



Ministero dell'Istruzione dell'Università e della Ricerca

ISTITUTO TECNICO COMMERCIALE STATALE "ABBA - BALLINI"

Via Tirandi n. 3 - 25128 BRESCIA – www.abba-ballini.gov.it

tel. 030/307332-393363 - fax 030/303379

bstd150001@pec.istruzione.it : info@abba-ballini.it: bstd150001@istruzione.it



PIANO DI LAVORO DEL DOCENTE

I.T.C. "Abba – Ballini" – Brescia

Anno scolastico 2017/2018

Docente: Giuliana Pederzoli	Classe: 3° A Indirizzo: SIA	Disciplina: INFORMATICA	Ore di lezione settimanali : 4
---	---	---------------------------------------	---------------------------------------

Situazione della classe : Omissis

Risultati di apprendimento da raggiungere : si fa riferimento al piano di lavoro del dipartimento

Competenze : si fa riferimento al piano di lavoro del dipartimento

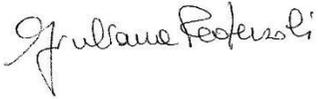
Abilità: si fa riferimento al piano di lavoro del dipartimento

Conoscenze: si fa riferimento al piano di lavoro del dipartimento

Contenuti: si fa riferimento alla tabella allegata

Metodi : si fa riferimento al piano di lavoro del dipartimento

Tempi: si fa riferimento alla tabella allegata	Verifiche e valutazioni: si fa riferimento al piano di lavoro del dipartimento
Strumenti : si fa riferimento al piano di lavoro del dipartimento	
Eventuali recuperi: tempi, saperi essenziali,metodi: si fa riferimento al piano di lavoro del dipartimento	

Firmato dal docente	Visto dal Dirigente Scolastico
	

Data di presentazione: 31 ottobre 2017

INFORMATICA

Piano di lavoro per la classe 3° A Sistemi Informativi Aziendali

Moduli previsti

Modulo	Unità	Conoscenze da acquisire	Competenze da acquisire	Abilità da acquisire	Verifiche previste
Metodo di studio	<i>Miglioramento capacità di comunicazione</i>	<ul style="list-style-type: none"> • Conoscenza metodi di studio (mediante esercizi mirati e con argomenti comuni ad altre discipline) • analisi dei linguaggi specifici : analisi della guida in linea del linguaggio di programmazione (modalità di utilizzo, sintassi delle istruzioni, utilizzo degli esempi) • analisi di testi : analisi testo in uso (analisi obiettivi e competenze, ricerca dei concetti chiave di ogni unità didattica, utilizzo delle domande di verifica per il controllo delle conoscenze e degli obiettivi minimi) 	<ul style="list-style-type: none"> - metodo di studio efficace - capacità di analisi dei testi - capacità di analisi dei linguaggi specifici 	<ul style="list-style-type: none"> - organizzare lo studio - pianificare il lavoro - realizzare mappe concettuali - cercare le informazioni corrette - utilizzo corretto dei manuali 	
Il sistema di elaborazione	1. Architettura hardware (approfondimento dei concetti introdotti nel biennio)	<ul style="list-style-type: none"> • Struttura del sistema di elaborazione • Architettura del computer • Processore, memorie e dispositivi 	<ul style="list-style-type: none"> - spiegare in termini funzionali l'architettura di un elaboratore 	<ul style="list-style-type: none"> - saper individuare le unità che compongono un s.di elaborazione - saper riconoscere i diversi supporti per la memorizzazione delle informazioni 	Test di conoscenza
	2. Software	<ul style="list-style-type: none"> • Sistema come combinazione di hw, firmware, sistema operativo, sw applicativo e system configuration data • Software applicativo e software di sistema 	<ul style="list-style-type: none"> - classificare i diversi SW utilizzati da un computer - spiegare in termini funzionali i moduli di un s.o. 	<ul style="list-style-type: none"> - riconoscere le funzioni fondamentali di un s.o. 	Test di conoscenza

