CLASSE: 1NF MATERIA: INFORMATICA E RETI DI COMUNICAZIONE
DOCENTE: SINOPOLI

PROGRAMMAZIONE DIDATTICA A.S. 2026-2027

☐ ASSE CULTURALE DEI LINGUAGGI
☒ ASSE CULTURALE SCIENTIFICO TECNOLOGICO
☐ ASSE CULTURALE MATEMATICO
☐ ASSE CULTURALE STORICO-SOCIALE

competenze chiave	competenze base	conoscenze	abilità
Programmazione in Python	• Utilizzare il linguaggio Python per analizzare problemi e sviluppare semplici soluzioni algoritmiche. • Individuare e applicare le strutture di controllo più appropriate in relazione alle caratteristiche del problema. • Verificare la correttezza dei programmi attraverso attività di analisi, esecuzione e correzione del codice.	• Principali caratteristiche e regole sintattiche del linguaggio Python. • Variabili, tipi di dati e operatori aritmetici, relazionali e logici. • Istruzioni di input e output. • Strutture di selezione semplice, doppia e multipla. • Strutture iterative definite e indefinite. • Concetto di annidamento delle istruzioni. • Principali tecniche di verifica e individuazione degli errori nei programmi.	• Scrivere programmi utilizzando istruzioni di input, output e assegnazione. • Utilizzare le strutture di selezione per gestire condizioni semplici e composte. • Applicare correttamente i cicli while e for. • Realizzare programmi che combinino strutture di selezione e iterazione, anche annidate. • Individuare il costrutto più appropriato per la risoluzione di un problema. • Analizzare l'esecuzione di un programma per individuare e correggere eventuali errori.
Le funzioni in Python	• Progettare programmi strutturati attraverso la	• Concetto di funzione e principi della	• Definire e richiamare funzioni in Python.

	suddivisione di problemi complessi in sottoproblemi. • Utilizzare le funzioni per migliorare l'organizzazione, la leggibilità e la riutilizzabilità del codice. • Applicare strategie di programmazione modulare per la risoluzione di problemi	programmazione modulare. • Definizione e richiamo delle funzioni in Python. • Differenza tra funzioni predefinite e funzioni definite dall'utente. • Parametri formali e argomenti effettivi. • Modalità di passaggio degli argomenti e restituzione dei valori. • Concetto di variabile locale e globale e ambito di visibilità. • Concetto di ricorsione e differenze rispetto all'iterazione	• Utilizzare funzioni predefinite e realizzare funzioni personalizzate. • Definire funzioni con parametri e valori di ritorno. • Distinguere e utilizzare correttamente variabili locali e globali. • Scomporre un problema in sottoproblemi e organizzare il programma mediante funzioni. • Realizzare programmi che prevedano l'utilizzo combinato di più funzioni. • Riconoscere semplici situazioni nelle quali può essere applicata la ricorsione. • Verificare il corretto funzionamento delle funzioni attraverso test e analisi dei risultati.
Strutture complesse in Python	• Utilizzare strutture dati appropriate per organizzare, memorizzare ed elaborare insiemi di informazioni. • Individuare e applicare strategie algoritmiche per la gestione di	• Concetto di struttura dati e differenza tra dati semplici e strutturati. • Caratteristiche degli array monodimensionali (vettori) e bidimensionali (matrici).	• Creare e utilizzare vettori e matrici in Python. • Accedere, modificare ed elaborare gli elementi di strutture dati monodimensionali e bidimensionali.

