

Dipartimento di INFORMATICA

Piano di lavoro funzionale alla redazione del piano di lavoro preventivo individuale del docente

Anno Scolastico 2018/2019

Indirizzo: Informatica e Telecomunicazioni

Articolazione: Informatica

Disciplina: Informatica

Secondo biennio: Terzo anno

Competenze disciplinari di riferimento

La disciplina INFORMATICA concorre a far conseguire allo studente al termine del percorso quinquennale i seguenti risultati di apprendimento relativi al profilo educativo, culturale e professionale dello studente: utilizzare, in contesti di ricerca applicata, procedure e tecniche per trovare soluzioni innovative e migliorative, in relazione ai campi di propria competenza; cogliere l'importanza dell'orientamento al risultato, del lavoro per obiettivi e della necessità di assumere responsabilità nel rispetto dell'etica e della deontologia professionale; orientarsi nella normativa che disciplina i processi produttivi del settore di riferimento, con particolare attenzione sia alla sicurezza sui luoghi di vita e di lavoro sia alla tutela dell'ambiente e del territorio; intervenire nelle diverse fasi e livelli del processo produttivo, dall'ideazione alla realizzazione del prodotto, per la parte di propria competenza, utilizzando gli strumenti di progettazione, documentazione e controllo; riconoscere gli aspetti di efficacia, efficienza e qualità nella propria attività lavorativa; utilizzare modelli appropriati per investigare su fenomeni e interpretare dati sperimentali; utilizzare le reti e gli strumenti informatici nelle attività di studio, ricerca ed approfondimento disciplinare.

Il percorso formativo si prefigge l'obiettivo di far sviluppare le seguenti competenze, declinate in termini di conoscenze e abilità, facendo riferimento alle linee guida ministeriali.

COMPETENZE

Utilizzare le strategie del pensiero razionale negli aspetti dialettici ed algoritmici per affrontare situazioni problematiche elaborando opportune soluzioni;

Sviluppare applicazioni informatiche per reti locali o servizi a distanza;

Scegliere dispositivi e strumenti in base alle loro caratteristiche funzionali;

Gestire progetti secondo le procedure e gli standard previsti dai sistemi aziendali di gestione della qualità e della sicurezza;

Redigere relazioni tecniche e documentare le attività individuali e di gruppo relative a situazioni professionali.

Utilizzare a livello avanzato la lingua inglese per interloquire in un ambito professionale caratterizzato da forte internazionalizzazione.

Conoscenze	Abilità	Competenze specifiche
Definizione (informale) di: Problema Algoritmo Programma Progettazione top-down e <i>metodologia dei raffinamenti successivi (step-wise refinement)</i>	Analisi, progettazione e soluzione di problemi di difficoltà crescente.	<i>Sviluppare applicazioni informatiche secondo i principi della programmazione strutturata e modulare.</i> <i>Sviluppare applicazioni informatiche utilizzando ambienti di sviluppo integrato</i>
<i>Linguaggi di progetto (relativamente rigorosi)</i> Linguaggi di programmazione	Codificare algoritmi	
Tipi di dato primitivi (semplici), operatori e regole di precedenza (priorità) e associatività.	Individuare utilizzare il tipo di dato più idoneo.	
<i>Strumenti per descrivere la sintassi di un linguaggio di programmazione:</i> Diagrammi Sintattici EBNF (Extended, Backus-Naur Formalism) e varianti	<i>Interpretare descrizioni sintattiche diverse.</i> <i>Trasformare / Tradurre da una rappresentazione all'altra le categorie sintattiche più semplici</i>	
Definizione informale di: Espressione numerica Espressione booleana <i>Precedenza e associatività degli operatori.</i> <i>Parentesi per 'forzare' associatività e precedenza degli operatori.</i>	Riconoscere correttezza sintattica di espressioni. <i>Valutare correttamente espressioni.</i> <i>Scrivere espressioni corrette.</i> <i>Trasformare espressioni nelle eventuali forme equivalenti.</i>	
Elementi di logica, algebra di Boole Tipo di dato booleano <i>Teoremi di De Morgan</i>	Costruire tabelle di verità. <i>Formalizzare condizioni complesse.</i> <i>Applicare i teoremi di De Morgan per riscrivere le espressioni booleane</i>	
<i>Interprete e compilatore analogie e differenze</i> <i>Compilazione</i> <i>Interpretazione</i> <i>Esecuzione</i> <i>Debugging</i> <i>IDE (Integrated Development Environment)</i>	Conoscere le differenze e i vantaggi / svantaggi di un linguaggio interpretato o compilati Utilizzare con sicurezza l'ambiente di sviluppo in tutte le fasi di sviluppo di un programma: Editing, Compilazione, Debugging, Esecuzione.	

