

## PROGRAMMA EFFETTIVAMENTE SVOLTO

Anno Scolastico 23 / 24

**Classe/Sede:** 3D2

**Docente:** Prof. Emidio Vecchione

**Codocente (ITP):** Prof. Mattia Bedani

**Materia insegnata:** Tecnologie Informatiche

**Testi adottati:** Slide del corso, piattaforma e-learning

MODULI E UNITÀ DI APPRENDIMENTO <sup>1</sup>		
N.B.: i contenuti didattici minimi per il passaggio alla classe successiva sono <u>sottolineati</u>		
TITOLO	CONTENUTI DIDATTICI <sup>2</sup>	TEMPI
<b>Modulo 1:</b> <b>Introduzione all'Informatica e Algoritmi</b>	1.1 – Concetti di base: definizione di informatica; dati e informazioni; schema di un sistema di elaborazione dati; macchina di von Neumann; tipi di computer; breve storia dell'Informatica 1.2 – <u>Concetto di algoritmo</u> : definizione formale di algoritmo; problemi e algoritmi; rappresentazione degli algoritmi; concetto di primitiva; rappresentazione degli algoritmi con pseudocodice e diagrammi di flusso; procedure 1.3 – <u>Scoperta di algoritmi</u> : arte della risoluzione dei problemi; trovare un punto di partenza; variabili e costanti (literal e simboliche); espressioni; assegnazione 1.4 – <u>Programmazione strutturata</u> : schemi di composizione delle istruzioni (sequenza, selezione, iterazione); cicli pre e post condizionali; equivalenza e passaggio da ciclo con pre-test a ciclo con post-test; teorema di Böhm-Jacopini; espressioni condizionali composte con gli operatori logici; ciclo con contatore; i vantaggi della programmazione strutturata; la documentazione del lavoro; l'utilizzo delle tabelle di traccia. Analisi di problemi e sviluppo degli algoritmi risolutivi.	Settembre – novembre
<b>Modulo 2:</b> <b>Linguaggi e strumenti per la programmazione – Linguaggio C#</b>	2.1 – Introduzione: definizione di linguaggio di programmazione; generazioni dei linguaggi di programmazione; concetto di paradigma di programmazione e caratteristiche dei paradigmi imperativo, dichiarativo, funzionale e orientato agli oggetti; alcuni importanti linguaggi di programmazione	Ottobre – febbraio

<sup>1</sup> Parte significativa, omogenea ed unitaria, del percorso formativo. Può essere di raccordo, di metodo, disciplinare o pluridisciplinare. Può essere eventualmente organizzato come U.d.A. (unità di apprendimento significativo, volte a sviluppare competenze disciplinari e trasversali attraverso l'utilizzo della didattica laboratoriale e di prove esperte).

<sup>2</sup> Contenuti del modulo articolati in unità didattiche (lezioni, capitoli, ecc.)

	<p>2.2 – <u>Sviluppo del software</u>: processo di traduzione; interpreti e compilatori; linguaggi con macchina virtuale; fasi di sviluppo di un programma (editing, compiling, linking, debugging – differenza tra linking statico e dinamico); DLL in Windows e Share Object in *nix. Ambienti integrati per lo sviluppo (IDE); ciclo di sviluppo del software</p> <p>2.3 – Piattaforma .NET: la piattaforma .NET; linguaggi .NET; Common Language Runtime (CLR); Intermediate Language (IL); confronto tra compilatore classico e compilatore IL; implementazioni alternative; .NET Core</p> <p>2.4 – <u>Primi elementi del linguaggio C#</u>: scheletro (semplificato) di un programma C# e metodo Main; concetto di namespace; classe Console; utilizzo di Visual Studio; direttiva using static; dichiarazione di variabili e costanti in C#; tipi valore e tipi riferimento; tipi valore predefiniti; conversione fra tipi (espressioni cast e classe Convert); conversione controllata con il metodo TryParse; tipizzazione implicita con var; il valore speciale null; ambito di visibilità (scope) delle variabili; operatori aritmetici, relazionali, logici e condizionali; precedenza degli operatori; operazioni di assegnazione semplice e composta; strutture di controllo in C# (costrutti if, switch, while, do ... while, for); istruzioni break e continue.</p>	
<p><b>Modulo 3:</b> <b>Strutture dati e algoritmi fondamentali, metodologia top-down</b></p>	<p>3.1 – <u>Stringhe in C#</u>: la classe String; stringhe verbatim; concatenazione e interpolazione di stringhe; caratteri di escape e stringhe verbatim; conversione in stringa; accesso ai singoli caratteri di una stringa; metodi per operare sulle stringhe</p> <p>3.2 – <u>Array in C#</u>: dichiarazione di array ad una o più dimensioni; allocazione e inizializzazione di array; array jagged; accesso ai singoli elementi e numero di elementi; il ciclo foreach; assegnazione e confronto tra array; metodi della classe Array; metodi sulle stringhe che usano array</p> <p>3.3 – <u>Algoritmi di ricerca e ordinamento sui vettori in C#</u>: algoritmo di ricerca sequenziale; determinazione del valore minimo e del valore massimo; algoritmi di ordinamento per selezione, per scambio e per inserzione; algoritmo di ricerca binaria (versione iterativa);</p> <p>3.4 – <u>Metodologia top-down e sottoprogrammi in C#</u>: approccio modulare top-down; sottoprogrammi (procedure e funzioni); metodi in C#; elementi essenziali di OOP e metodi statici; campi dato, variabili locali e ambito di visibilità; metodi funzione e valore di ritorno; metodi con parametri; passaggio dei parametri per valore; passaggio di parametri array ad un metodo; passaggio dei parametri per riferimento; commenti di documentazione in Visual Studio;</p>	<p>Gennaio – maggio</p>
<p><b>Modulo 4:</b> <b>Sviluppo web lato client</b></p>	<p>4.1 – Sviluppo web: breve cronistoria del web; ruolo, utilizzo e storia dei linguaggi HTML, CSS e JavaScript; come si progetta un sito web; strumenti per lo sviluppatore web; il Document Object Model (DOM) di una pagina web; il Browser Object Model (BOM); Responsive Web Design</p> <p>4.2 – HTML: anatomia di un elemento HTML; sintassi dei tag; struttura base di un documento HTML; meta tag; sintassi degli attributi degli elementi; attributi globali; elementi di blocco e in linea; elementi per il testo ed elementi semantici introdotti da HTML5; elementi per elenchi, tabelle, link, immagini, mappe, video e audio</p> <p>4.3 – CSS: sintassi dei fogli di stile; selettori CSS (semplici, combinatori, pseudo classi, pseudo elementi e in base ad attributi); posizionamento degli stili (esterni, interni, in linea); regole di applicazione degli stili; set di caratteri (@charset); importazione di fogli esterni (@import); caricamento di font (@font face); media queries (@media); animazioni degli stili (@keyframes); stili per la formattazione di testo; tipologie di font (serif, sans serif e monospaced); stili per bordi, margini, padding; box model; stili per colori del testo e background; effetti su testo e background; layout responsive flex e grid</p>	<p>Maggio – Giugno</p>

Valdagno, \_\_\_\_\_

Studenti

\_\_\_\_\_

\_\_\_\_\_

Docenti

\_\_\_\_\_

\_\_\_\_\_