



Concetti di Base sulla Programmazione

Prof.Ing.S.Cavalieri



La Programmazione

Dato un problema INFORMATICO, programmare significa:

- Definire i dati in ingresso e il risultato che si vuole raggiungere
- Definire l'algoritmo risolutivo, che permette di raggiungere il risultato a partire dai dati; l'algoritmo deve essere finito e realizzabile con le risorse disponibili
- Tradurre l'algoritmo nel linguaggio di programmazione scelto, attraverso la particolare tecnica di programmazione scelta, rispettando le regole sintattiche del linguaggio



Linguaggi di programmazione: Linguaggio Macchina

- I primi linguaggi di programmazione coincidevano con l'insieme delle istruzioni eseguibili dall'hardware.
- Le istruzioni hardware sono codificate in codice binario: ogni informazione è rappresentata, all'interno della macchina, come una sequenza di bit.
 - ❖ Esempio: **00110010**, dove 0011 è la rappresentazione interna di un comando e 0010 è un operando
- Enorme sforzo richiesto per codificare algoritmi semplici.



Linguaggi di programmazione: Assembly

- Dalla nascita dei primi calcolatori, si è cercato di diminuire lo sforzo e aumentare la produttività dell'utente, anche a costo di caricare la macchina di maggiori compiti.
- E' meglio risparmiare il tempo dell'uomo anche a costo di sprecare tempo-macchina (traduzione di programmi e scarsa efficienza del programma).
- La prima evoluzione dei linguaggi di programmazione ha portato ad una codifica di tipo simbolico, anziché binaria, dei programmi.
- Tali versioni simboliche dei linguaggi hardware sono note come linguaggi **assembly**

Linguaggi di programmazione: Evoluzione

- Il passo successivo nell'evoluzione dei linguaggi di programmazione tesa a rendere la codifica degli algoritmi il più possibile “vicina” al problema da risolvere anziché all'architettura della macchina destinata all'esecuzione del programma.
- Non a caso i primi due linguaggi di alto livello, FORTRAN e COBOL, hanno costrutti fortemente ispirati alla notazione usata per descrivere i problemi di maggior rilievo per il calcolo automatico degli anni '50, cioè **l'elaborazione numerica** e quella **gestionale**.

Alto Livello

Assembly

Macchina/binario

MicroProgrammi

Hardware



Linguaggi di programmazione: Evoluzione

- Gli anni '70 vedono il fiorire di linguaggi general-purpose:
 - ❖ il SIMULA 67, l'ALGOL 68, ma soprattutto il PASCAL e il C
- Dagli anni '80 in poi si ha l'affermarsi dei linguaggi orientati agli oggetti (Object-Oriented -OO):
 - ❖ C++, Java, C#



La Tecnica di Programmazione

- Non è legata ad un linguaggio, anche se alcuni linguaggi sono ideali per essere abbinati ad una specifica tecnica di programmazione

- Esistono tre tecniche:
 - a) Programmazione NON Strutturata, che produce lo Spaghetti Code
 - b) Programmazione Strutturata: Sequenza, Selezione, Ripetizione
 - c) Programmazione ad Oggetti: Classe, Oggetto, Metodo



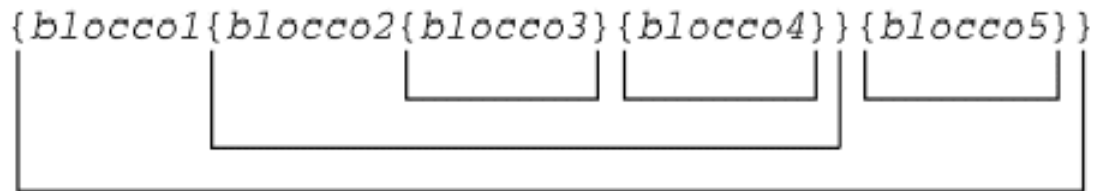
La Programmazione Strutturata

- L'obiettivo della *programmazione strutturata* è di rendere un flusso ordinato il passaggio tra le istruzioni dall'inizio alla fine dei programmi.
- Corrado Böhm e Giuseppe Jacopini hanno dimostrato che tutti i programmi possono essere scritti con l'utilizzo di:
 - ❖ *sequenza*,
 - ❖ *selezione* e
 - ❖ *iterazione* (senza l'uso del salto *goto* tipico dello *Spaghetti Code*).