<p>Strutture di controllo Sequenza Selezione binaria</p> <p><i>Iterazione determinata (for)</i> <i>Iterazione indeterminata a controllo iniziale (while)</i> <i>Iterazione indeterminata a controllo finale (do/while)</i> Selezione multipla</p>	<p>Individuare la strutture di controllo necessaria a risolvere un problema</p> <p>Individuare il costrutto iterativo più adatto alla natura del problema</p> <p>Utilizzare il costrutto selezione multipla nei casi in cui sia applicabile</p>	
<p>Teorema di Boehm-Jacopini</p>	<p>Saper trasformare algoritmi (anche informali) in algoritmi equivalenti che utilizzino i tre costrutti: Sequenza Selezione Iterazione</p>	
<p>Parametro formale Parametro attuale (effettivo) Definizione di un sottoprogramma (Intestazione e corpo)</p> <p>Regole di visibilità (scope)</p> <p><i>Meccanismi di passaggio dei parametri</i></p> <p>Chiamata di un sottoprogramma</p>	<p>Codificare algoritmi con sottoprogrammi opportunamente parametrizzati</p>	
<p><i>Paradigma funzionale.</i> <i>Ricorsione per:</i> <i>definire (insiemi, strutture, ...)</i> <i>risolvere problemi</i> Sottoprogrammi ricorsivi</p>	<p>Codificare algoritmi ricorsivi con sottoprogrammi.</p>	

Conoscenze	Abilità	Competenze specifiche
Dichiarazione e creazione di oggetti di classe String Metodi per la gestione di stringhe Stringhe modificabili (StringBuffer)	Creare e manipolare stringhe.	Sviluppare applicazioni informatiche per la manipolazione di stringhe.
Algoritmi fondamentali sulle stringhe (Appartenenza di un carattere a una stringa, appartenenza di una sotto-stringa ad una stringa, verificare se una stringa è palindroma...)	Utilizzare algoritmi fondamentali sulle stringhe	Utilizzare diversi formalismi per esprimere procedure di calcolo o rappresentare informazione strutturata.
Pattern - Matching	Riconoscimento di pattern	
Espressioni regolari Meta caratteri principali	Interpretare espressioni regolari di complessità crescente (Identificatore, CAP, Codice Fiscale, email...) Costruire espressioni regolari Utilizzo di strumenti software per definire espressioni regolari www.regex101.com	
Concetto di Automa a Stati Finiti Deterministico (ASFD) Rappresentazioni di automi tramite grafi e matrici di transizione Automi Riconoscitori	Disegnare e interpretare il grafo di un automa. Interpretare la matrice di transizione i un automa Costruire automi per riconoscere semplici categorie sintattiche	
Scrivere semplici programmi (costruire semplici macchine di Turing) per riconoscere stringhe, contare occorrenze ...	La macchina di Turing	

Conoscenze	Abilità	Competenze specifiche
<p>Concetto di array: sequenza di valori ad accesso diretto Array come parametri di metodi e come valori di ritorno.</p> <p>Algoritmi fondamentali di ricerca (ingenua, con sentinella, dicotomica)</p> <p>Algoritmi fondamentali di ordinamento (bubble sort, selection sort...)</p> <p>Array bidimensionali (matrici)</p>	<p>Dichiarare array di vario tipo Elaborare array</p> <p>Ricerca elementi in un array</p> <p>Ordinare un array</p> <p>Dichiarare ed elaborare matrici.</p>	<p>Progettare e implementare applicazioni che gestiscono dati organizzati in collezioni omogenee.</p>
<p>Concetto di lista: sequenze di oggetti con accesso diretto o sequenziale</p> <p>Dizionari (mapping) come coppie (Key, Value)</p>	<p>Elaborazione di liste</p> <p>Elaborazione di dizionari</p>	

