

PIANO ANNUALE DELLE ATTIVITÀ – A.S. 2023-2024

Disciplina: INFORMATICA

PERIODO DIDATTICO: SECONDO

Docente: ARMANI PAOLO

Classe: 3INS, 4INS

Co-Docente: DANIELE MARRA

STRATEGIE DIDATTICHE

Metodologie didattiche (lezioni frontali, didattica laboratoriale, problem solving,...) e strumenti utilizzati

Lezione frontale e a FAD. Didattica laboratoriale. Lezione metodologica e dialogica. Simulazioni pre-verifica.

PIANO DISCIPLINARE PER L'ORGANIZZAZIONE DELLE ATTIVITÀ DI DIDATTICA DIGITALE INTEGRATA

Partecipazione sincrona alle lezioni e alla pre-ora tramite Google Meet. Condivisione documenti ed esercizi con Google Classroom, Google Drive, repl.it e similari

MATERIALE DIDATTICO

Dispense ed esercizi a cura del docente e di altre risorse web; uso di materiale didattico multimediale di presentazione; uso della piattaforma Google Classroom di condivisione e collaborazione a distanza.

CRITERI E STRUMENTI DI VALUTAZIONE

Verifica scritta e/o pratica delle competenze acquisite al termine di ogni modulo didattico.

ORGANIZZAZIONE MODULARE (UdA) DELLA PROGRAMMAZIONE RELATIVA AL PERIODO DIDATTICO

MODULI	PERIODO	CONOSCENZE	ABILITÀ	COMPETENZE	OBIETTIVI MINIMI
<p>Modulo 1.2</p> <p>Introduzione alla programmazione</p>	<p>lun 11 set ven 03 nov</p>	<p>Algebra Booleana;</p> <p>Tipi di dati elementari;</p> <p>Tecniche per la rappresentazione degli algoritmi: diagramma a blocchi e pseudocodice;</p> <p>Principali istruzioni e costrutti: input/output, assegnamento, selezione, iterazione;</p> <p>Metodologie per la verifica della correttezza degli algoritmi (debugger);</p> <p>Metodologie top-down e bottom-up;</p> <p>Elementi del linguaggio Java: tipi di dati, operazioni di I/O, assegnazione, strutture di controllo;</p> <p>Documentazione del codice tramite commenti, nomenclatura autoesplicativa</p>	<p>Progettare algoritmi.</p> <p>Analizzare e confrontare algoritmi diversi per la soluzione dello stesso problema.</p> <p>Scegliere i tipi di dati più idonei.</p> <p>Scegliere i casi di test in modo opportuno per verificare la correttezza delle scelte effettuate</p> <p>Progettare e implementare applicazioni secondo criteri modulari e scalabili.</p> <p>Identificare le fasi di un progetto nel contesto di un modello del ciclo di sviluppo.</p>	<p>Utilizzare le strategie del pensiero razionale negli aspetti dialettici ed algoritmici per affrontare situazioni problematiche elaborando opportune soluzioni;</p> <p>Sviluppare applicazioni informatiche per reti locali o servizi a distanza;</p> <p>Progettare;</p> <p>Risolvere problemi;</p> <p>Acquisire ed interpretare l'informazione.</p>	<p>Saper scrivere un programma per risolvere semplici problemi.</p> <p>Uso delle funzioni basilari di I/O, uso appropriato delle variabili.</p> <p>Padronanza del linguaggio nell'uso delle istruzioni condizionali e dei cicli.</p>
<p>Modulo 2.2</p> <p>Vettori, matrici.</p> <p>Funzioni e procedure.</p>	<p>lun 06 nov ven 26 gen</p>	<p>Principali strutture dati e loro implementazione: array monodimensionali, array paralleli, array bidimensionali;</p>	<p>Progettare e implementare algoritmi utilizzando diverse strutture di dati.</p>	<p>Utilizzare le strategie del pensiero razionale negli aspetti dialettici ed algoritmici per affrontare situazioni problematiche</p>	<p>Saper dichiarare, inizializzare ed utilizzare uno o più vettori monodimensionali.</p>

