



ISO 9001: 2015 Cert. N° 50 100 14484 – Rev.002
Validità 14.09.2018 – 11.03.2021

ISTITUTO DI ISTRUZIONE SUPERIORE “BUCCARI – MARCONI”

Sede Buccari: Viale Colombo 60 – 09125 Cagliari - Uff. Presidenza / Segreteria ☎ 070300303 – 0703011793

Sede Marconi: Via Pisano, 7 Cagliari ☎ 070554758

Codice Fiscale: 92200270921 – Codice Univoco: UFAXY4 - Codice Meccanografico: CAIS02300D

Rev. N.02 del 20.12.2018

SCHEDA PROGRAMMA SVOLTO

Anno scolastico 2022/2023

Classe: TERZA sez. IS Indirizzo INFORMATICA (NT/ITIA)

DISCIPLINA: Informatica

Libro di testo: CORSO DI INFORMATICA 2ED. - PERINFORMATICA. VOLUME 1 (LD) ALGORITMI ELINGUAGGIO C/C++.
PAGINE WEB CON HTML, CSS E JAVASCRIPT
FORMICHI FIORENZO MEINI GIORGIO VENUTI IVAN
ZANICHELLI EDITORE

Altri strumenti o sussidi: Appunti realizzati dai docenti e contenuti multimediali reperibili in rete, utilizzo della classe virtuale Classroom.

Nota del docente: Totale ore svolte 160



Trasporti e Logistica – Conduzione del mezzo navale
Trasporti e Logistica – Conduzione di apparati e impianti marittimi
Trasporti e Logistica – Logistica
Elettronica ed Elettrotecnica - Informatica e Telecomunicazioni (Diurno e Serale)

Viale Colombo, 60 - Cagliari
Viale Colombo, 60 - Cagliari
Viale Colombo, 60 - Cagliari
Via Pisano, 7 - Cagliari

Contenuti delle lezioni, delle unità didattiche o dei moduli svolti/MODULO	Eventuali discipline coinvolte	Attività (extrascolastiche o integrative) coerenti con lo svolgimento del programma	Conoscenze, abilità e competenze	Criterio di sufficienza delle abilità, conoscenze e delle competenze da acquisire	Tipologie delle prove utilizzati per la valutazione	Ore per modulo
Modulo n° 1 Linguaggi e macchine						10
- Le informazioni, i linguaggi, i sistemi; - I linguaggi informatici; - I paradigmi di programmazione;	Sistemi e reti, TPSIT		Conoscenze: Consolidare le conoscenze fondamentali dell'informatica, cogliere l'aspetto sistemico di una macchina per il calcolo. Abilità: Paradigmi di programmazione Competenze: Identificare un linguaggio di programmazione	Essere in grado di riconoscere le principali tipologie di linguaggi e sapere come il calcolatore gestisce l'informazione	Verifica scritta teorica.	
Modulo n° 2 Progettazione degli Algoritmi						35

• Generare il modello del problema; • Gli algoritmi; • Modalità di rappresentazione di un algoritmo. • Risolvere problemi mediante algoritmi e rappresentarli con	TPSIT		Competenze: Individuare fasi di passaggio dal problema alla soluzione, capire il concetto di Algoritmo, saper utilizzare i formalismi che lo compongono. Conoscenze: Definizione e proprietà degli algoritmi. Formalismi per la costruzione, i tre principali elementi di un algoritmo, Teorema di Böhm-Jacopini, tabella traccia. Abilità: distinguere tra variabili e costanti, dati e azioni. Utilizzare le principali strutture per costruire un algoritmo, realizzare la tabella traccia.	Essere in grado di realizzare un algoritmo risolutivo di qualsiasi problema, rappresentarlo secondo diversi linguaggi formali utilizzando le principali strutture. Saper distinguere una variabile da una costante. Costruire la tabella traccia.	Verifica scritta teorica e di laboratorio	
Modulo n° 3 Codifica nel linguaggio C++						45

