

Disciplina Informatica 3°

A.S. 2016/17

Finalità

L'insegnamento di Informatica si propone di:

- Affrontare la soluzione di un problema di complessità adeguata
- Scegliere le metodologie e gli strumenti più idonei alla soluzione di un problema
- Affrontare con autonomia l'evoluzione delle tecnologie informatiche
- Fornire le conoscenze e le abilità necessarie per l'uso di un sistema di elaborazione
- Sviluppare le capacità di analisi e sintesi

Obiettivi di apprendimento

Lo studente alla fine del triennio deve essere in grado di:

1. Risolvere problemi, indipendentemente dal linguaggio di programmazione
2. Verificare la correttezza di una soluzione
3. Usare il linguaggio di comando di un sistema operativo
4. Usare un linguaggio imperativo
5. Leggere e interpretare un programma
6. Organizzare i dati
7. Documentare software.

N°	Modulo	Obiettivi	Tempi	Contenuti	Periodo
1	Concetti di base e ambiente di lavoro e	3, 4	12 h	Terminologia Lo schema di Von Neumann Il ruolo dei S. O.	Settembre
2	Dal problema all' algoritmo	1, 2, 4, 6, 7	16 h	Metodologia di lavoro Comprensione del problema Ricerca della soluzione Dati e risultati Azioni, processi e procedure Algoritmi e programmi Costanti e variabili	Settembre Ottobre
3	Strumenti per la stesura di un algoritmo	2, 4, 5, 6, 7	94 h	Rappresentazione di un algoritmo mediante diagrammi di flusso Istruzioni semplici: assegnazione, input/output, condizionali Costrutti: sequenza, alternativa semplice e nidificata, Iterazioni precondizionali e poscondizionali Contatori e sommatore Funzioni	Ottobre Novembre Dicembre Gennaio
4	Introduzione ai linguaggi	3, 4, 5	14 h	Caratteristiche dei linguaggi di programmazione a basso e a alto livello Compilatori e interpreti Dichiarazione delle variabili Istruzioni	Da Ottobre a Maggio
5	Le strutture dati	2, 4, 5, 6, 7	60 h	Array monodimensionali: caratteristiche, dichiarazione, rappresentazione in memoria Algoritmi di ricerca Algoritmi di ordinamento Stringhe	Gennaio-Febbraio Marzo Aprile- Maggio

MODULO N. 1 : Concetti di base e ambiente di lavoro

PREREQUISITI PER L'ACCESSO AL MODULO: Non sono necessari prerequisiti specifici.

U.D	Titolo	Contenuti	Conoscenze teoriche	Abilità pratiche	Ore
1	Concetti di base	1. L'evoluzione dell'informatica. 2. Le risorse tecnologiche e le applicazioni. 3. I termini di uso comune	Conoscere la terminologia di base dell'informatica	Saper utilizzare la terminologia di base dell'informatica	6
2	Ambiente di lavoro	1. Concetto intuitivo di dispositivo automatico. 2. Struttura generale del sistema di elaborazione (modello di Von Neumann) e sue componenti.	Conoscere in termini funzionali l'architettura di un sistema di elaborazione	Saper individuare le unità che compongono un sistema di elaborazione. Saper riconoscere i diversi supporti di memorizzazione delle informazioni.	6
Attività di laboratorio					
Familiarizzazione con il laboratorio ed il PC					
Verifica sommativa di fine modulo					
Questionario a risposte chiuse, vero/falso					
Recupero					
Curricolare					
Verifica sommativa di recupero					
Questionario a risposte chiuse, vero/falso					

Esempio questionario

L'affermazione che la RAM è una memoria volatile significa che:

- A. I dati vengono persi quando si chiude una applicazione
- B. I dati vengono persi quando si spegne l'elaboratore
- C. I dati vengono persi quando si hanno guasti all'hardware
- D. I dati vengono persi nel caso di errori software

Le componenti dell'Unità Centrale di Elaborazione sono:

- A. RAM e ROM
- B. ALU e RAM
- C. ALU e ROM
- D. ALU e Unità di controllo

Il software usato nei computer può essere classificato in:

- A. Software di base e software applicativo
- B. Software centrale e software periferico
- C. Software volatile e software non volatile
- D. Software primario e software secondario

Il firewall è un sistema:

- A. Antincendio
- B. Per la protezione dei dati
- C. Per garantire l'alimentazione elettrica del computer
- D. Per effettuare il backup automatico dei dati

Quale di queste componenti non influenza le prestazioni di un computer?

