



**MINISTERO dell'ISTRUZIONE e del MERITO**  
**Istituto Statale di Istruzione Superiore**  
**ISAAC NEWTON**  
**VARESE**



**PROGRAMMAZIONE DISCIPLINARE PER COMPETENZE**  
**ARTICOLATA PER UDA**

**ANNO SCOLASTICO 2024/25**

**INDIRIZZO ITIS INFORMATICA**

**CLASSE 2^ Periodo Didattico      SEZIONE A INF**

**DISCIPLINA INFORMATICA**

**DOCENTE GIUSEPPE LEMME e IANNONE GUIDO**

**QUADRO ORARIO (N. ore settimanali nella classe) 7 ( di cui 3 Laboratorio)**

**1. QUADRO DEGLI OBIETTIVI DI COMPETENZA**

**ASSE CULTURALE: Tecnico - Scientifico**

<u><b>Competenze disciplinari del Terzo e Quarto anno triennio Itis</b></u>	<ul style="list-style-type: none"><li>• Realizzazione di strategie risolutive di semplici problemi</li><li>• Individuazione delle parti fondamentali di un programma codificato nel linguaggio studiato</li><li>• Riconoscimento ed uso delle strutture di controllo fondamentali</li><li>• Individuazione delle variabili di i/o e dichiarazione di variabili</li><li>• Acquisizione delle nozioni di base necessarie all'attività di laboratorio (ambiente di lavoro)</li><li>• Documentazione dei lavori e sviluppo con metodologia top down</li><li>• Analisi di problemi che implicino l'uso di strutture dati fondamentali (vettori, matrici, tabelle, file)</li><li>• Codifica delle soluzioni ai problemi proposti nei linguaggi studiati</li><li>• La Object Oriented Programming</li><li>• Consapevole uso del sw introdotto</li></ul>
---	--

OBIETTIVI DI COMPETENZA		ARTICOLAZIONE DELLE COMPETENZE IN ABILITA' E CONOSCENZE		
	COMPETENZE IN ESITO	ABILITA'	CONOSCENZE DELL'ASSE	CONOSCENZE DELLA DISCIPLINA
	<ul style="list-style-type: none"> <li>Analizzare il ruolo dell'elaboratore in un generico processo di elaborazione</li> <li>Riconoscere un problema di tipo logico</li> </ul>	Conoscere i principi alla base del funzionamento di un sistema di elaborazione	Sistemi informatici	<ul style="list-style-type: none"> <li>Modello logico funzionale di una macchina a controllo di programma</li> <li>Modello esecutivo: il modello classico di Von Neumann</li> </ul>
		Saper rappresentare un'informazione in codice	I codici	<ul style="list-style-type: none"> <li>Ruolo del Sistema Operativo</li> <li>Il mondo dei problemi: classificazioni e generalizzazioni</li> </ul>
		Saper risolvere un semplice problema	Fasi risolutive di un problema	Concetto di algoritmo come soluzione di un problema parametrico
		Saper utilizzare i formalismi per la rappresentazione di algoritmi	Algoritmi e loro rappresentazione	<ul style="list-style-type: none"> <li>Metodi di rappresentazione degli algoritmi</li> <li>Tecniche di verifica di un algoritmo</li> <li>Costruzione di algoritmi relativi a problemi di varia natura e loro rappresentazione</li> </ul>
	<ul style="list-style-type: none"> <li>Riconoscere il ruolo delle variabili strutturate nelle fasi della programmazione</li> <li>Utilizzare le variabili strutturate per la risoluzione di un problema</li> </ul>	Conoscere la differenza tra variabili semplici e variabili strutturate	Organizzazione logica dei dati	La variabile semplice e i tipi di dati
		Conoscere le funzionalità delle stringhe e le funzioni per la loro manipolazione		La variabile strutturata stringa Funzioni delle librerie per la manipolazione delle stringhe
		Saper gestire un array e realizzare le elaborazioni classiche	Fondamenti di programmazione e sviluppo di semplici programmi in un linguaggio a scelta	Array monodimensionali: vettori Inizializzazione e gestione dei vettori Elaborazioni classiche degli array: <ul style="list-style-type: none"> <li>Massimo,</li> <li>minimo,</li> <li>media</li> <li>Ricerca</li> <li>Ordinamento</li> <li>Merge</li> </ul> Array bidimensionali: matrici Inizializzazione e gestione di matrici
	<ul style="list-style-type: none"> <li>Saper utilizzare le funzioni predefinite</li> <li>Definire una funzione</li> <li>Definire la modalità del passaggio dei parametri</li> </ul>	Scrivere algoritmi utilizzando le funzioni Utilizzare funzioni predefinite nei programmi Utilizzare funzioni personali Scrivere funzioni parametriche		Comprendere il meccanismo del passaggio delle variabili Comprendere la differenza tra passaggio per indirizzo e passaggio per valore