La Programmazione Strutturata

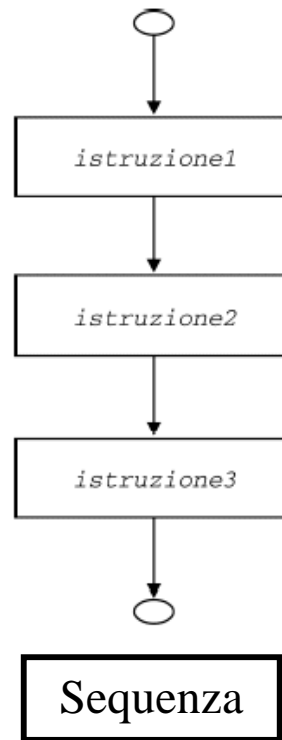
Il *Blocco* è essenziale per la Programmazione Strutturata

Blocco d'istruzioni: un insieme d'istruzioni con una sola entrata, da dove inizia sempre l'elaborazione, e una sola uscita, dove termina sempre l'elaborazione del blocco {blocco}.



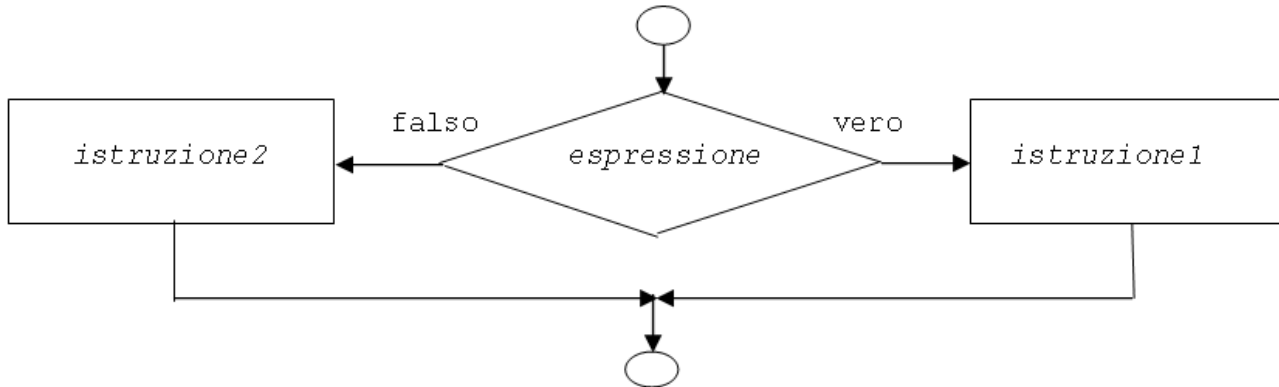
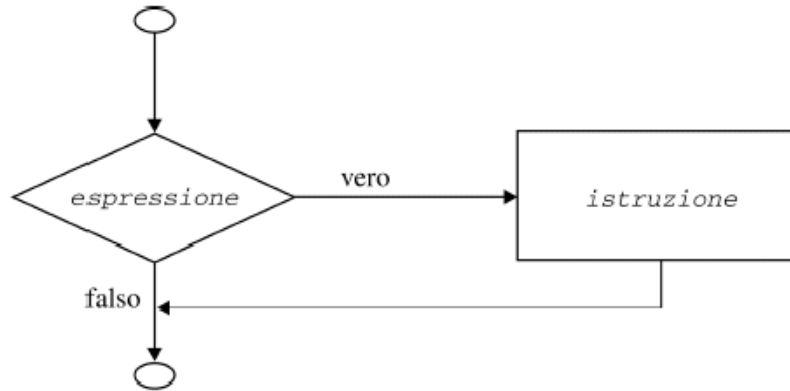
Da ora in poi, quando si parlerà di istruzione si può riferire ad una singola istruzione o a un blocco di istruzione.

