**DIPARTIMENTO DI ELETTRONICA**

**INDIRIZZO – ARTICOLAZIONE LICEO SCIENTIFICO – OPZIONE SCIENZE APPLICATE**

**MATERIA INFORMATICA**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **PRIMO BIENNIO** |  |  | **SECONDO BIENNIO** | X |  | **QUINTO ANNO** |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **CONOSCENZE** | **ABILITA'** | **CONTENUTI SPECIFICI** |
| **Anno 3°** | Conosce in modo approfondito un linguaggio di alto livello.  . | Saper tradurre un problema scientifico in un programma mediante i costrutti della programmazione strutturata ed orientate agli oggetti. Saper scompossre un problema in sottoproblemi. Saper utilizzare i vettori.  Impostare una applicazione in ambiente visual. Scrivere funzioni di risposta agli eventi utilizzando gli strumenti della programmazione strutturata. Implementare una classe con I relativi metodi e proprietà | *UA1 – Le basi della programmazione a oggetti, visuale e ad eventi*  *Progettazione dell’interfaccia grafica: struttura di un form, controlli di input, di output, proprietà.*  Interazione con l’utente: metodi di gestione degli eventi. Struttura di una classe. Uso del debugger.  *Costrutti della programmazione strutturata.* Algoritmi notevoli utilizzati nello studio delle discipline scientifiche.  Uso delle funzioni matematiche e creazione di oggetti. Vettori: metodi di ordinamento e ricerca del massimo e del minimo. Gestione delle stringhe e degli array di stringhe. Suddivisione di un problema in sottoproblemi, funzioni e metodi static. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Anno 4°** | Conosce un linguaggio di programmazione a oggetti.  Conosce un linguaggio per la gestione di un database. | Saper realizzare semplici programmi mediante il paradigma della programmazione orientate agli oggetti.  Rilevare i limiti di una gestione non integrata degli archivi - Progettare semplici modelli E-R con alcune entità e le relative associazioni - Riconoscere i tipi di associazioni tra entità - Tradurre un modello E-R in un modello logico relazionale - Definire il modello fisico - Usare gli operatori fonda-mentali dell’algebra relazionale (selezione, proiezione, join) - Riconoscere basi di dati normalizzate/non normalizzate. Codificare e validare in linguaggio SQL le operazioni di selezione, proiezione, join. Codificare e validare in linguaggio SQL raggruppa-menti, ordinamenti, funzioni di aggregazione. - Codificare e validare in linguaggio SQL condizioni sui raggruppamenti - Utilizzare un particolare DBMS per creare tabelle e impostare interrogazioni SQL. | *UA1 – Programmazione orientata agli oggetti*  *Principi della programmazione ad oggetti: incapsulamento, polimorfismo ed ereditarietà.*  *Progettazione delle classi e creazione degli oggetti all’interno di un metodo main per risolvere semplici problemi scientifici.*  *UA2 – Sistemi informativi e modelli di dati*  Ciclo di vita di un sistema informativo. Dati e informazioni, schema ed istanza dei dati, terminologia specifica sulle basi di dati. Caratteristiche di un DBMS. Definizione di modello concettuale, modello logico, modello fisico. Viste. Indipendenza tra i livelli. Linguaggi per database: DDL, DML, Query Language. Modello concettuale: il modello E-R, entità, associazioni, attributi, chiavi. Associazioni binarie tra entità di tipo 1-1,1-N,N-M,ISA; associazioni n-arie. Associazioni ricorsive. Rappresentazione grafica del modello concettuale attraverso gli schemi E-R.  Modello logico: il modello relazionale, terminologia e rappresentazione tabellare. Regole di derivazione del modello relazionale dal modello E-R. Vincoli di integrità referenziale. Operazioni relazionali: selezione, proiezione, join. Il problema delle anomalie. Il processo di normalizzazione fino alla terza forma normale. Modello fisico: costruzione del modello fisico. Tipi di dati e dimensionamento degli attributi. Scelta delle chiavi, indicizzazione.  *UA3 – SQL*  Creazione di tabelle e aggiornamento dei dati. Data definition language (DDL): comandi CREATE, ALTER, DROP. Data manipulation language (DML): comandi INSERT, UPDATE, DELETE. Interrogazione dei dati: il comando SELECT, operazioni di selezione, proiezione, join. Operatori AND, OR, NOT, LIKE, BETWEEN. Funzioni di aggregazione: COUNT, SUM, AVG, MAX, MIN. ORDER BY. Raggruppamenti con GROUP BY, clausola HAVING  *UA4 – Presentazione di un DBMS specifico: Access, MySql, Postgres,…*  Definizione di tabelle, attributi, vincoli sugli attributi, associazioni tra tabelle, vincoli di integrità referenziale, indici, gestione di query, aggiornamenti. |

**COMPETENZE**: le strutture dati di vettore, matrice e record sono presentate in relazione a semplici problemi da risolvere. Sono affrontati gli algoritmi notevoli per la gestione dei vettori, senza uso di puntatori, con lo scopo di modellizzare problemi la cui soluzione richiede dati strutturati. I file di record vengono sviluppati come termine di confronto con i database.

Lo studente è in grado di gestire semplici applicazioni con interfaccia visuale che rispondono ad eventi e che implementano/utilizzano classi di oggetti (il linguaggio può essere quello del primo biennio). E’ inoltre in grado di modellare i dati presenti in un determinato contesto, progettando a livelli diversi di astrazione: livello concettuale, più vicino alle esigenze del committente, livello logico, che implica la scelta di un legame di tipo matematico tra i dati con le relative operazioni, livello fisico, che riguarda il modo in cui i dati vengono memorizzati sui supporti.

Sa infine utilizzare un linguaggio non procedurale per l’implementazione, la manipolazione e l’interrogazione di schemi relazionali di basi di dati (SQL).