|  |
| --- |
| ***Premessa:*** |
| 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 **(PECUP)**:   * 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.   **Competenze Disciplinari**   * 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.   La metodologia utilizzata sarà strettamente connessa all'individuazione degli obiettivi e dei contenuti, di particolare importanza è la scelta delle strategie operative e dei mezzi più idonei alla realizzazione del processo di apprendimento da parte degli alunni. In tale ottica, gli strumenti della mediazione didattica terranno conto del contesto e assicureranno la sostenibilità delle attività proposte e un generale livello di inclusività.  Per quanto non espressamente esplicitato si fa riferimento ai Verbali di dipartimento ratificati dal Collegio dei Docenti per il corrente anno scolastico (standard di valutazione, verifica trasversale per “Assi culturali” e classi parallele, attività interdisciplinari, visite, viaggi, sopralluoghi aziendali, attività formative extracurricolari, P.T.O.F.) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Modulo1: la progettazione degli algoritmi** | | | | **Periodo: Settembre - Novembre** | | |
| **Competenze disciplinari** | **Livelli di apprendimento** | **Conoscenze** | **Abilità** | | **Indicazioni**  **Metodologiche** | **Attività e Verifiche** |
| Individuare le fasi necessarie per passare da un problema alla sua soluzione  Capire il concetto di algoritmo e le istruzioni che lo compongono  Saper utilizzare i formalismi per descrivere un algoritmo | **A - Livello di apprendimento avanzato** | Definizione e proprietà degli algoritmi.  Formalismi per la costruzione di algoritmi.  Istruzioni di input/output.  Operazioni di assegnazione, aritmetiche e matematiche.  Le strutture di sequenza, selezione, iterazione (anche annidate).  Teorema di Bohm-Jacopini.  Tabella di traccia. | Utilizzare la pseudo codifica e i diagrammi a blocchi.  Costruire algoritmi strutturati.  Utilizzare le strutture di selezione binaria e di selezione multipla.  Individuare quando utilizzare le diverse forme della struttura di ripetizione.  Costruire la tabella di traccia per verificare la correttezza dell’algoritmo. | | Lezione dialogata.  Problem solving e lavoro individuale e/o di gruppo.  Esercitazioni individuali e/o di gruppo. | Interrogazione individuale.  Verifica formativa.  Test online.  Prove pratiche. |
| **B - Livello di apprendimento intermedio** | Definizione e proprietà degli algoritmi.  Formalismi per la costruzione di algoritmi.  Istruzioni di input/output.  Semplici operazioni di assegnazione, aritmetiche e matematiche.  Strutture di sequenza, selezione, iterazione (non annidate). | Utilizzare la pseudo codifica e i diagrammi a blocchi.  Costruire algoritmi strutturati.  Utilizzare le strutture di selezione binaria e di selezione multipla.  Utilizzare le diverse forme della struttura di ripetizione. | |
| **C - Livello di apprendimento base** | Definizione e proprietà degli algoritmi.  Istruzioni di input/output semplici.  Semplici operazioni di assegnazione ed aritmetiche.  Strutture di sequenza, selezione, iterazione (non annidate). | Costruire semplici algoritmi strutturati.  Utilizzare la struttura di selezione binaria.  Utilizzare le diverse forme della struttura di ripetizione. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Modulo 2: il linguaggio C++** | | | | **Periodo: Ottobre - Novembre** | | |
| **Competenze disciplinari** | **Livelli di apprendimento** | **Conoscenze** | **Abilità** | | **Indicazioni**  **Metodologiche** | **Attività e Verifiche** |
| Sviluppare i problemi codificando l’algoritmo risolutivo con un linguaggio di programmazione  Costruire programmi eseguibili dal computer e controllare l’esecuzione del programma | **A - Livello di apprendimento avanzato** | Struttura generale di un programma in linguaggio C++.  Variabili e costanti.  Tipi di dati - Operatore di casting.  Caratteristiche delle istruzioni e degli operatori.  Istruzioni di input/output.  Codifica della struttura di selezione anche multipla (switch).  Codifica delle strutture di iterazione anche con forme particolari. | Scrivere programmi utilizzando in modo ragionato e corretto la sintassi e le potenzialità del linguaggio C++.  Saper predisporre un’interfaccia utente ordinata.  Saper documentare un programma utilizzando il linguaggio tecnico.  Codificare e validare gli algoritmi. | | Lezione dialogata.  Problem solving e lavoro individuale e/o di gruppo.  Esercitazioni individuali e/o di gruppo | Interrogazione individuale.  Verifica formativa.  Test online.  Prove pratiche. |
| **B - Livello di apprendimento intermedio** | Struttura generale di un programma in linguaggio C++.  Variabili e costanti.  Principali tipi di dati - Operatore di casting.  Caratteristiche delle istruzioni.  Istruzioni di input/output.  Codifica della struttura di selezione anche multipla (switch).  Codifica delle strutture di iterazione. | Scrivere programmi utilizzando in modo corretto la sintassi e le potenzialità del linguaggio C++.  Saper predisporre un’interfaccia utente ordinata.  Codificare e validare gli algoritmi. | |
| **C - Livello di apprendimento base** | Struttura generale di un programma in linguaggio C++.  Variabili e costanti.  Tipi di dati essenziali.  Istruzioni di input/output.  Codifica della struttura di selezione non annidata.  Codifica delle strutture di iterazione. | Scrivere programmi utilizzando in modo corretto la sintassi di semplici istruzioni del linguaggio C++.  Saper utilizzare correttamente le istruzioni di input/output. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Modulo 3: le funzioni** | | | | **Periodo: Dicembre - Gennaio** | | |