Algoritmo di ordinamento		Algoritmi di ordinamento e ricerca negli array. Funzioni e procedure; Complessità computazionale (cenni) Metodi ricorsivi (cenni)	Analizzare e confrontare algoritmi diversi per la soluzione dello stesso problema. Scegliere l'organizzazione dei dati più idonea Progettare e implementare applicazioni secondo criteri modulari e scalabili	elaborando opportune soluzioni; Sviluppare applicazioni informatiche; Progettare; Risolvere problemi; Acquisire ed interpretare l'informazione.	Essere in grado di estrarre min/max da un vettore. Saper usare almeno un algoritmo di ordinamento
Modulo 3.2 Introduzione alla programmazione ad oggetti	lun 29 gen ven 29 mar	Programmazione a oggetti: classi, oggetti, metodi, attributi, costruttori, principio di information hiding (public, private, getter e setter) Oggetto corrente this Creazione degli oggetti, metodi costruttori, overloading dei costruttori Programmazione guidata dagli eventi e interfacce grafiche (cenni) I/O su File Metodi speciali toString(), equals()	Progettare e implementare applicazioni secondo il paradigma ad oggetti Progettare e implementare applicazioni secondo criteri modulari e scalabili Scegliere l'organizzazione dei dati più idonea Progettare e realizzare interfacce utente Analizzare e confrontare algoritmi diversi per la soluzione dello stesso problema.	Utilizzare le strategie del pensiero razionale negli aspetti dialettici ed algoritmici per affrontare situazioni problematiche elaborando opportune soluzioni; Sviluppare applicazioni informatiche Progettare; Risolvere problemi; Acquisire ed interpretare l'informazione.	Saper definire una classe con uso di attributi e metodi. Saper istanziare ed usare correttamente oggetti della classe definita. Saper definire ed usare correttamente un costruttore .
Modulo 4.2 HTML e CSS	lun 01 apr mar 11 giu	Elementi del linguaggio HTML Fogli di Stile (CSS) Strumenti e librerie per la realizzazione di siti web statici	Progettare, realizzare e gestire pagine web statiche e con interazione locale Realizzare semplici siti web Identificare le fasi di un progetto nel contesto di un modello del ciclo di sviluppo	Utilizzare le strategie del pensiero razionale negli aspetti dialettici ed algoritmici per affrontare situazioni problematiche elaborando opportune soluzioni;	Saper creare alcune pagine web statiche con uso del linguaggio HTML e dei fogli di stile CSS; saper inserire link, immagini; saper assegnare una semplice formattazione

				<p>Sviluppare semplici applicazioni web;</p> <p>Progettare;</p> <p>Risolvere problemi;</p> <p>Acquisire ed interpretare l'informazione.</p>	
<p>Modulo 5.2</p> <p>Ricorsione, ereditarietà e polimorfismo nella programmazione ad oggetti</p>	<p>lun 11 set ven 03 nov</p>	<p>Algoritmi iterativi e algoritmi ricorsivi (ripasso); Algoritmi di ordinamento e ricorsivi (ripasso);</p> <p>Programmazione a oggetti: principi fondamentali (incapsulamento, information hiding, polimorfismo, ereditarietà), interazioni fra classi (relazioni d'uso, associazioni, aggregazione, composizione), metodi e classi astratti, interfacce, classi final;</p> <p>Diagrammi UML: diagramma delle classi (strumento di analisi)</p> <p>Attributi e metodi di istanza e di classe</p> <p>Classi nello stesso file o in file separati</p> <p>Gerarchie di Package e Livelli di visibilità</p>	<p>Progettare algoritmi secondo metodologie più facili da mantenere</p> <p>Progettare algoritmi utilizzando diverse strutture di dati.</p> <p>Progettare applicazioni secondo criteri modulari e scalabili</p> <p>Scegliere l'organizzazione dei dati più idonea</p> <p>Implementare applicazioni secondo il paradigma ad oggetti</p> <p>Progettare l'architettura di un prodotto/servizio individuandone le componenti tecnologiche</p>	<p>Utilizzare le strategie del pensiero razionale negli aspetti dialettici ed algoritmici per affrontare situazioni problematiche elaborando opportune soluzioni;</p> <p>Sviluppare applicazioni informatiche;</p> <p>Progettare;</p> <p>Risolvere problemi;</p> <p>Individuare collegamenti e relazioni</p> <p>Acquisire ed interpretare l'informazione.</p>	<p>Progettare applicazioni secondo criteri modulari e scalabili;</p> <p>Implementare applicazioni secondo il paradigma ad oggetti;</p> <p>Riutilizzo delle classi.</p> <p>Padronanza della programmazione ad oggetti facendo uso anche di caratteristiche quali ereditarietà e overriding e overloading.</p>