La Programmazione Strutturata



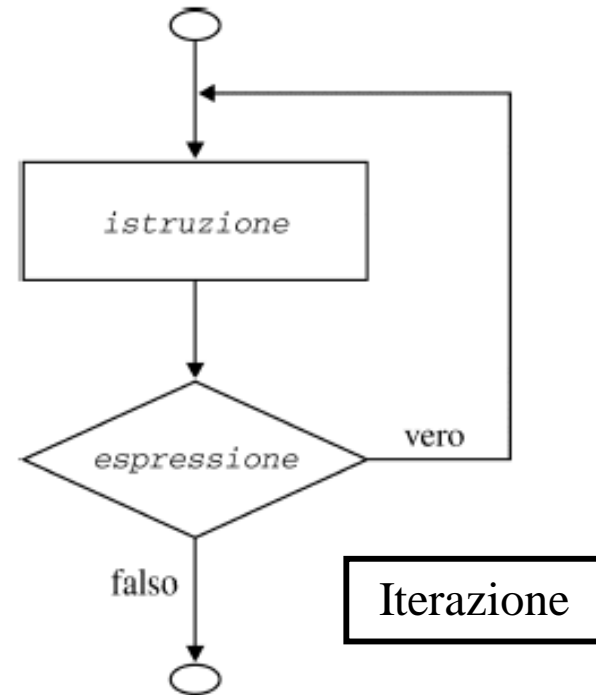
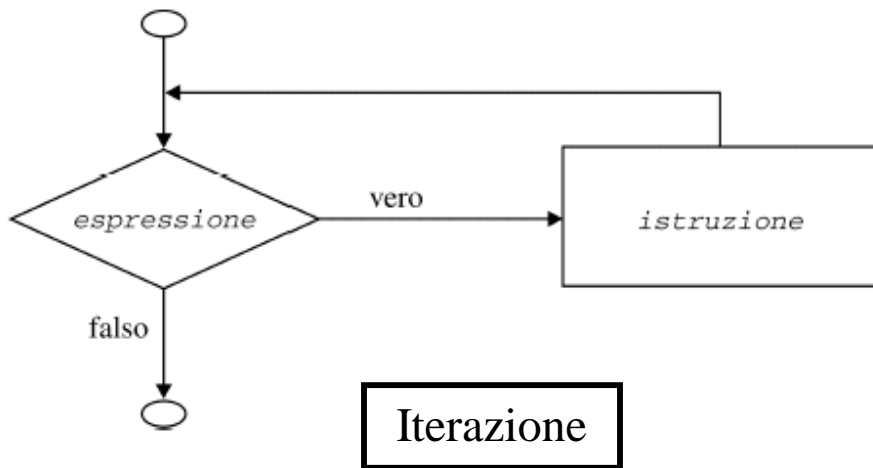
La Programmazione Strutturata

