  5. Ibridi.
  6. Primi cenni agli ambienti di sviluppo.
-

#### **PARTE IV: Linguaggi di programmazione.**

1. Introduzione.
  2. Linguaggio macchina e assembly.
  3. Linguaggi intermedi: C e Pascal.
  4. Linguaggi ad alto e altissimo livello: java e python.
  5. Ambienti di sviluppo specifici e non, per i linguaggi.
  6. Definizione ed esempi di:
    1. editor,
    2. IDE,
    3. RAD,
    4. framework.
- 

#### **Principali strutture dati e loro implementazione.**

Introduzione. Dal dato atomico alla struttura dati. Strutture dati in relazione al linguaggio. Strutture statiche. Strutture dinamiche. Vettori. Strutture (linguaggio C). Liste. Dizionari.

---

#### **File di testo.**

Introduzione. Richiami alla codifica dell'informazione: ASCII, UNICODE. Struttura di un file. Operazioni sui file di testo: creazione, cancellazione, aggiornamento, copia, trasferimento.

---

#### **SEZIONE III: LABORATORIO**

PARTE I: Progettare e implementare algoritmi utilizzando diverse strutture di dati.

## PROG\_SVOLTO\_3\_INFO\_2022

PARTE II: Analizzare e confrontare algoritmi diversi per la soluzione dello stesso problema.

PARTE III: Scegliere il tipo di organizzazione dei dati più adatto a gestire le informazioni in una situazione data.

### **1 Linguaggi trattati.**

Di seguito sono elencati i linguaggi trattati con semplici esempi ed esercizi: C, PHP, Python

---

Studenti:

Prof. Enrico Contini

Prof. Vincenzo Zanda