

PROGRAMMAZIONE DISCIPLINARE PER COMPETENZE ARTICOLATA PER UDA

ANNO SCOLASTICO 2020/21

INDIRIZZO ITIS INFORMATICA

CLASSE 2 Periodo Didattico SEZIONE A INF

DISCIPLINA INFORMATICA

DOCENTE GIUSEPPE LEMME e GIUSEPPE BISCEGLIA

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

1. QUADRO DEGLI OBIETTIVI DI COMPETENZA

ASSE CULTURALE: Tecnico - Scientifico

<u>Competenze disciplinari del Terzo e Quarto anno triennio Itis</u>	<ul style="list-style-type: none">• 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 implicano 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	ABILITA'	CONOSCENZE DELL'ASSE	CONOSCENZE DELLA DISCIPLINA
	<ul style="list-style-type: none"> Analizzare il ruolo dell'elaboratore in un generico processo di elaborazione Riconoscere un problema di tipo logico 	Conoscere i principi alla base del funzionamento di un sistema di elaborazione	Sistemi informatici	<ul style="list-style-type: none"> Modello logico funzionale di una macchina a controllo di programma Modello esecutivo: il modello classico di Von Neumann
		Saper rappresentare un'informazione in codice	I codici	<ul style="list-style-type: none"> Ruolo del Sistema Operativo Il mondo dei problemi: classificazioni e generalizzazioni
		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"> Metodi di rappresentazione degli algoritmi Tecniche di verifica di un algoritmo Costruzione di algoritmi relativi a problemi di varia natura e loro rappresentazione
	<ul style="list-style-type: none"> Riconoscere il ruolo delle variabili strutturate nelle fasi della programmazione Utilizzare le variabili strutturate per la risoluzione di un problema 	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"> Massimo, minimo, media Ricerca Ordinamento Merge Array bidimensionali: matrici Inizializzazione e gestione di matrici
	<ul style="list-style-type: none"> Saper utilizzare le funzioni predefinite Definire una funzione Definire la modalità del passaggio dei parametri 	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

CONTENUTI DEL PROGRAMMA						
UDA	ore	COMP UDA	Titolo	Attività docente	Metodologia	Prestazioni studente
n. 1	45	P2	GLI ALGORITMI SETTEMBRE-OTTOBRE	<p>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 .</p>	<ul style="list-style-type: none"> Lezione frontale Lezione interattiva Problem solving 	<ul style="list-style-type: none"> Dal problema all'algoritmo Analisi e comprensione del problema; Analisi dei dati in ingresso e di quelli che si vogliono ottenere in uscita; individuazione delle "variabili" utili alla costruzione dell'algoritmo ed eliminazione di eventuali dati inutili Istruzioni operative, di input/output, di controllo (selezione, iterazione) Strategie di soluzione Scrittura dell'algoritmo in linguaggio di progetto Simbologia dei diagrammi di flusso Stesura del diagramma di flusso di semplici algoritmi
2	90	P2 M6	IL LINGUAGGIO C NOVEMBRE-FEBBRAIO	<p>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 .</p> <p>Utilizzare struttura di selezione semplice e nidificata Utilizzare struttura di selezione multipla Utilizzare struttura di iterazione 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.</p> <p>Progettare e implementare algoritmi utilizzando diverse strutture di dati: Dimensionare e utilizzare correttamente dati strutturati. □Analizzare e confrontare algoritmi diversi per la soluzione dello stesso</p>	<ul style="list-style-type: none"> Lezione frontale Lezione interattiva Problem solving 	<ul style="list-style-type: none"> 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.

				problema		
--	--	--	--	----------	--	--

3	56	P2 M6	LE FUNZIONI 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"> Lezione frontale Lezione interattiva problem solving 	<ul style="list-style-type: none"> Funzioni e librerie Funzioni, parametri per riferimento e per valore, Funzioni ricorsive. Gli algoritmi fondamentali di ricerca e ordinamento di un vettore e successivamente alle altre strutture dati. record/ la struttura
4	35	P2 M6	LA OOP MAGGIO-GIUGNO	Capacità di realizzare semplici programmi stand-alone; Capacità di progettare e realizzare semplici applicazioni	<ul style="list-style-type: none"> Lezione frontale Lezione interattiva problem solving 	<p>Introduzione alla OOP:</p> <ul style="list-style-type: none"> 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 <p>Metodi a collegamento statico e dinamico di un metodo. Persistenza di un oggetto.</p>

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

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;

	<u>TIPOLOGIA DI PROVE DI VERIFICA</u>	<u>SCANSIONE TEMPORALE</u>
	Interrogazione lunga	Numero minimo di verifiche sommative previste per il quadrimestre: 3
	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.