Selezione

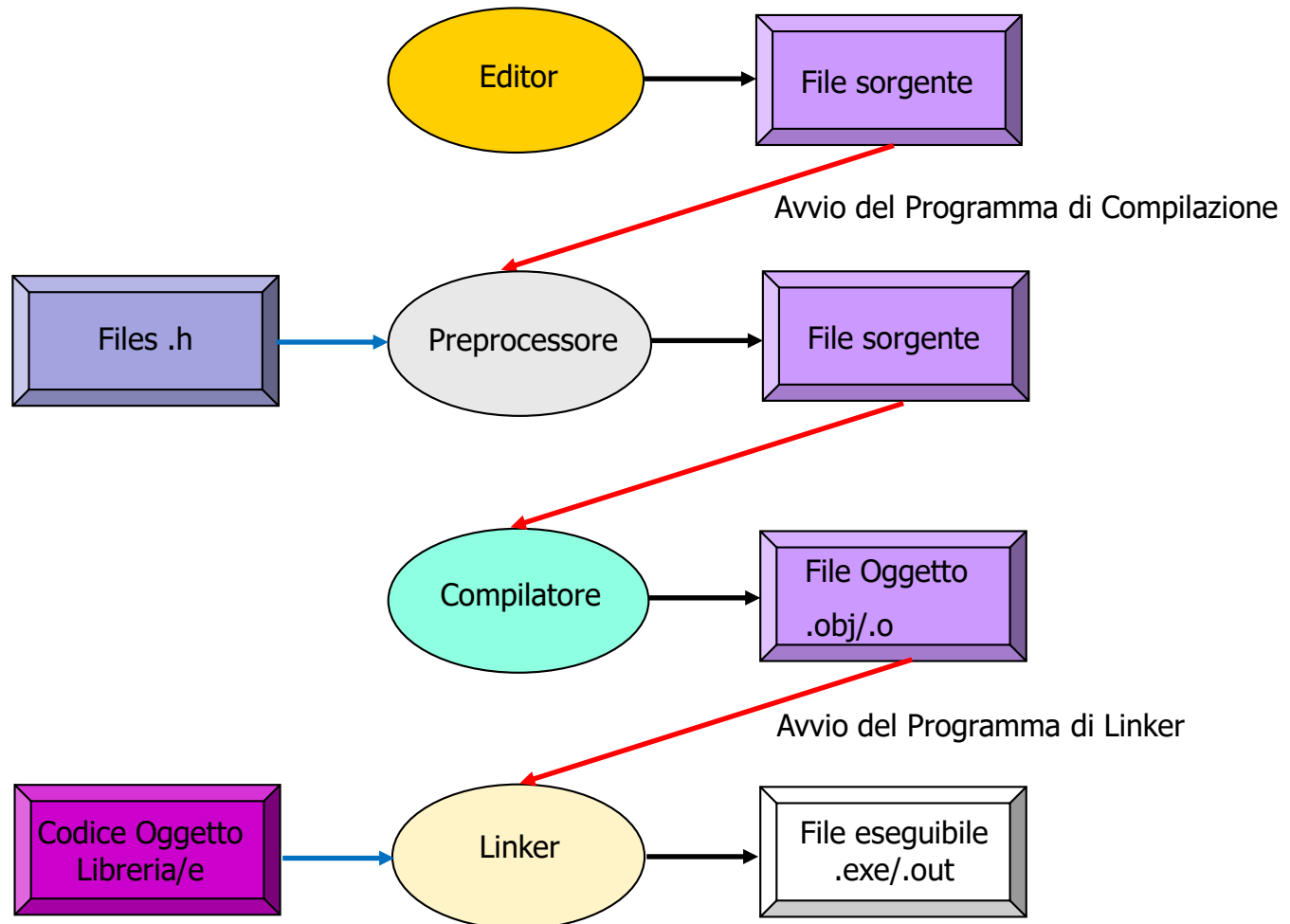


Selezione

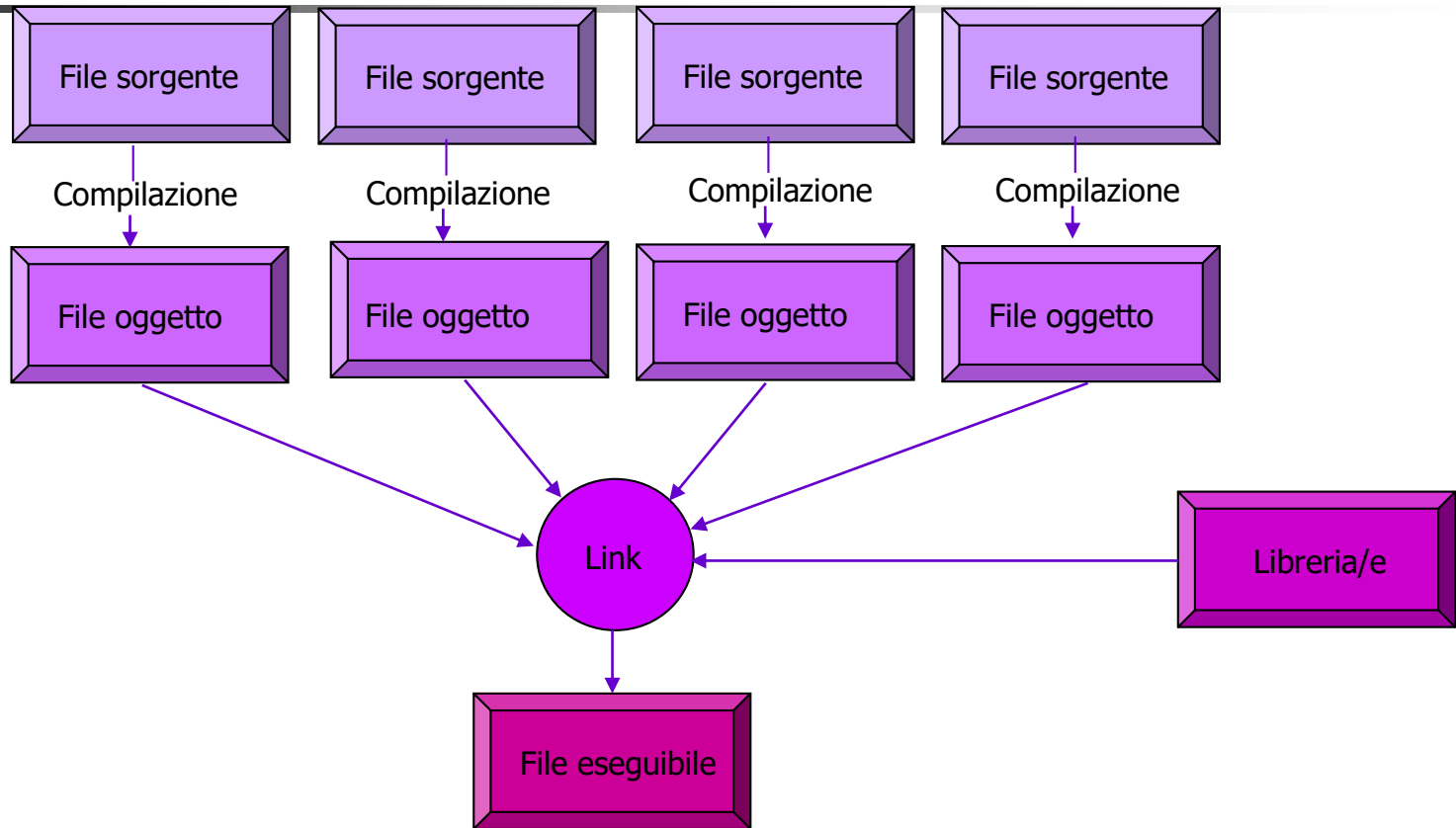
La Programmazione Strutturata



Generazione File Eseguibile



Generazione File Eseguibile



- I codici sorgente ed oggetto possono essere suddivisi in più file, il codice eseguibile di un programma risiede in un unico file



Il Compilatore

- Il compilatore è esso stesso un programma (o un gruppo di programmi)
- Durante la compilazione avviene:
 - ❖ Controllo del codice sorgente da parte di un preprocessore: interpretare speciali direttive di precompilazione,
 - ❖ Controllo di eventuali errori sintattici (NON LOGICI),
 - ❖ Produzione di un file in codice Assembly,
 - ❖ Trasformazione del file in codice Assembly in un altro file in codice OGGETTO (Macchina): estensione .obj



Il Linker

- Il Linker è generalmente un programma distinto dal compilatore: in alcuni ambienti il programma di link deve essere lanciato separatamente
- Il Linker fa in modo che:
 - ❖ i riferimenti tra diversi file oggetto e con le librerie vengano risolti
 - ❖ i file oggetto creati dal compilatore vengano trasformati in un unico file eseguibile, al quale vengano associate le librerie necessarie all'esecuzione
- Il risultato prodotto dal Linker è un programma eseguibile, che può essere caricato in memoria



La Fase di Caricamento

- Il Caricamento di un programma in memoria, avviene nel momento in cui l'utente richiede la sua esecuzione
- Durante la fase di caricamento (o *loading*), il programma eseguibile viene caricato nella memoria del calcolatore;
- Le Fasi di Compilazione, Linker e Loading sono separate (anche temporalmente)
 - ❖ Attenzione: l'uso dei compilatori crea false impressioni !