- A. Potenza della CPU
- B. Dimensione della memoria
- C. Utilizzo contemporaneo di più programmi
- D. Dimensioni dello schermo video

MODULO N. 2 : Dal problema all' algoritmo

PREREQUISITI PER L'ACCESSO AL MODULO: Non sono necessari prerequisiti specifici. L'alunno dovrebbe possedere intuito, fantasia, curiosità e spirito critico non disgiunti da una certa predisposizione per l'area logico-matematica.

U.D	Titolo	Contenuti	Conoscenze teoriche	Abilità pratiche	Ore
1	Dal problema all'algoritmo	1. Un approccio sistematico ai problemi. 2. La comprensione del problema. 3. La modellizzazione della soluzione. 4. La ricerca della soluzione. 5. Definizione di problema, dati e risultati. 6. Azioni elementari, processi, procedure, esecutori. 7. La metodologia di lavoro. 8. Definizione di algoritmo. 9. Requisiti di un buon algoritmo. 10. Algoritmi e programmi. 11. Costanti, variabili, espressioni e loro valutazione.	Conoscere la differenza fra costante e variabile e fra dato e azione Conoscere la differenza tra istruzione e azione, algoritmo e processo, programmatore e processore	Saper distinguere all'interno di un problema: - tra variabili e costanti - tra dati e azioni Saper distinguere tra: - istruzione e azione - algoritmo e processo - programmatore e processore	16
Attività di laboratorio					
Analisi di semplici problemi: individuazione delle costanti, delle variabili e del procedimento risolutivo					
Verifica sommativa di fine modulo					
Questionario a risposta aperta, Attività di laboratorio					
Recupero					
Curricolare e/o classi aperte					
Verifica sommativa di recupero					
Questionario a risposta aperta					

Esempi problemi (da risolvere divisi in squadre)

- Sono nell'isola di Chec , dove vivono solo cavalieri e furfanti. I cavalieri dicono sempre la verit ; i furfanti mentono sempre.

Incontro cinque abitanti dell'isola: Aristide, Basilio, Carlo, Donato ed Evasio.

Aristide afferma: "Carlo   un cavaliere"

Basilio afferma: "Evasio   un furfante"

Carlo afferma: "Basilio   un furfante"

Donato afferma: "Aristide   un cavaliere"

Evasio afferma: "Carlo e Donato sono di diversa natura"

Chi sono i cavalieri e chi i furfanti?
- Un beduino in punto di morte chiama a s  i suoi tre figli e dice loro che lascer  i suoi 11 cammelli in eredit  secondo le seguenti proporzioni:

il primo figlio potr  prendere la met  dei cammelli;

il secondo figlio potr  prendere un quarto dei cammelli;

il terzo figlio potr  prenderne un sesto.

Quando il beduino muore, i suoi eredi si domandano come sia possibile eseguire queste eccentriche istruzioni: i cammelli valgono ben poco quando vengono divisi in parti!

Come si pu  risolvere il problema senza ammazzare nessun cammello?

MODULO N. 3 : Strumenti per la stesura di un algoritmo

PREREQUISITI PER L'ACCESSO AL MODULO: Conoscenze, competenze e capacità acquisite nel Modulo 2.

U.D	Titolo	Contenuti	Conoscenze teoriche	Abilità pratiche	Ore
1	Strumenti per la stesura di un algoritmo	1. Il linguaggio della pseudocodifica. 2. La dichiarazione delle variabili. 3. Tipi di dato semplici (int, float, double, char). 4. L'istruzione di assegnazione. 5. Le istruzioni di input/output. 6. Il controllo delle azioni.	Conoscere le differenze fra le istruzioni principali Conoscere la sequenza di istruzioni per risolvere semplici problemi Conoscere la differenza tra i diversi tipi di dato	Saper analizzare problemi non complessi Saper risolvere problemi non complessi Saper formalizzare la soluzione con l'uso della pseudo codifica Saper scegliere il tipo di dato più idoneo per una variabile	24
		Attività di laboratorio Risoluzione di semplici problemi			
2	Le strutture di controllo	1. La programmazione strutturata. 2. Regole generali per la stesura degli algoritmi. 3. La sequenza. 4. La struttura alternativa semplice e nidificata. 5. La selezione multipla 6. Le iterazioni. 7. Costrutto WHILE 8. Costrutto DO.....WHILE 9. Costrutto FOR 10. Confronto fra le strutture di ripetizione. 11. Variabili di tipo contatore e sommatore	Conoscere le caratteristiche della programmazione strutturata Conoscere i costrutti fondamentali della programmazione Utilizzare la pseudo codifica per formalizzare un algoritmo.	Saper analizzare e risolvere un problema Saper scrivere un algoritmo strutturato Saper individuare la necessità di una strutture alternativa semplice o nidificata Saper individuare la necessità di una iterazione Saper scegliere il tipo di iterazione più idoneo alla soluzione del problema	54
3	Le Funzioni	1. Funzioni e Procedure. 2. Regole generali per dichiarazione e definizione delle funzioni. 3. Parametri formali e attuali 4. Variabili locali e globali 5. Passaggio dei parametri.	Conoscere l'utilità della programmazione procedurale Individuare la differenza tra funzione e procedura	Saper scomporre un problema Saper scegliere il tipo di passaggio dei parametri	16
Attività di laboratorio					
Risoluzione di problemi di varia natura attraverso l'uso di algoritmi e codifiche in C++					
Verifica sommativa di fine modulo					
Compito in classe, Attività di laboratorio					
Recupero					
classi aperte					
Verifica sommativa di recupero					
Compito in classe					

