

### **Dal problema all' algoritmo**

- Problema e strategia risolutiva
- Risolutore, esecutore e utente
- Linguaggio naturale e di programmazione
- Concetto e definizione di algoritmo

### **Codifica degli algoritmi**

- Variabili, costanti e valutazione delle espressioni
- Istruzioni operative: input, output e assegnazione
- Algebra booleana: operatori NOT, AND, OR, XOR e leggi di De Morgan
- Paradigma della programmazione strutturata
- Istruzioni di controllo: sequenza, selezione e iterazione
- Rappresentazione di algoritmi tramite diagrammi di flusso
- Realizzazione di diagrammi di flusso usando Flowgorithm
- Debugging e tavola di traccia

### **Fondamenti di programmazione in C++ e Java**

- Dal codice sorgente all'esecuzione: fase di progetto e fase di collaudo
- Linguaggi compilati, interpretati e pseudocompilati
- Peculiarità dei linguaggi di programmazione C++ e Java
- Tipi di dati elementari: interi, reali, caratteri e booleani
- Tipi di dati strutturati: stringhe
- Costrutti condizionali
- Costrutti iterativi
- Codifica degli algoritmi dai diagrammi di flusso ai linguaggi di programmazione in C++ e Java

### **Scomposizione di programmi in sottoprogrammi in C++ e Java**

- Ambiente e regole di visibilità
- Sottoprogrammi: funzioni e procedure
- Passaggio dei parametri: per valore e per riferimento
- Funzioni ricorsive

### **Dati strutturati in C++ e Java**

- Array, array paralleli e matrici (accenni teorici)
- Passaggio degli array come parametri dei sottoprogrammi
- Record e tabelle (accenni teorici)

### **Algoritmi classici**

- Algoritmi di ricerca: ricerca lineare (o sequenziale), ricerca binaria (o dicotomica)
- Algoritmi di ordinamento: selection sort, bubble sort, insertion sort, merge sort, quick sort