



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 Dal problema all'algoritmo		<ul style="list-style-type: none">La simbologia dei diagrammi di flussoLa rappresentazione delle figure strutturaliLa definizione e le caratteristiche di un algoritmoLa relazione tra algoritmo e programmaIl concetto di linguaggio di progetto e di pseudo-codifica	<ul style="list-style-type: none">Descrivere procedure mediante algoritmiRappresentare gli algoritmi mediante Flow chatMemorizzare le informazioni nelle variabili	<ul style="list-style-type: none">Saper descrivere la soluzione di semplici problemi mediante algoritmiAcquisire il concetto di variabile e cella di memoriaUtilizzare i diagrammi di flusso per rappresentare gli algoritmi	<ul style="list-style-type: none">Realizzazione di diagrammi di Flusso tramite software applicativo (ad esempio Flowgorithm)	<ul style="list-style-type: none">Inglese: linguaggio tecnicoSistemi e Reti: la struttura del calcolatore	settembre - ottobre
MODULO 2 Programmare in C++	<ul style="list-style-type: none">Capacità logica di analizzare un problema.Conoscenza del concetto di algoritmo.	<ul style="list-style-type: none">Linguaggi a basso livello, ad alto livello e per il webLinguaggi compilati ed interpretatiLa struttura di un programmaDichiarazione e inizializzazione di una variabileTipi di variabiliLe costantiGestione dell'input e dell'output (cin e cout)Commenti al codiceMetodologia per la risoluzione di un problema	<ul style="list-style-type: none">Disporre l'output sullo schermoUtilizzare le variabili intere e reali nei programmiSaper scambiare il contenuto di due variabili (swap)Saper creare un programma che interagisca con l'utente	<ul style="list-style-type: none">Scrivere un programma in linguaggio di progettoEditare, testare e collaudare un programma sempliceGestire l'input dei dati e formattare l'output sullo schermoScrivere programmi con istruzioni in sequenza e in blocchiSaper progettare algoritmi semplici con tecnica Top-down	<ul style="list-style-type: none">Utilizzo di un ambiente di sviluppo Software (ad esempio DevC++ o CodeBlocks)Codifica di algoritmi di sequenza in linguaggio C++	<ul style="list-style-type: none">Inglese: linguaggio tecnicoMatematica: risoluzione di esercizi con pensiero algoritmico e relativa codificaTPSIT: i tipi di variabili e la codifica con la quale vengono salvati in memoria. L'uso del codice ASCII	ottobre



Istituto di Istruzione Superiore "Giorgi – Woolf"

Viale Palmiro Togliatti, 1161 – 00155 Roma

MODULO	PREREQUISITI	CONOSCENZE	ABILITÀ	COMPETENZE	ATTIVITÀ DI LABORATORIO	CONNESSIONI INTERDISCIPLINARI	TEMPI
MODULO 3 Algoritmi di selezione	<ul style="list-style-type: none"> Dichiarazione e inizializzazione di variabili e costanti Gestione dell'input e dell'output 	<ul style="list-style-type: none"> La selezione semplice e doppia La selezione annidata La selezione multipla Variabili booleane e proposizioni logiche I connettivi logici Priorità degli operatori 	<ul style="list-style-type: none"> Scrivere un codice con istruzioni condizionali Scrivere un codice complesso con blocchi di istruzioni annidate Combinare più selezioni con condizione logiche composte Scrivere un codice utilizzando l'istruzione di selezione multipla 	<ul style="list-style-type: none"> Riconoscere quando utilizzare selezioni annidate Codificare la selezione semplice e doppia Effettuare l'annidamento delle istruzioni Utilizzare le variabili di tipo bool Utilizzare gli operatori logici 	<ul style="list-style-type: none"> Esercizi di selezione svolti in linguaggio C++ 	<ul style="list-style-type: none"> Inglese: linguaggio tecnico Matematica: risoluzione e/o verifica di equazioni e disequazioni di primo e secondo grado in C++ 	novembre
MODULO 4 Algoritmi di iterazione	<ul style="list-style-type: none"> Sintassi base di un programma in C++ Selezione semplice, doppia, annidata, multipla Proposizioni logiche 	<ul style="list-style-type: none"> L'iterazione pre-condizionata Il ciclo a condizione iniziale: while Tabella di traccia L'iterazione post-condizionata Il ciclo a condizione finale: do while L'istruzione di iterazione definita Il ciclo a conteggio: for Generazione dei numeri casuali I cicli annidati 	<ul style="list-style-type: none"> Conoscere la differenza tra controllo di testa e di coda Conoscere la differenza tra iterazione definita e indefinita Scrivere programmi con selezioni e iterazioni Utilizzare la tabella di traccia per individuare errori nel codice 	<ul style="list-style-type: none"> Saper scegliere il tipo di iterazione adeguato alle diverse soluzioni Generare numeri casuali Codificare l'iterazione definita Codificare l'iterazione indefinita Codificare programmi con cicli annidati 	<ul style="list-style-type: none"> Esercizi di iterazione svolti in linguaggio C++ 	<ul style="list-style-type: none"> Inglese: linguaggio tecnico Matematica: scrittura in C++ dei procedimenti di problemi matematici (Fibonacci, Fattoriale, Triangolo di Tartaglia) Telecomunicazioni: scrittura in C++ di programmi per la gestione di componenti con arduino (Tinkercad). 	dicembre
MODULO 5 Le funzioni	<ul style="list-style-type: none"> Sintassi base di un programma in C++ Modulo 3 e modulo 4 	<ul style="list-style-type: none"> Definizione di funzione e vantaggi Header di una funzione: tipo di ritorno, nome della funzione e parametri La condivisione delle variabili nei sottoprogrammi Visibilità delle variabili Passaggio dei parametri per valore e per riferimento Differenza tra parametri formali e attuali Funzioni ricorsive 	<ul style="list-style-type: none"> Comprendere il meccanismo del passaggio dei parametri Comprendere le regole di visibilità Scrivere algoritmi usando le funzioni Utilizzare funzioni presenti nelle librerie di C++ Saper scrivere e usare funzioni personali Individuare un problema ricorsivo Scrivere funzione ricorsive 	<ul style="list-style-type: none"> Progettare un programma con approccio Top-Down Definire la modalità del passaggio dei parametri Distinguere i parametri formali e attuali Organizzare un programma con un menu interattivo 	<ul style="list-style-type: none"> Codifica di programmi con funzioni personali. Codifica di menù interattivi in linguaggio C++ 	<ul style="list-style-type: none"> Inglese: linguaggio tecnico Matematica: Fibonacci ricorsivo 	gennaio - marzo