Conoscenze	Abilità	Competenze specifiche
<p>Componenti di una GUI (oggetti componenti e oggetti contenitori). Proprietà dei componenti grafici. Disposizione degli oggetti all'interno dei contenitori secondo diversi layout.</p> <p>Concetto di evento Generatori di eventi e ascoltatori (gestori di eventi)</p>	<p>Utilizzare componenti grafici per la Costruzione di un'interfaccia per un'applicazione</p> <p>Costruzione di applicazioni con interfaccia grafica con gestione degli eventi anche provenienti da componenti diversi.</p>	<p>Sviluppare applicazioni informatiche che interagiscono con l'utente tramite un'interfaccia grafica.</p>

Conoscenze	Abilità	Competenze specifiche
<p>Definizione di ipertesto. Principali tag HTML per impaginare il testo. <i>Elenchi puntati e numerati, tabelle, immagini.</i> Riferimenti assoluti e relativi, concetto di URL, link ed etichette. Fogli di stile CSS (inline, incorporati ed esterni) per la definizione del layout. Pagine web, selettori (di classe e id), proprietà e valori di proprietà. Collegamento di un file HTML ad un foglio di stile. <i>Tag DIV per strutturare una pagina web in sezioni distinte e disposizione degli elementi html con CSS.</i></p>	<p>Progettare e realizzare documenti ipertestuali</p>	<p>Realizzare pagine/siti web.</p>

Strumenti didattici:

Libro di testo "INFORMATICA PER ISTITUTI TECNOLOGICI" Vol. B A. Lorenzi, E. Rizzi – Edizioni Atlas e materiale scaricato da web (tutorial, presentazioni, video lezioni, documentazione tecnica...) o fornito dal docente, lavagna tradizionale, LIM, attrezzatura hardware e software di laboratorio, rete LAN Marconi, piattaforme di apprendimento on-line (Campus Marconi), corsi on-line, Aula Polifunzionale A042.

Metodologie didattiche:

- Lezione frontale e/o partecipata
- Esercitazioni guidate dal docente teoriche e pratiche (problem solving)
- Esercitazioni individuali e/o in gruppi (team working), teoriche e pratiche (learning by doing, cooperative learning).
- Stesura di relazioni sulle esperienze effettuate
- Interventi di consolidamento e/o approfondimento
- Uscite didattiche

Modalità di recupero/sostegno:

in itinere mediante interventi del docente durante le ore di laboratorio, in modalità alla pari (peer to peer) mediante lavoro di gruppo dove gli alunni più capaci e con buone doti di comunicazione supportano in compagni in difficoltà.

Numero e tipologie di prove di verifica:

- Una prova sommativa per ogni argomento (con eventuale prova di recupero) costituita da test ed esercizi per la verifica di conoscenze e abilità e da problemi per la verifica di abilità e competenze.
- Verifiche formative in itinere:
 - Test: domande a risposta multipla, a completamento, abbinamento (anche online)
 - Interrogazione orale (anche in forma scritta), brevi interrogazioni dal posto
 - Presentazione (multimediale)

- Valutazione delle attività di laboratorio, individuali e di gruppo e delle relazioni relative a tali esperienze.

Criteri di valutazione verifiche:

Scritte:

- Linguaggio appropriato
- Conoscenza dei concetti e delle tecniche
- Correttezza dell'esercizio
- Completezza dell'esercizio
- Leggibilità dell'elaborato
- Capacità di utilizzare le conoscenze in situazioni nuove

Orali:

- Linguaggio appropriato
- Conoscenza dei concetti
- Correttezza dell'esposizione
- Completezza dell'esposizione
- Capacità di trovare soluzioni equivalenti
- Capacità di applicare le conoscenze a situazioni nuove
- Capacità di trovare esemplificazioni
- Capacità di correlare argomenti

Progetti e/o AreaLab:

- Correttezza e completezza delle singole fasi
- Capacità di argomentare le soluzioni trovate e le ipotesi di lavoro
- Capacità di confrontare soluzioni alternative
- Leggibilità e completezza della documentazione
- Organizzazione complessiva del progetto

Pratiche:

- Capacità di utilizzare strumenti informatici
- Conoscenza dei procedimenti
- Capacità di realizzare un prodotto funzionante e ben documentato

Individuazione degli obiettivi minimi:

Gli obiettivi minimi sono stati evidenziati in grassetto nella tabella relativa alle competenze.

Il Dirigente Scolastico

Dott.ssa Gabriella Piccoli