PROGRAMMAZIONE DISCIPLINARE PER COMPETENZE ARTICOLATA PER UDA

ANNO SCOLASTICO **2026/27**

INDIRIZZO **ITIS INFORMATICA**

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

DISCIPLINA **INFORMATICA**

DOCENTI **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**

Competenze disciplinari del Terzo e Quarto anno triennio Itis	• Realizzazione di strategie risolutive di semplici problemi • Individuazione delle parti fondamentali di un programma codificato nel linguaggio studiato • Riconoscimento ed uso delle strutture di controllo fondamentali • Individuazione delle variabili di i/o e dichiarazione di variabili • Acquisizione delle nozioni di base necessarie all'attività di laboratorio (ambiente di lavoro) • Documentazione dei lavori e sviluppo con metodologia top down • Analisi di problemi che implichino l'uso di strutture dati fondamentali (vettori, matrici, tabelle, file) • Codifica delle soluzioni ai problemi proposti nei linguaggi studiati • La Object Oriented Programming • Consapevole uso del sw introdotto

OBIETTIVI DI COMPETENZA		ARTICOLAZIONE DELLE COMPETENZE IN ABILITA' E CONOSCENZE	
COMPETENZE IN ESITO	**COMP MINIST**	**ABILITA'**	**CONOSCENZE DELLA DISCIPLINA**
Analizzare il ruolo dell'elaboratore in un generico processo di elaborazione. Riconoscere un problema di tipo logico	M6 P2 P6 P11	Conoscere i principi alla base del funzionamento di un sistema di elaborazione	• Modello logico funzionale di una macchina a controllo di programma • Modello esecutivo: il modello classico di Von Neumann
		Saper rappresentare un'informazione in codice	• Ruolo del Sistema Operativo • Il mondo dei problemi: classificazioni e generalizzazioni
		Saper risolvere un semplice problema	• Concetto di algoritmo come soluzione di un problema parametrico
		Sapere utilizzare i formalismi per la rappresentazione di algoritmi	• Metodi di rappresentazione degli algoritmi • Tecniche di verifica di un algoritmo • Costruzione di algoritmi relativi a problemi di varia natura e loro rappresentazione
Riconoscere il ruolo delle variabili strutturate nelle fasi della programmazione. Utilizzare le variabili strutturate per la risoluzione di un problema	M6 P2 P6 P11	Conoscere la differenza tra variabili semplici e variabili strutturate	• 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	• Array monodimensionali: vettori • Inizializzazione e gestione dei vettori • Elaborazioni classiche degli array: • Array bidimensionali:matrici Inizializzazione e gestione di matrici
Saper utilizzare le funzioni predefinite Definire una funzione Definire la modalità del passaggio dei parametri	M6 P2 P6 P11	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
Conoscere la programmazione OOP e saper progettare semplici applicazioni.	M6 P2 P6 P11	Capacità di realizzare semplici programmi stand-alone; Capacità di progettare e realizzare semplici applicazioni	• Concetto di OOP: • Interazione tra gli oggetti. • Concetti fondamentali di ereditarietà e polimorfismo.

La gestione delle liste nella programmazione strutturata e nella programmazione ad oggetti	P6 P10	Saper gestire collezioni di oggetti tramite strutture dati predefinite	• Concetto di interfaccia Classi: List ArrayList
Gestione pagine web lato client JAVASCRIPT	P3 P10	Gestire un sito web lato client in modo dinamico	• Conoscere la sintassi di base e le funzioni. • Iterazione con il DOM. • Saper convalidare un form.

CONTENUTI DEL PROGRAMMA

UDA	ore	COMP MINIS	Titolo	Contenuti delle conoscenze	Metodologia	Abilità	Tipologie di verifica
1	60 presenza	M6 P2 P6 P11	GLI ALGORITMI	*Risolvere problemi* *Realizzare algoritmi per risolvere semplici problemi matematici o logici sia nella forma di diagrammi di flusso che in linguaggio C* Scegliere il tipo corretto di dati e variabili; Gli operatori aritmetici, logici e del confronto con precedenza e associatività; Procedure per acquisire informazioni dall'esterno e comunicare i risultati .	• Lezione frontale • Lezione interattiva • Problem solving	☐ Dal problema all'algoritmo ☐ Analisi e comprensione del problema; ☐ Analisi dei dati in ingresso e uscita; ☐ individuazione delle "variabili" utili alla costruzione ☐ Istruzioni operative, di input/output, di controllo (sequenza, selezione e iterazione) ☐ Strategie di soluzione ☐ Scrittura dell'algoritmo in linguaggio di progetto ☐ Simbologia dei diagrammi di flusso ☐ Stesura del diagramma di flusso di semplici algoritmi	Interrogazione lunga Interrogazione breve Tema o Problema Prove strutturate Prove semi strutturate Prove pratiche Questionari Relazioni Esercizi
2	90 presenza	M6 P2 P6 P11	IL LINGUAGGIO C	Scegliere il tipo corretto di dati e variabili; Gli operatori aritmetici, logici e del confronto con precedenza e associatività; Procedure per acquisire informazioni dall'esterno e comunicare i risultati . Utilizzare struttura di selezione semplice e nidificata Utilizzare struttura di selezione multipla Utilizzare struttura di iterazione pre e post condizionale. Le iterazioni derivate Abilità nel dettaglio Le varie Funzioni di libreria. Risolvere problemi secondo il metodo top down e documentare il lavoro svolto secondo gli standard della programmazione C. Progettare e implementare algoritmi	• Lezione frontale • Lezione interattiva • Problem solving	☐ Programmazione ☐ Visibilità ed ambiente di esecuzione ☐ Variabili, costanti, tipi valore, tipi riferimento ☐ Sintassi delle funzioni predefinite ☐ Modificare l'ordine di esecuzione delle istruzioni ☐ Funzione degli operatori nel calcolo di espressioni ☐ Vettori, matrici, stringhe.	Interrogazione lunga Interrogazione breve Tema o Problema Prove strutturate Prove semi strutturate Prove pratiche Questionari Relazioni Esercizi