Istituto di Istruzione Superiore "Giorgi – Woolf"

Viale Palmiro Togliatti, 1161 – 00155 Roma

MODULO	PREREQUISITI	CONOSCENZE	ABILITÀ	COMPETENZE	ATTIVITÀ DI LABORATORIO	CONNESSIONI INTERDISCIPLINARI	TEMPI
MODULO 6 Gli array monodimensionali e le stringhe	<ul style="list-style-type: none"> Sintassi base di un programma in C++ Modulo 3 e modulo 4 Funzioni 	<ul style="list-style-type: none"> Introduzione ai dati strutturati: array e struct Il vettore o array monodimensionale Dichiarazione e inizializzazione di un vettore Inserimento di valori nel vettore Lettura e stampa dei valori presenti nel vettore Somma e media dei valori di un vettore Ricerca di un elemento nel vettore Ricerca del massimo e del minimo Conteggio delle occorrenze in un vettore Ordinamento di un vettore: bubble sort, insertion sort e selection sort Vettori paralleli Passaggio di un vettore ad una funzione Stringhe in C++ 	<ul style="list-style-type: none"> Saper codificare gli algoritmi di ricerca e ordinamento in un vettore Scrivere, leggere e ricercare dati in un vettore Operare con stringhe e sottostringhe Saper passare un array ad una funzione 	<ul style="list-style-type: none"> Saper organizzare tipi di dati complessi Definire array monodimensionali Saper ordinare un array Saper definire e manipolare una stringa 	<ul style="list-style-type: none"> Esercizi di codifica degli algoritmi sui vettori svolti in linguaggio C++ 	<ul style="list-style-type: none"> Inglese: linguaggio tecnico 	marzo – maggio
MODULO 7 Algoritmi evoluti e strutture dati avanzate	<ul style="list-style-type: none"> Sintassi base del linguaggio C++ Concetti e sintassi di codifica spiegata nei precedenti moduli 	<ul style="list-style-type: none"> Le matrici I record (struct) Ricerca dicotomica in un vettore ordinato Algoritmo di ordinamento Quicksort I puntatori 	<ul style="list-style-type: none"> Riconoscere l'importanza dell'archiviazione dei dati Utilizzare array a due dimensioni Scrivere e leggere dati da una matrice quadrata Elaborare matrici con simmetrie Manipolare record con tipi semplici Saper gestire dati di tipo diverso in relazione tra loro Saper gestire un array di record. 	<ul style="list-style-type: none"> Definire array bidimensionali Ricercare le simmetrie in una matrice quadrata Saper scrivere un programma definendo e usando le struct Definire record e strutture complesse di dati 	<ul style="list-style-type: none"> Codifica degli algoritmi svolti in linguaggio C++ Esercizi di codifica sugli argomenti affrontati 	<ul style="list-style-type: none"> Matematica: il piano cartesiano e la retta. Risolvere sistemi lineari con Cramer in C++ 	maggio - giugno



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. Dagli obiettivi minimi si possono ulteriormente escludere i seguenti argomenti: le funzioni ricorsive e l'intero modulo 7.	
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.• Lavoro di gruppo• Lavoro individuale, interventi personalizzati.• Brainstorming.• Flipped classroom.	<ul style="list-style-type: none">• 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.• Test a risposta chiusa e aperta.• Sussidi audiovisivi.• Ambiente di sviluppo integrato (IDE).