MODULO N. 4 : Introduzione ai linguaggi

PREREQUISITI PER L'ACCESSO AL MODULO: Conoscenze, competenze e capacità acquisite nel Modulo 3.

U.D	Titolo	Contenuti	Conoscenze teoriche	Abilità pratiche	Ore
1	Introduzione al linguaggio	1. I linguaggi di programmazione 2. Compilatori e interpreti 3. Le fasi della compilazione	Conoscere la differenza fra compilatore e interprete Codificare un algoritmo Conoscere un ambiente di sviluppo	Saper distinguere un linguaggio compilato da un linguaggio interpretato Saper utilizzare una sintassi corretta	14
Attività di laboratorio					
L'ambiente di sviluppo. Codifica e testing di algoritmi già svolti					
Verifica sommativa di fine modulo					
Questionario, Attività di laboratorio					
Recupero					
Curricolare					
Verifica sommativa di recupero					
Questionario					

Esempio Questionario

Cosa viene stampato dal seguente frammento di programma?

```
float f = 1/5, i;
for (i=0.0; i<1.0; i=i+f ) cout<<"ok";
```

- (a) stampa 5 volte ok (b) stampa 4 volte ok (c) il compilatore segnala un errore (d) stampa infinite volte ok

Cosa stampa il seguente programma?

```
void main()
{int in,r,nd,ind=0; long int pot,nb=0;
cout<<"Inserisci un intero:"; cin>>nd;
do {pot=1;r=nd%2;
for(in=0;in!=ind;in++) pot=pot*10;
nb=nb+(r*pot); ++ind;nd=nd/2;
}while(nd!=0);
cout<<nb; }
```

MODULO N. 5 : Le strutture dati

PREREQUISITI PER L'ACCESSO AL MODULO: Conoscenze, competenze e capacità acquisite nei Moduli precedenti.

U.D	Titolo	Contenuti	Conoscenze teoriche	Abilità pratiche	Ore
1	Array monodimensionali (vettori)	1. Gli array monodimensionali: caratteristiche, dichiarazione e rappresentazione in memoria. 2. Algoritmi di ricerca: ingenua e dicotomica (binaria). 3. Algoritmi di ordinamento: ingenuo, bubble sort	Conoscere le caratteristiche di un array monodimensionale Conoscere l'utilità di un array monodimensionale	Saper organizzare i dati in un array Saper individuare e implementare l'algoritmo di ricerca più idoneo alla soluzione del problema	40
		Attività di laboratorio Risoluzioni di problemi di varia natura con uso di array monodimensionali. Risoluzioni di problemi di ricerca e ordinamento..	Conoscere le caratteristiche e le differenze fra i diversi tipi di ricerca Conoscere gli algoritmi di ricerca e di ordinamento	Saper individuare e implementare l'algoritmo di ordinamento	
2	Le stringhe	1. Le stringhe, loro dichiarazione e rappresentazione in memoria. 2. Funzioni di gestione di stringhe	Conoscere le caratteristiche di stringa	Saper associare ad ogni situazione problematica la struttura dati più idonea	20
		Attività di laboratorio Risoluzioni di problemi di varia natura con uso di stringhe	Conoscere le funzioni di string.h		
Verifica sommativa di fine modulo					
Compito in classe, Attività di laboratorio					
Recupero					
classi aperte					
Verifica sommativa di recupero					
Compito in classe					