				correttamente dati strutturati. Analizzare e confrontare algoritmi diversi per la soluzione dello stesso problema		
3	56	P2  M6	<b>LE FUNZIONI</b>  MARZO-APRILE	Capacità di utilizzare le funzioni predefinite; Capacità di realizzare procedure e funzioni parametriche Capacità di saper scegliere i parametri ed individuare la modalità di passaggio	<ul style="list-style-type: none"> <li>Lezione frontale</li> <li>Lezione interattiva</li> <li>problem solving</li> </ul>	<ul style="list-style-type: none"> <li>Funzioni e librerie</li> <li>Funzioni, parametri per riferimento e per valore, Funzioni ricorsive.</li> <li>Gli algoritmi fondamentali di ricerca e ordinamento di un vettore e successivamente alle altre strutture dati.</li> <li>record/ la struttura</li> <li>Le tabelle</li> </ul>
4	35	P2  M6	<b>LA OOP</b>  MAGGIO-GIUGNO	Capacità di realizzare semplici programmi stand-alone; Capacità di progettare e realizzare semplici applicazioni	<ul style="list-style-type: none"> <li>Lezione frontale</li> <li>Lezione interattiva</li> <li>problem solving</li> </ul>	<p>Introduzione alla OOP:</p> <ul style="list-style-type: none"> <li>Le classi: attributi e metodi.</li> <li>Gli oggetti.</li> <li>L'incapsulamento.</li> <li>Visibilità privata e pubblica.</li> <li>Il costruttore.</li> <li>Interazione tra gli oggetti.</li> <li>Concetti fondamentali di ereditarietà e polimorfismo: overriding, overloading, riferimento this, super</li> </ul> <p>Metodi a collegamento statico e dinamico di un metodo. Persistenza di un oggetto.</p>
5(*)	16	P6  P10	La gestione delle liste nella programmazione strutturata e nella programmazione ad oggetti	Concetto di interfaccia Classi: List ArrayList	<ul style="list-style-type: none"> <li>Lezione Interattiva</li> <li>Laboratorio</li> <li>FAD</li> </ul>	Saper gestire collezioni di oggetti tramite strutture dati predefinite
6(**)	24	P3  P10	JAVASCRIPT gestione lato client	Sintassi di base Funzioni Interazione con il DOM Convalida di form	<ul style="list-style-type: none"> <li>Lezione Interattiva</li> <li>Laboratorio</li> <li>FAD</li> </ul>	Gestire un sito web lato client in modo dinamico

(\*) Il modulo 5 rappresenta un modulo FAD interdisciplinare comune con TPSI della durata complessiva di 25 ore (9 ore di TPSI e 16 ore di INFORMATICA)

(\*\*) Il modulo 6 rappresenta un modulo FAD interdisciplinare in comune con Sistemi e Reti della durata complessiva di 50 ore erogato in FAD (26 ore in SER e 24 ore in INFORMATICA)

### **Esercitazioni e applicazioni in laboratorio**

- Capacità di utilizzare strumenti SW
- documentazione del lavoro svolto
- Utilizzare il lessico e la terminologia tecnica di settore anche in lingua inglese

Note: Il percorso di laboratorio, per quasi tutti gli argomenti, è parallelo allo svolgimento degli argomenti teorici e si basa sugli strumenti SW di sviluppo disponibili sul mercato open source

**P2** - Individuare e utilizzare gli strumenti di comunicazione e di team working più appropriati per intervenire nei contesti organizzativi e professionali di riferimento

**M6** - utilizzare le strategie del pensiero razionale negli aspetti dialettici ed algoritmici per affrontare situazioni problematiche elaborando opportune soluzioni;

	<b><u>TIPOLOGIA DI PROVE DI VERIFICA</u></b>	<b><u>SCANSIONE TEMPORALE</u></b>
	Interrogazione lunga	Numero minimo di verifiche sommative previste per il quadrimestre: 4
X	Interrogazione breve	
X	Tema o problema	
X	Prove strutturate	
X	Prove semistrutturate	
	Prove grafiche	
X	Prove pratiche	
	Questionario	
	Relazione	
X	Esercizi	
	Altro da specificare	

### **MATERIALE DIDATTICO:**

Libri di testo consigliati:

- Informatica Generale vol. 1, Gallo P., Salerno F. (Minerva Italica).
- Programmare in C, Lorenzi A., Meriggia V., (Atlas)

Appunti, dispense e presentazioni forniti dal docente su supporto cartaceo oppure multimediale.