

INFORMATICA classi 3G e 3K a.s. 2018-19 Prof. Gerardo Fariello	
MODULO	ARGOMENTO
Introduzione alla programmazione	
	Primi elementi di programmazione
	Dal problema al programma
	Variabili e costanti
	Le fasi di simulazione e codifica dell'algoritmo
	Diagrammi a blocchi
	Prime applicazioni dei diagrammi a blocchi
	Il blocco di controllo multidecisionale
	Gli operatori booleani
	Gli schemi di composizione fondamentali e la programmazione strutturata
	SCF di sequenza
	SCF di selezione
	SCF di iterazione (ripetizione)
	I cicli strutturati
	La struttura iterativa postcondizionale
	La struttura iterativa precondizionale
	Ciclo for
Le variabili strutturate e la loro manipolazione	
	Variabili semplici e variabili strutturate
	I vettori e la loro manipolazione
	L'elaborazione dei vettori
	Le matrici e la loro manipolazione
	L'elaborazione delle matrici
	Le stringhe di caratteri e la loro manipolazione
Gli algoritmi fondamentali	
	Gli algoritmi di ricerca
	Gli algoritmi di ordinamento
	Gli algoritmi di fusione
La metodologia top-down e i sottoprogrammi	
	L'approccio top-down e bottom-up
	La metodologia bottom-up
	I vantaggi della programmazione modulare
	I sottoprogrammi
	Parametri formali e attuali
	Passaggio di parametri
	Variabili globali e locali
	La ricorsione
Dalle strutture in memoria centrale a quelle in memoria di massa	
	I record e le tabelle
	Le tabelle
	Implementazione di una tabella
	Introduzione al concetto di file
	Operazione di base sui file di dati
Linguaggi e strumenti per la programmazione	
	I linguaggi di programmazione e i traduttori
	Sviluppo ed esecuzione di un programma
	I pacchetti IDE
	Classificazione dei linguaggi di programmazione

COMPETENZE:

Utilizzare le strategie del pensiero razionale negli aspetti dialettici e algoritmici per affrontare situazioni problematiche elaborando opportune soluzioni. (Obiettivo minimo: situazioni problematiche semplici.

Utilizzare le istruzioni di lettura, scrittura e assegnazione per risolvere problemi. (Obiettivo minimo)

Individuare le fasi necessarie per passare da un problema alla sua soluzione. (Obiettivo minimo: semplici problemi)

Saper riconoscere i dati di input e di output di un problema. (Obiettivo minimo)

Risolvere problemi che prevedono diverse condizioni. (Obiettivo minimo)

Risolvere problemi che prevedono cicli. (Obiettivo minimo: semplici problemi)

Saper affrontare un problema scomponendolo in sottoprogrammi. (Obiettivo minimo: semplici problemi)

Risolvere problemi usando le procedure e le funzioni quando necessario. (Obiettivo minimo: semplici problemi)

Saper caricare e visualizzare dati di un vettore. (Obiettivo minimo)

Saper cercare informazioni all'interno di strutture dati. (Obiettivo minimo)

Saper ordinare le strutture dati.

Saper fornire totali parziali elaborando i dati di un vettore. (Obiettivo minimo)

Saper fondere due vettori ordinati in un terzo mantenendo l'ordinamento.

Risolvere problemi con il supporto di strutture dati complesse.

CONOSCENZE:

Conoscere le relazioni fondamentali tra computer, problemi, informazioni e linguaggi.

Conoscere cos'è un linguaggio di programmazione. (Obiettivo minimo)

Capire i problemi esistenti nella comunicazione tra l'uomo e l'elaboratore. (Obiettivo minimo)

Conoscere il concetto di variabile come strumento per immagazzinare valori. (Obiettivo minimo)

Conoscere la simbologia utilizzata nei diagrammi di flusso. (Obiettivo minimo)

Conoscere le regole di base di un linguaggio di programmazione. (Obiettivo minimo)

Conoscere le strutture condizionali semplici. (Obiettivo minimo)

Conoscere le basi della logica simbolica e del calcolo proposizionale. (Obiettivo minimo)

Conoscere le strutture condizionali complesse. (Obiettivo minimo)

Conoscere la struttura iterativa con controllo in testa. (Obiettivo minimo)

Conoscere la struttura iterativa con controllo in coda. (Obiettivo minimo)

Conoscere la struttura iterativa con numero prefissato di cicli. (Obiettivo minimo)

Conoscere la progettazione top-down.

Conoscere la scomposizione in sottoprogrammi.

Conoscere le variabili locali e globali. (Obiettivo minimo)

Conoscere le procedure e le funzioni. (Obiettivo minimo)

Conoscere la differenza tra variabili semplici e variabili strutturate. (Obiettivo minimo)

Conoscere gli algoritmi fondamentali applicati alle strutture dati. (Obiettivo minimo)

Conoscere la matrice come struttura dati bidimensionale.

Conoscere il record come insieme di elementi non omogenei. (Obiettivo minimo)

Conoscere la tabella come vettore di record. (Obiettivo minimo)

ABILITA':

Padronanza della terminologia informatica di base. (Obiettivo Minimo)

Saper realizzare algoritmi che prevedono istruzioni di lettura, scrittura e assegnazione. (Obiettivo minimo: semplici algoritmi)

Saper descrivere algoritmi tramite i diagrammi di flusso. (Obiettivo minimo: semplici algoritmi)
Saper tradurre l'algoritmo in un programma mediante l'uso di un linguaggio di programmazione. (Obiettivo minimo: semplici algoritmi)
Individuare le strutture di controllo più idonee a risolvere un determinato problema. (Obiettivo minimo: semplici problemi)
Codificare programmi che includono le strutture condizionali. (Obiettivo minimo)
Testare algoritmi tramite casi di prova significativi. (Obiettivo minimo: semplici algoritmi)
Applicare agli algoritmi i principi della logica proposizionale.
Individuare il tipo di ciclo più adatto a risolvere un determinato problema.
Saper trasformare un ciclo postcondizionale in uno precondizionale e viceversa.
Codificare programmi che includono le strutture iterative. (Obiettivo minimo)
Scrivere programmi che utilizzino le funzioni. (Obiettivo minimo: semplici programmi)
Conoscere e saper usare le variabili locali e globali. (Obiettivo minimo)
Organizzare i dati nei vettori. (Obiettivo minimo)
Applicare gli algoritmi di ricerca. (Obiettivo minimo)
Applicare gli algoritmi di ordinamento.
Applicare gli algoritmi di fusione e totalizzazione.
Scrivere programmi in grado di trovare massimi, minimi e medie di righe e colonne di una matrice.

LABORATORIO: Si prevede l'uso di un laboratorio di informatica per attività di problem solving inerenti:

- ◆ la progettazione di algoritmi rappresentati tramite i diagrammi a blocchi;
- ◆ la verifica della soluzione mediante l'uso di un software specifico (Algobuild);
- ◆ la codifica della soluzione in un linguaggio di programmazione ad alto livello.

VALUTAZIONE:

Per quel che concerne la valutazione si adotta la rubrica definita in sede di dipartimento alla quale si rimanda.