	dati organizzati in strutture monodimensionali e bidimensionali. • Sviluppare programmi che utilizzino strutture dati complesse per la risoluzione di problemi. • Selezionare le strutture dati più adeguate in relazione alle caratteristiche delle informazioni da gestire	• Modalità di rappresentazione di vettori e matrici in Python. • Concetto di indice e accesso agli elementi di una struttura dati. • Principali operazioni di inserimento, modifica, ricerca ed elaborazione dei dati. • Caratteristiche delle stringhe e principali operazioni sui caratteri e sulle sottostringhe. • Liste e tuple: caratteristiche, differenze e modalità di utilizzo. • Insiemi (set): caratteristiche e operazioni fondamentali. • Dizionari: organizzazione delle informazioni mediante coppie chiave-valore. •	• Utilizzare cicli per attraversare vettori e matrici. • Applicare algoritmi di ricerca ed elaborazione dei dati. • Eseguire operazioni su stringhe, caratteri e sottostringhe. • Creare e manipolare liste e tuple. • Utilizzare gli insiemi per rappresentare collezioni di elementi ed effettuare operazioni insiemistiche. • Creare e utilizzare dizionari per organizzare e recuperare informazioni. • Individuare la struttura dati più appropriata per la risoluzione di un problema. • Realizzare programmi che integrino strutture dati, funzioni e strutture di controllo
Programmazione in C++	• Utilizzare il linguaggio Python per analizzare problemi e sviluppare semplici	• Principali caratteristiche e regole sintattiche del linguaggio C++. • Variabili, tipi di dati e operatori aritmetici,	• Scrivere programmi utilizzando istruzioni di input, output e assegnazione. • Utilizzare le strutture di

	soluzioni algoritmiche. • Individuare e applicare le strutture di controllo più appropriate in relazione alle caratteristiche del problema. • Verificare la correttezza dei programmi attraverso attività di analisi, esecuzione e correzione del codice.	relazionali e logici. • Istruzioni di input e output. • Strutture di selezione semplice, doppia e multipla. • Strutture iterative definite e indefinite. • Concetto di annidamento delle istruzioni. • Principali tecniche di verifica e individuazione degli errori nei programmi.	selezione per gestire condizioni semplici e composte. • Applicare correttamente i cicli while e for. • Realizzare programmi che combinino strutture di selezione e iterazione, anche annidate. • Individuare il costrutto più appropriato per la risoluzione di un problema. • Analizzare l'esecuzione di un programma per individuare e correggere eventuali errori.
Siti Web e HTML	• Progettare e realizzare semplici siti Web, organizzando contenuti e collegamenti in modo coerente. • Utilizzare il linguaggio HTML per strutturare e rappresentare informazioni sul Web. • Valutare le caratteristiche di un sito Web considerando principi di usabilità e accessibilità	• Caratteristiche e funzionamento generale del Web. • Differenze tra siti Web statici e dinamici. • Principali fasi della progettazione e realizzazione di un sito Web. • Concetti fondamentali di usabilità e accessibilità. • Struttura di un documento HTML e principali elementi del linguaggio. • Tag per la formattazione e l'organizzazione dei contenuti testuali.	• Individuare gli elementi necessari alla progettazione di un semplice sito Web. • Organizzare i contenuti di un sito attraverso una struttura logica di pagine e collegamenti. • Creare documenti HTML rispettando la sintassi del linguaggio. • Utilizzare i principali tag HTML per strutturare testi, elenchi e tabelle. • Inserire immagini, collegamenti ipertestuali e contenuti