• Le basi del linguaggio; • Conversione dal linguaggio formale degli algoritmi al linguaggio C++ • Programmazione di tipo sequenziale. • Dichiarazione di variabili e costanti; • Frasi di commento; • Assegnazione dei valori; • Lettura e scrittura di valori dallo standard Input e Output; • Operatori relazionali e logici; • Tipi di dati; • La sequenza; • Struttura alternativa; • Ripetizione pre e post condizionale; • La ripetizione con contatore; • Strutture a scelta multipla.			Competenze: Costruire programmi eseguibili dal computer e controllarne l'esecuzione. Conoscenze: Struttura di un programma, tipi di dati, istruzioni e operatori, istruzioni di Input Output. Codifica delle strutture di controllo. Abilità: Utilizzare un ambiente di sviluppo (IDE) per la programmazione in linguaggio C++, scrivere in maniera corretta un linguaggio utilizzando la sintassi appropriata, riconoscere le diverse tipologie di errore. Documentare un programma.	Essere in grado di realizzare un programma in linguaggio C++ utilizzando un IDE specifico. Saperlo compilare e eseguire, identificando eventuali errori e correggendoli.	Verifica teorica e di laboratorio con esecuzione di un progetto.	
Contenuti delle lezioni, delle unità didattiche o dei moduli svolti/MODULO	Eventuali discipline coinvolte	Attività (extrascolastiche o integrative) coerenti con lo svolgimento del programma	Conoscenze, abilità e competenze	Criterio di sufficienza delle abilità, conoscenze e delle competenze da acquisire	Tipologie delle prove utilizzati per la valutazione	Ore per modulo
Modulo n° 4 Programmazione Top/Down						25

• Le funzioni; • Firma di una funzione; • Le funzioni con parametri; • Passaggio di parametri; • Dichiarazione delle funzioni con prototipi; • Suddivisione del programma in più file.			Competenze: Controllare la complessità degli algoritmi organizzando il programma in moduli. Conoscenze: Sviluppo TopDown, organizzare il programma in funzioni, definire i prototipi, individuare parametri di IO, regole di visibilità. Abilità: Scomporre il programma in funzioni, riutilizzare più volte le funzioni assegnando diversi valori ai parametri, distinguere tra variabili locali e variabili globali, suddividere il programma in più file e richiamarli tutti all'interno del programma principale.	Essere in grado di scomporre un problema in sotto problemi più semplici e risolvere ognuno di questi mediante una funzione. Riutilizzare la medesima funzione in differenti problemi, scomporre il programma in più file e generare librerie di funzioni.	Verifica scritta e orale di teoria, verifica scritta di laboratorio.	
Modulo n° 5 Strutture dati						30

• Gli array; • Le strutture;			Competenze: Organizzare i dati in strutture e implementare la loro gestione. Conoscenze: Strutture dati: Array, matrici, strutture. Abilità: Definire strutture per dati dello stesso tipo o di tipo diverso, scegliere la struttura dati più idonea a ogni problema.	Essere in grado di identificare la struttura dati più idonea al problema, riuscire a popolarla, valorizzando i singoli campi, individuare all'interno eguaglianze e differenze, riordinare gli elementi di una struttura dati.	Verifica scritta e orale di teoria, verifica scritta di laboratorio.	
Modulo n° 6 Interfacce Grafiche						15

• Installare e utilizzare un IDE per la realizzazione di interfacce grafiche: QT Library: • Principali elementi grafici; • Eventi e interazioni.			Competenze: Creare interfacce grafiche dinamiche per l'interazione tra software e programma. Conoscenze: Installare e utilizzare le QT Library, conoscere i principali elementi di una interfaccia grafica. Saper gestire gli eventi e interagire con il programma. Abilità: Installare in autonomia le librerie QT, creare un nuovo programma, definire e posizionare gli elementi principali in funzione degli input/output richiesti, gestire l'interazione con il software sottostante.	Essere in grado di installare il software delle librerie QT, creare una interfaccia grafica in funzione degli input e output richiesti dal programma, gestire gli eventi.	Verifica scritta e orale di teoria, verifica scritta di laboratorio.	
--	--	--	--	--	---	--

Cagliari, 15/06/2023

I Docenti:

Prof. Giuseppe Ancora

Prof. Fabrizio Passiu