



Istituto di Istruzione Superiore "Giorgi – Woolf"

Viale Palmiro Togliatti, 1161 – 00155 Roma

Curricolo verticale della disciplina INFORMATICA

Percorso di studio: **Istituto Tecnico**

Indirizzo di studio: **Informatica e Telecomunicazioni**

SECONDO BIENNIO

TERZO ANNO

MODULO	PREREQUISITI	CONOSCENZE	ABILITÀ	COMPETENZE	ATTIVITÀ DI LABORATORIO	CONNESSIONI INTERDISCIPLINARI	TEMPI
MODULO 1 Concetti fondamentali dell'informatica di base	<ul style="list-style-type: none">• Capacità logica di analizzare un problema.• Conoscenze di base del concetto di algoritmo.• Primi passi di codifica in linguaggio C.• Conoscenza sommaria dei costrutti del linguaggio C.• Conoscenza di base di un Sistema Operativo.	<ul style="list-style-type: none">• Concetti fondamentali: problema, dati, algoritmo.• Programmazione strutturata: programmazione Top-down programmazione Bottom-up.• Teorema di Bohm-Iacopini.• Le strutture fondamentali della programmazione strutturata: sequenza, alternativa (selezione), ripetizione (precondizionale, con contatore).• Descrizione algoritmo tramite pseudocodifica.• Il linguaggio del diagramma a blocchi: flow-charts. Caratteristiche generali del linguaggio e fasi di produzione di un programma (editing,	<ul style="list-style-type: none">• Imparare a comprendere un problema.• Definire i dati di Ingresso/Uscita/Lavoro.• Corretto uso delle Costanti.• Scrivere una Analisi risolutiva del problema.• Raccogliere dati in tabelle e costruirne il diagramma.• Disegnare un diagramma di Flusso.• Stendere una codifica nel linguaggio C.• Utilizzare il lessico e la terminologia tecnica di settore anche in lingua inglese.	<ul style="list-style-type: none">• Saper analizzare e risolvere un problema attraverso la scomposizione funzionale.• Saper descrivere algoritmi tramite flow-chart e pseudocodifica.• Saper usare le principali strutture dati elementari.• Utilizzare un ambiente di sviluppo integrato. Essere in grado di codificare un algoritmo mediante il linguaggio di programmazione C.• Saper gestire gli errori logici e di run/time.	<ul style="list-style-type: none">• Realizzazione di diagrammi di Flusso tramite SW applicativo.• Realizzazione di pseudocodifica tramite ambiente di sviluppo.• Codifica degli algoritmi svolti in linguaggio C.• Documentazione di un Software.• Stesura di programmi su i tipi di dati elementari, sulla sequenza, controllo e iterazione. Uso di Menù gestionale.• Utilizzo dell'ambiente di sviluppo Software DevC su Windows .• Uso del Sistema Operativo Windows/Linux (Ubuntu)	<ul style="list-style-type: none">• Struttura di un elaboratore con Sistemi e Reti.• Codifica dei dati e sistemi di numerazione con TPSIT• Gestione del codice ASCII con TPSIT.• Conoscenza dell'architettura di un calcolatore con Sistemi e Reti	Settembre Gennaio



Istituto di Istruzione Superiore "Giorgi – Woolf"

Viale Palmiro Togliatti, 1161 – 00155 Roma

		<p>compilazione, linking).</p> <ul style="list-style-type: none"> • Comandi in ambiente DEV C++ 4.9.9.2. Tipi di dato in linguaggio C: integer, byte, float, char, string, boolean. • La conversione di tipo ed il casting. • Il linguaggio C++: vocabolario e struttura. Sintassi del linguaggio. Creazione di un programma in linguaggio C++. • Costrutti fondamentali: { .. }; If...Else; Switch...; For...; While...; o.. While.. • Costrutti fondamentali del linguaggio C++. 					
<p>MODULO 2 Funzioni, parametri e strutture dati complesse</p>		<ul style="list-style-type: none"> • Concetto di funzione e di passaggio di parametri. • Le funzioni standard. • Funzioni definite dall'utente. • Variabili globali e locali. • Parametri formali ed attuali. • Passaggio dei dati per valore e per indirizzo. • Array modimensionali e loro gestione. • Algoritmi fondamentali su array: ordinamento, ricerca. • Stringhe e loro gestione. • Array bidimensionali. • Record (struct). • Array di record. • Utilizzo dei puntatori alle celle di memoria per la gestione dinamica delle memorizzazioni di liste, pile e code 	<ul style="list-style-type: none"> • Imparare a comprendere un problema complesso. Studiare le strutture di memorizzazione dei dati in modo opportuno per risolvere il problema. Raccogliere dati in tabelle e costruirne il diagramma. Disegnare un diagramma di Flusso. Stendere una codifica nel linguaggio C++. • Utilizzare il lessico e la terminologia tecnica di settore anche in lingua inglese. 	<ul style="list-style-type: none"> • Suddividere un programma in moduli. • Saper passare parametri ai moduli in ingresso ed in uscita. • Saper usare le principali strutture dati. Essere in grado di codificare un algoritmo mediante il linguaggio di programmazione C++. • Essere in grado di gestire stringhe, array, record e strutture dinamiche di memorizzazione tipo:liste, pile, code. 	<ul style="list-style-type: none"> • Stesura di programmi in linguaggio C++ relativi alle codifica di funzioni, strutture dati semplici e complesse, statiche e dinamiche. 	<ul style="list-style-type: none"> • Uso dei Puntatori alla memoria con TPSIT 	<p>Gennaio Maggio</p>



Istituto di Istruzione Superiore "Giorgi – Woolf"

Viale Palmiro Togliatti, 1161 – 00155 Roma

OBIETTIVI MINIMI DISCIPLINARI	Le conoscenze relative agli obiettivi minimi sono quelle riportate nel curriculum, ma in contesti basilari ed accettabili a livello di approfondimento/difficoltà e con competenze/abilità minime o parziali.	
APPROCCIO DIDATTICO COMUNE A TUTTI I MODULI	METODOLOGIE	STRUMENTI
	<ul style="list-style-type: none">• Didattica laboratoriale.• Lezione frontale.• Lezione dialogata.• Classe capovolta.• Apprendimento per scoperta.• Apprendimento per progetti.• Peer to peer.• Lavoro di gruppo	<ul style="list-style-type: none">• Lezione frontale• lezione dialogata• lavoro di gruppo• Lavoro individuale, interventi personalizzati.• Compiti in classe• interrogazioni alla cattedra• interrogazioni al posto• interrogazioni programmate• questionari (prove oggettive)• relazioni da svolgere a casa• Lavagna.• Uso di PC in Laboratorio.• Piattaforma di e-learning.• Appunti elaborati dal docente.• Libro di testo.• Manuali tecnici.• Verifiche orali e scritte.• Brainstorming.• Flipped classroom.• Test a risposta chiusa e aperta.• Sussidi audiovisivi.• Ambiente di sviluppo integrato (IDE).