

Docenti: Salvemini Lorenzo, Tabor Luca (I.T.P.)

I Materiali di documentazione e studio sono disponibili sulla bacheca del registro elettronico,
N. 118 ore realizzate di lezione (2 ore alla settimana di lezione effettuate in laboratorio e 2 in classe).

UD#1	COMPETENZE	ABILITA'	CONOSCENZE
Architettura hardware e software di un computer	Analizzare tipologie e caratteristiche tecniche delle macchine elettriche e delle apparecchiature elettroniche con riferimento ai criteri di scelta per la loro utilizzazione e interfacciamento.	Abilità essenziali (in neretto): Individuare le parti fondamentali che compongono l'hardware di un computer: CPU, memorie dispositivi di I/O, etc..	Tipologie di Computer RAM,ROM,CACHE, memorie di massa periferiche di input e output Architettura Von Neumann La fase di Fetch-Execute cenni su linguaggio macchina e , assembly
		Riconoscere le caratteristiche funzionali di un computer. Saper calcolare il tempo di download /upload di file video, musicali utilizzando periferiche seriali	Concetti di : Trasmissione seriale/parallela Modem schede NIC Velocità e standard di trasmissione seriali
		Classificare i computer in base alle prestazioni : <ul style="list-style-type: none"> • potenza di calcolo • capacità della RAM e delle memorie di massa • frequenza clock • velocità di trasmissione periferiche 	Concetti di : Potenza di calcolo, velocità del microprocessore cache, tempo di accesso delle memorie
		Attività laboratorio: Assemblaggio di un PC	
UD#2	COMPETENZE	ABILITA'	CONOSCENZE
Algoritmi	Utilizzare linguaggi di programmazione, di diversi livelli, riferiti ad ambiti specifici di applicazione.	Abilità essenziali (in neretto) Riconoscere e analizzare una situazione problematica. Rappresentare una procedura rigorosa e generale della soluzione di un problema	Problemi proposti: soluzione di equazione di primo e secondo grado, la serie dei numeri di Fibonacci, ordinamento tra tre numeri, visualizzazione di N numeri primi , visualizzazione della somma di numeri interi, pari o dispari, fattoriale.
		Sa analizzare e spiegare la soluzione di un problema dato scritto in linguaggio di progetto Sa modificare / completare la soluzione di un problema Sa formulare la soluzione di problemi rappresentandoli in linguaggio di progetto o	Indentazione Assegnazione , concetto di variabile Sequenza di istruzioni , Selezione, se (condizione) Iterazione. Esegui....fintantoché la condizione è vera Esegui finchè non è vera fintantoché è vera finché non è vera Concetto di algoritmo: definizione, proprietà, input output,

UD#3	COMPETENZE	ABILITA'	CONOSCENZE
Linguaggi di programmazione C/C ++	Utilizzare linguaggi di programmazione, di diversi livelli, riferiti ad ambiti specifici di applicazione	Abilità essenziali (in neretto) Sa riconoscere in un programma dato : i tipi di variabili le strutture di controllo lo scopo del programma.	Programmazione in C++ Istruzioni di ingresso e di uscita. Istruzioni di assegnazione. Tipi di variabili, Istruzioni di selezione: if, if else, switch case, ecc.. Istruzioni di iterazione: cicli while, do while, for.
		Sa modificare/completare un programma dato verso la soluzione ottimale	Le funzioni: Istruzioni inerenti le funzioni. Vantaggi derivanti dall'utilizzo delle funzioni. Passaggio di parametri per valore. Array: definizione, vantaggi derivanti dal loro impiego.
		Utilizzare le strutture di controllo, i tipi di variabili opportuni e le funzioni per scrivere semplici programmi.	Realizzazione in linguaggio C++ dei programmi scritti in linguaggio di progetto nell'UD#2 Riscrittura degli stessi utilizzando le funzioni Utilizzo di replit.com e leetcode per lo svolgimento degli esercizi