				utilizzando diverse strutture di dati: Dimensionare e utilizzare correttamente dati strutturati. Analizzare e confrontare algoritmi diversi per la soluzione dello stesso problema			
3	50 presenza	M6 P2 P6 P11	LE FUNZIONI	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	• Lezione frontale • Lezione interattiva • Problem solving	☐ Funzioni e librerie ☐ Funzioni, parametri per riferimento e per valore. ☐ Funzioni ricorsive. ☐ Gli algoritmi fondamentali di ricerca e ordinamento di un vettore e successivamente delle altre strutture dati. ☐ record/ la struttura ☐ Le tabelle	Interrogazione breve Tema o Problema Prove semi strutturate Prove pratiche Questionari Esercizi
4	30 presenza	M6 P2 P6 P11	LA OOP	Capacità di realizzare semplici programmi stand-alone; Capacità di progettare e realizzare semplici applicazioni	• Lezione frontale • Lezione interattiva • Problem solving	☐ Introduzione alla OOP: ○ Le classi: attributi e metodi. ○ Gli oggetti. ○ L'incapsulamento. ○ Visibilità privata e pubblica. ○ Il costruttore. ○ Interazione tra gli oggetti. ○ Concetti fondamentali di ereditarietà e polimorfismo: overriding, overloading, riferimento this, super ☐ Metodi a collegamento statico e dinamico. ☐ Persistenza di un oggetto.	Interrogazione breve Tema o Problema Prove semi strutturate Prove pratiche Questionari Esercizi
5	16 FAD	P6 P10	LA GESTIONE DELLE LISTE	Capacità di saper gestire collezioni di oggetti tramite strutture dati predefinite	• Lezione frontale • Lezione interattiva • Problem solving	☐ Concetto di interfaccia ○ Classi: ▪ List ▪ ArrayList	Prove semi strutturate Prove pratiche Questionari Esercizi
6	24 FAD	P3 P10	GESTIONE PAGINE WEB LATO CLIENT JAVASCRIPT	Capacità di gestire un sito web lato client in modo dinamico	• Lezione frontale • Lezione interattiva • Problem solving	☐ Sintassi di base ☐ Funzioni ☐ Interazione con il DOM ☐ Convalida di form	Prove semi strutturate Prove pratiche Questionari Esercizi

N.B. *Nel modulo 1 la parte in corsivo della durata di 12 ore sono da considerarsi in chiave orientativa.*

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

Competenze - Informatica
P1- Riconoscere, descrivere e comparare il funzionamento e la struttura di sistemi e dispositivi informatici.
P2 - Individuare e utilizzare gli strumenti di comunicazione e di team working
P6 - Intervenire nelle fasi di analisi, progettazione, installazione e manutenzione di sistemi e reti
P7 - Gestire basi di dati e realizzare interfacce per la loro fruizione
P8 - Configurare, installare e gestire sistemi di elaborazione e reti locali
P10 - Sviluppare applicazioni informatiche (reti locali o servizi a distanza)
P11 - Analizzare e comparare applicazioni informatiche per soluzioni ottimali
P12 - Progettare, realizzare e gestire applicazioni orientate ai servizi e distribuite
M6 - utilizzare le strategie del pensiero razionale negli aspetti dialettici ed algoritmici per affrontare situazioni problematiche elaborando opportune soluzioni;

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 sopporto cartaceo oppure multimediale

RIEPILOGO ORE E SINTESI UDA

UDA	Titolo	ORE		Obiettivo sintetico	Tipologia Verifica
		PRESENZA	FAD		
UDA 1	GLI ALGORITMI	60		Realizzare algoritmi, con diagrammi di flusso, che utilizzino tutte le istruzioni fondamentali della programmazione imperativa	Tema o Problema Prove semi strutturate Prove pratiche Questionari Esercizi
UDA 2	IL LINGUAGGIO C	90		Codificare ed implementare nel linguaggio C tutti gli algoritmi visti nelle ore di lezione con le diverse strutture dati studiate.	Tema o Problema Interrogazione breve. Prove semi strutturate Prove pratiche. Realizzazione applicazioni da consegnare Esercizi
UDA 3	LE FUNZIONI	50		Utilizzo consapevole delle funzioni predefinite; Realizzare procedure e funzioni parametriche, con corretto utilizzo del passaggio dei parametri.	Tema o Problema Interrogazione breve. Prove semi strutturate Prove pratiche Realizzazione applicazioni da consegnare Esercizi
UDA 4	LA OOP	30		Capacità di progettare e realizzare semplici applicazioni stand-alone;	Tema o Problema Prove semi strutturate Prove pratiche Realizzazione applicazioni da consegnare Esercizi
UDA 5	LA GESTIONE DELLE LISTE		16	Saper gestire collezioni di oggetti tramite strutture dati predefinite	Prove pratiche Realizzazione applicazioni da consegnare Esercizi
UDA 6	GESTIONE PAGINE WEB LATO CLIENT JAVASCRIPT		24	Saper gestire un sito web lato client in modo dinamico	Prove pratiche Realizzazione applicazioni da consegnare Esercizi

Data Compilazione: 13/11/2026