<p>Modulo 6.2 Programmazione avanzata</p>	<p>lun 06 nov ven 26 gen</p>	<p>Strumenti per lo sviluppo del software</p> <p>Interfacce, classi astratte;</p> <p>La gestione delle eccezioni;</p> <p>Array di oggetti;</p> <p>ArrayList;</p> <p>Principali strutture dati e loro implementazione: classi e ADT pila, coda, lista; cenni ad altre strutture (alberi, grafi).</p> <p>Tipi generici.</p>	<p>Progettare e implementare algoritmi utilizzando diverse strutture di dati.</p> <p>Analizzare e confrontare algoritmi diversi per la soluzione dello stesso problema.</p> <p>Scegliere l'organizzazione dei dati più idonea</p>	<p>Utilizzare le strategie del pensiero razionale negli aspetti dialettici ed algoritmici per affrontare situazioni problematiche elaborando opportune soluzioni;</p> <p>Sviluppare applicazioni informatiche;</p> <p>Gestire progetti secondo le procedure e gli standard previsti dai sistemi aziendali di gestione della qualità e della sicurezza;</p> <p>Redigere relazioni tecniche e documentare le attività individuali e di gruppo relative a situazioni professionali;</p> <p>Progettare;</p> <p>Risolvere problemi;</p> <p>Individuare collegamenti e relazioni;</p> <p>Acquisire ed interpretare l'informazione.</p>	<p>Progettare e implementare algoritmi utilizzando diverse strutture di dati.</p> <p>Scegliere l'organizzazione dei dati più idonea.</p> <p>Saper far uso di strutture dati come pile e liste.</p>
<p>Modulo 7.2 HTML, CSS. Introduzione a javascript</p>	<p>lun 29 gen ven 29 mar</p>	<p>Approfondimenti su HTML e CSS</p> <p>Linguaggio di programmazione lato client per la gestione locale di eventi in pagine web (Javascript)</p>	<p>Progettare, realizzare e gestire pagine web dinamiche con interazione locale</p>	<p>Sviluppare applicazioni informatiche per reti locali o servizi a distanza;</p> <p>Progettare;</p> <p>Comunicare;</p> <p>Risolvere problemi;</p>	<p>Padroneggiare i principali tag HTML (tabelle, div, span, form). Saper assegnare gli stili opportuni usando i fogli di stile CSS;</p>

				Individuare collegamenti e relazioni.	Saper usare Javascript per inserire elementi dinamici all'interno di una pagina web
Modulo 8.2 javascript e librerie per il web: jquery, bootstrap	lun 01 apr mar 11 giu	Approfondimenti su Javascript; Uso di librerie quali JQuery e bootstrap; Creazione di pagine web dinamica (lato client) e responsive.	Progettare e implementare applicazioni secondo criteri modulari e scalabili; Progettare, realizzare e gestire pagine web dinamiche con interazione locale; Identificare le fasi di un progetto nel contesto di un modello del ciclo di sviluppo.	Sviluppare applicazioni informatiche per reti locali o servizi a distanza; Progettare; Comunicare; Risolvere problemi; Individuare collegamenti e relazioni.	Saper usare librerie per formattare in modo dinamico e responsive delle pagine web. Saper usare librerie Javascript per interagire in modo dinamico con la pagina lato client.

Trento, 29 settembre 2023

IL DOCENTE

Paolo Armani

IL CO-DOCENTE

Daniele Marra