| **Competenze disciplinari** | **Livelli di apprendimento** | **Conoscenze** | **Abilità** | | **Indicazioni**  **Metodologiche** | **Attività e Verifiche** |
| Controllare la complessità degli algoritmi organizzando il programma in moduli | **A - Livello di apprendimento avanzato** | Concetto di sviluppo top-down.  Organizzazione del programma in funzioni.  Funzioni con parametri e passaggio dei parametri.  Prototipi delle funzioni.  Regole di visibilità delle risorse (risorse globali e locali).  Le funzioni ricorsive. | Scomporre il programma in funzioni.  Riutilizzare più volte le stesse funzioni assegnando diversi valori ai parametri.  Saper scrivere funzioni ricorsive. | | Lezione dialogata.  Problem solving e lavoro individuale e/o di gruppo.  Esercitazioni individuali e/o di gruppo. | Interrogazione individuale.  Verifica formativa.  Test online.  Prove pratiche. |
| **B - Livello di apprendimento intermedio** | Concetto di sviluppo top-down.  Organizzazione del programma in funzioni.  Funzioni con parametri e passaggio dei parametri.  Regole di visibilità delle risorse (risorse globali e locali). | Scomporre il programma in funzioni.  Riutilizzare più volte le stesse funzioni assegnando diversi valori ai parametri. | |
| **C - Livello di apprendimento base** | Concetto di sviluppo top-down.  Organizzazione del programma in funzioni senza parametri.  Variabili globali e locali. | Scomporre il programma in funzioni. | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Modulo 4: le strutture di dati E algoritmi classici** | | | | **Periodo: Gennaio - Maggio** | | |
| **Competenze disciplinari** | **Livelli di apprendimento** | **Conoscenze** | **Abilità** | | **Indicazioni**  **Metodologiche** | **Attività e Verifiche** |
| Organizzare i dati in strutture e implementare gli algoritmi per la loro gestione  Rifarsi a metodologie classiche e soluzioni ottimali nell’individuare una soluzione per problemi specifici | **A - Livello di apprendimento avanzato** | Strutture di dati: vettori, vettori paralleli, matrici, record, vettori di record.  Sommatori e contatori.  Operazioni sui vettori (somma di elementi, media di elementi, ricerca del minimo e del massimo).  Operazioni sulle matrici (somma di elementi, media di elementi, ricerca di minimo e massimo in una matrice).  Algoritmi di ordinamento: per selezione e bubblesort su qualsiasi tipologia di vettore.  Algoritmi di ricerca sequenziale: su chiave secondaria, su chiave primaria, ricerca dicotomica su qualsiasi tipologia di vettore.  Scegliere l’algoritmo adeguato alla situazione.  La complessità degli algoritmi. | Scegliere la struttura di dati più idonea per ogni situazione problematica.  Saper effettuare operazioni di caricamento e stampa dei dati in una struttura qualsiasi  Saper effettuare le operazioni anche complesse sulle diverse strutture dati.  Saper codificare gli algoritmi classici (tutti).  Saper applicare le metodologie utilizzate negli algoritmi classici di ordinamento (tutti) e di ricerca (tutti).  Saper applicare l’algoritmo adeguato alla situazione.  Saper calcolare la complessità di un algoritmo. | | Lezione dialogata.  Problem solving e lavoro individuale e/o di gruppo.  Esercitazioni individuali e/o di gruppo. | Interrogazione individuale.  Verifica formativa.  Test online.  Prove pratiche. |
| **B - Livello di apprendimento intermedio** | Strutture di dati: vettori, vettori paralleli, matrici, record, vettori di record.  Sommatori e contatori.  Operazioni sui vettori (somma di elementi, media di elementi, ricerca del minimo e del massimo).  Operazioni sulle matrici (somma di elementi, media di elementi, ricerca di minimo e massimo in una matrice).  Algoritmo di ordinamento per selezione su qualsiasi tipologia di vettore.  Algoritmi di ricerca sequenziale: su chiave secondaria, su chiave primaria, ricerca dicotomica su qualsiasi tipologia di vettore. | Scegliere la struttura di dati più idonea per ogni situazione problematica.  Saper effettuare operazioni di caricamento e stampa dei dati in una struttura qualsiasi.  Saper effettuare semplici operazioni sulle diverse strutture dati.  Saper codificare gli algoritmi classici.  Saper applicare le metodologie utilizzate negli algoritmi classici di ordinamento (per selezione) e di ricerca (tutti).  Saper applicare l’algoritmo adeguato alla situazione. | |
| **C - Livello di apprendimento base** | Strutture di dati: vettori, matrici, record, vettori di record.  Sommatori e contatori.  Operazioni sui vettori (somma di elementi, media di elementi).  Operazioni sulle matrici (somma di elementi, media di elementi).  Algoritmo di ordinamento per selezione di un vettore.  Algoritmi di ricerca sequenziale: su chiave secondaria e su chiave primaria su un vettore (non parallelo o di record). | Scegliere la struttura di dati più idonea per ogni situazione problematica.  Saper effettuare operazioni di caricamento e stampa dei dati in una struttura qualsiasi.  Saper effettuare semplici operazioni su vettori e matrici.  Saper codificare l’algoritmo di ordinamento per selezione e applicarlo ad un semplice vettore  Saper codificare l’algoritmo di ricerca sequenziale e applicarlo ad un semplice vettore | |