		• Elementi HTML per liste, immagini, collegamenti ipertestuali e tabelle. • Elementi HTML per l'inserimento di contenuti multimediali	multimediali nelle pagine Web. • Realizzare un semplice sito composto da più pagine collegate tra loro. • Riconoscere e valutare gli elementi fondamentali di usabilità e accessibilità di un sito Web. • Verificare la corretta visualizzazione e il funzionamento delle pagine realizzate
Siti Web e HTML	• Progettare e realizzare semplici siti Web, organizzando contenuti e collegamenti in modo coerente. • Utilizzare il linguaggio HTML per strutturare e rappresentare informazioni sul Web. • Valutare le caratteristiche di un sito Web considerando principi di usabilità e accessibilità	• Caratteristiche e funzionamento generale del Web. • Differenze tra siti Web statici e dinamici. • Principali fasi della progettazione e realizzazione di un sito Web. • Concetti fondamentali di usabilità e accessibilità. • Struttura di un documento HTML e principali elementi del linguaggio. • Tag per la formattazione e l'organizzazione dei contenuti testuali. • Elementi HTML per liste, immagini, collegamenti ipertestuali e tabelle. • Elementi HTML per l'inserimento di contenuti multimediali	• Individuare gli elementi necessari alla progettazione di un semplice sito Web. • Organizzare i contenuti di un sito attraverso una struttura logica di pagine e collegamenti. • Creare documenti HTML rispettando la sintassi del linguaggio. • Utilizzare i principali tag HTML per strutturare testi, elenchi e tabelle. • Inserire immagini, collegamenti ipertestuali e contenuti multimediali nelle pagine Web. • Realizzare un semplice sito composto da più pagine collegate tra loro.

			• Riconoscere e valutare gli elementi fondamentali di usabilità e accessibilità di un sito Web. • Verificare la corretta visualizzazione e il funzionamento delle pagine realizzate

CONTENUTI DEL PROGRAMMA:

1. **Programmazione in Python**
    a. Programmazione base in Python
        i. Istruzioni in sequenza
        ii. La selezione: semplice, doppia, annidata, operatori logici e di confronto
        iii. I cicli while e i cicli for
    b. Le funzioni in Python
    c. Strutture dati complesse in Python
        i. Creazione degli array monodimensionali (vettori) e bidimensionali (matrici) in Python
        ii. Operazioni con gli array

2. **Programmazione in C++**
    a. Programmazione base in C++
        i. Tipi di dati, variabili e costanti
        ii. Dichiarazione delle variabili
        iii. Istruzioni in sequenza
        iv. La selezione: semplice, doppia, annidata, operatori logici e di confronto
        v. I cicli while e i cicli for

3. **Siti web e HTML**
    a. L'HTML
    b. La sintassi HTML
    c. Il corpo del documento
    d. La formattazione del testo
    e. Le liste numerate e puntate
    f. L'inserimento di immagini
    g. I link
    h. Le tabelle HTML
    i. Inserire file audio e video
    j. I fogli di stile

Attività del docente e metodologia	**Attività dello studente**	**Materiali e spazi utilizzati**
• Esporre le conoscenze agli studenti nelle lezioni frontali utilizzando il supporto multimediale. • Assegnare agli alunni i riferimenti sul libro e i compiti settimanali da svolgere. • Assicurarsi che gli studenti comprendano le consegne	• Comprendere un messaggio orale o scritto • Utilizzare il linguaggio proprio della disciplina • Esporre con chiarezza il proprio pensiero • Prendere appunti, annotare informazioni e saperli riorganizzarli in schemi logici, mappe concettuali, mappe	• Libri di testo, e-book, presentazioni Canva • Laboratori e PC in ambiente Windows • LIM • Software di laboratorio

• Programmare le unit, di apprendimento. • Creare motivazioni, stimolare all'invenzione, orientare le attività pratiche. Valutare i risultati delle unit, di apprendimento. • Insistere sull'uso di un linguaggio corretto • Insegnare a prendere appunti, a recepire, di un testo, ciò che è significativo • Favorire il dialogo in classe, incentivare la curiosità e sviluppare la creatività • Sviluppare la capacità di individuare, scegliere, utilizzare informazioni di varia natura • Fornire gli strumenti utili alla risoluzione di problemi • Riprogrammare le eventuali unit, di apprendimento di recupero per gli alunni che non hanno conseguito gli obiettivi perseguiti	mentali (usando anche tecnologie appropriate) • Scoprire analogie e differenze, sviluppare un modello utile alla risoluzione del problema in esame • Partecipare positivamente all'attività didattica, ponendo domande, proponendo soluzioni, riflettendo sugli interventi di docente e compagni • Lavorare in gruppo, condividendo informazioni e azioni e accogliendo suggerimenti o ipotesi alternative • Effettuare semplici esperimenti e trarne conclusioni pertinenti • Acquisire un metodo di studio proficuo che gli consenta di razionalizzare il tempo dedicato allo studio e di giudicare il livello della propria preparazione per condurlo a una corretta autovalutazione	