		<ul style="list-style-type: none"> • Software open source, proprietario, freeware 			
	3. Il sistema operativo Windows.		- utilizzare le funzioni del s.o. attraverso l'interfaccia grafica	- saper organizzare il lavoro per directory - saper usare la guida in linea - saper usare gli accessori - saper condividere le risorse in rete *	- prova pratica al PC sulle funzioni fondamentali del s.o. in uso - test di conoscenza
Soluzione dei problemi: dall' algoritmo al programma	1. Dal problema all'algoritmo	<ul style="list-style-type: none"> • Analisi del problema • Formalismi per rappresentare algoritmi • Flow chart e pseudocodifica 	- saper distinguere tra variabili e costanti, tra dati e azioni - riconoscere le caratteristiche fondamentali delle istruzioni di un algoritmo - rappresentare con F.C. e pseudocodifica algoritmi di base	- individuazione di dati di I/O, variabili di lavoro, sequenze di lavoro -rappresentazione tramite F.C. e pseudoc.	- test sui concetti di base - esercizi di applicazione
	2. Tecniche di progettazione sw	<ul style="list-style-type: none"> • Differenti paradigmi di programmazione (strutturata, object oriented) • Approccio top down e bottom up • Astrazione dei dati • Problematiche nell'integrazione con i sistemi legacy 	- affrontare i problemi scomponendoli per parti -distinguere i vari tipi di dati -organizzare i dati in strutture -associare ad ogni situazione l'adatta struttura di dati	- utilizzare il metodo top down - organizzare i prog. per sottoprogr.	- prove di costruzione di prog. organizzati a menu
	3. Generalità sui linguaggi di programmazione	<ul style="list-style-type: none"> • Generazioni di linguaggi: funzionali, procedurali, OO, scripting • Sintassi- semantica • Compilatori, interpreti e assembleri 			Test di conoscenza
	4. Linguaggio di programmazione strutturato	<ul style="list-style-type: none"> • costrutti di base di una programmazione strutturata • strutture dati • scomposizione funzionale e passaggio dei parametri • regole per un codice ben strutturato e ben commentato 	-applicare correttamente i principi della programm. strutturata - codificare e validare algoritmi effettuando le necessarie correzioni	- individuare le strutture di controllo necessarie alla soluzione di un problema -saper organizzare i dati in array -utilizzare algoritmi di ricerca e ordinamento di array	- prove di costruzione di algoritmi

	5. Testing e documentazione di supporto	<ul style="list-style-type: none"> • concetti di base del testing • metodologie di test statiche e dinamiche • cambiamenti nel sw e impatti nella documentazione • user manual e technical document reference 	- codificare e validare algoritmi effettuando le necessarie correzioni	- fare il tracing per verificare la correttezza di un algoritmo	- Prove in laboratorio - Sviluppo documentazione
Interfaccia utente e progettazione ipertestuale	Linee guida per interfaccia utente	<ul style="list-style-type: none"> • Interazione uomo macchina: concetti base della teoria della comunicazione • Concetto di user interface e diversi tipi di interfaccia utente • Metodologie per comunicare informazioni 	Analisi e valutazione di interfacce in base all'utilizzo	Realizzazione di interfacce	
	Concetti base di grafica	Formati multimediali Immagini bitmap e vettoriali Formati audio e video Principi di base per l'uso di colori, contrasto e animazioni	Utilizzo dei formati corretti a seconda dei contesti	Elaborazione e trasformazione delle immagini	Prova pratica al computer
	Internet e www	Cenni storici Intranet e extranet Website per un'azienda	- conoscenza delle principali problematiche e strumenti relativi alle reti di computer	-capacità di utilizzo dei principali strumenti per la navigazione e la ricerca delle informazioni, nonché per l'utilizzo della posta elettronica	-test di conoscenza - prove pratiche al computer
	Progettazione web	Bisogni dell'utente Uso dei messaggi e dei colori Leggibilità, facile navigabilità Structure diagram, story board Strumenti per lo sviluppo			- Sviluppo di un sito web con lavoro di gruppo
	Creazione di pagine ipertestuali	Linguaggio di markup Costrutti base html Fogli di stile Cenni su linguaggi client side e server side			- Test di conoscenza - Sviluppo di pagine html