VALUTAZIONE:

CONOSCENZE	VALUTAZIONE	ABILITA'	VALUTAZIONE	COMPETENZE	VALUTAZIONE
Espone ed inquadra nel corretto contesto gli argomento trattato. Usa con proprietà il linguaggio scientifico anche in lingua	9-10	Applica i procedimenti risolutivi in modo corretto e approfondito. Utilizza in modo completamente autonomo i software di laboratorio. Usa i formalismi dell'informatica in modo corretto.	9-10	Utilizza in modo completamente autonomo le strategie del pensiero razionale per risolvere problemi elaborando le opportune soluzioni dal punto di vista informatico.	Eccellente/ottimo

Espone in modo corretto gli argomenti trattati ed utilizza il lessico in modo appropriato anche in lingua	8	Utilizza in modo autonomo i software di laboratorio. Individua in modo autonomo le soluzioni ai problemi proposti e le realizza dal punto di vista informatico.	8	Utilizza in modo autonomo le strategie del pensiero razionale per risolvere problemi elaborando le opportune soluzioni dal punto di vista informatico.	buono
Espone in modo corretto gli argomenti trattati ed utilizza il lessico in modo accettabile anche in lingua	7	Utilizza in modo adeguato i software di laboratorio. Individua con discreta autonomia le soluzioni ai problemi proposti e le realizza dal punto di vista informatico.	7	Utilizza in modo corretto le strategie del pensiero razionale per risolvere problemi ed è in grado di elaborare in modo autonomo le opportune soluzioni dal punto di vista informatico.	discreto
Coglie gli elementi di base dell'argomento trattato. Qualche incertezza nell'uso del linguaggio tecnico.	6	Utilizza in modo parzialmente adeguato i software di laboratorio. Individua con parziale autonomia le soluzioni ai problemi proposti e le realizza dal punto di vista informatico.	6	Utilizza in modo parzialmente autonomo le strategie del pensiero razionale per risolvere problemi elaborando in modo quasi autonomo le opportune soluzioni dal punto di vista informatico.	sufficiente
Conosce in modo frammentario l'argomento trattato. Usa il linguaggio tecnico in modo incerto	5	Utilizza in modo non completamente adeguato i software di laboratorio. Individua con difficoltà le soluzioni ai problemi proposti e le realizza dal punto di vista informatico.	5	Utilizza in modo improprio le strategie del pensiero razionale per risolvere problemi e elabora con errori le opportune soluzioni dal punto di vista informatico	mediocre
Conosce in modo lacunoso l'argomento trattato. Gravi improprietà nell'uso del linguaggio tecnico.	4	Utilizza in modo inadeguato i software di laboratorio. Non è in grado di individuare le soluzioni ai problemi proposti e li implementa con gravi errori.	4	Utilizza in modo improprio le strategie del pensiero razionale per risolvere problemi e elabora con gravi errori le opportune soluzioni dal punto di vista informatico	insufficiente
Conosce in modo gravemente lacunoso l'argomento trattato. Gravi improprietà nell'uso del linguaggio tecnico.	2-3	Ignora completamente il funzionamento dei software di laboratorio. Non è in grado di individuare le soluzioni ai problemi proposti e non è nemmeno in grado di implementarli.	2-3	Non è in grado di utilizzare le strategie del pensiero razionale per risolvere problemi e elabora con gravi errori le opportune soluzioni dal punto di vista informatico	Gravemente insufficiente