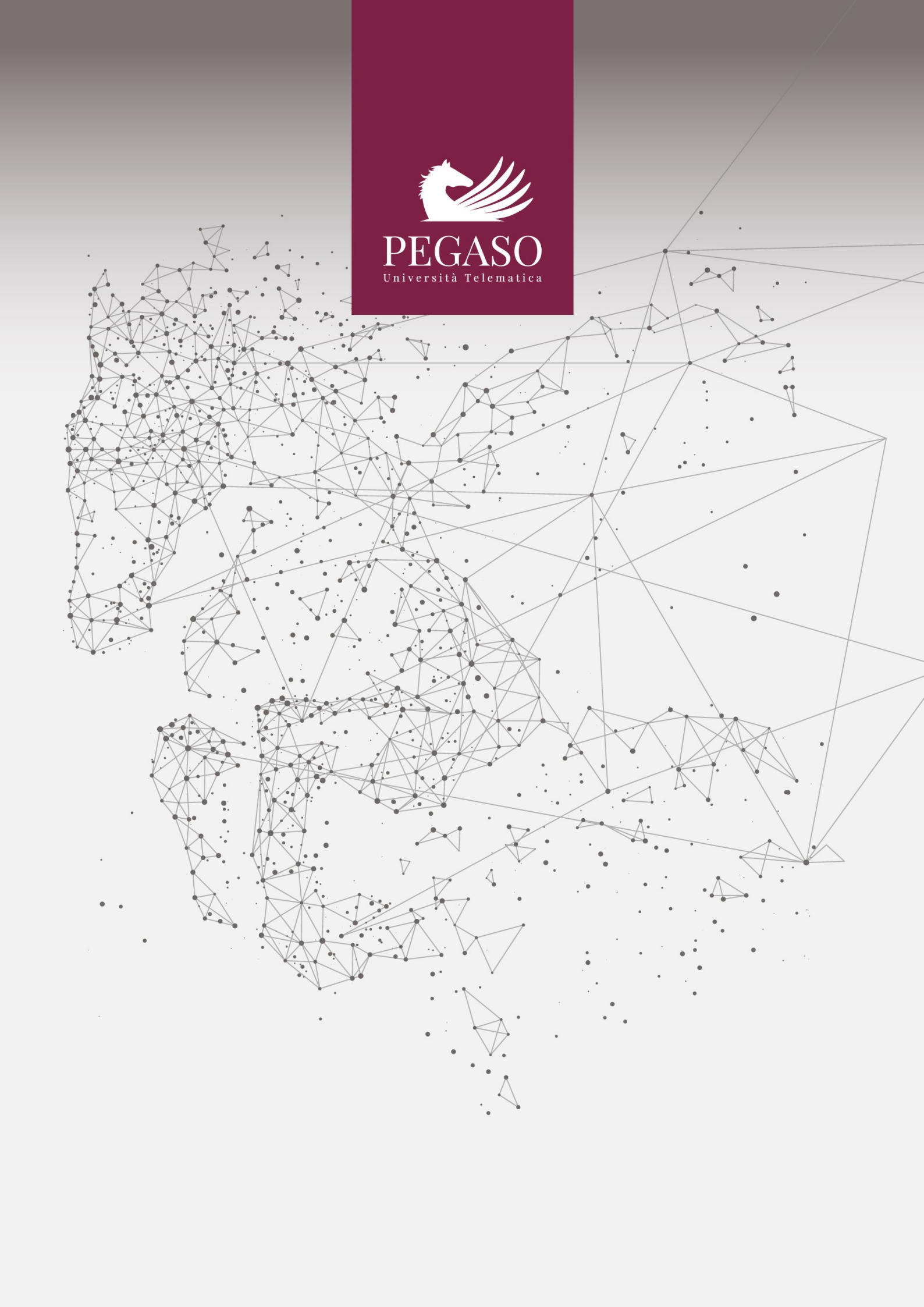

PEGASO
Università Telematica

Indice

1. Istruzione if...else

Ricordiamo che l'istruzione di selezione if esegue un'azione indicata solo quando la condizione è vera; altrimenti l'azione viene saltata.

L'istruzione di selezione if...else permette di specificare che vanno eseguite azioni differenti quando la condizione è vera e quando è falsa.

Ad esempio, l'istruzione nello pseudocodice:

Se il voto dello studente è maggiore o uguale a 18

Stampa "Passed"

altrimenti

Stampa "Failed"

stampa Passed se il voto dello studente è maggiore o uguale a 18 e Failed se è inferiore a 18.

In entrambi i casi, dopo che avviene la stampa, viene "eseguita" la successiva istruzione in sequenza nello pseudocodice.

Anche il corpo dell'altrimenti viene indentato.

La precedente istruzione nello pseudocodice Se…altrimenti può essere scritta in C come

```
if ( grade >= 18 ) {
   puts( "Passed" );
} // fine di if
else {
   puts( "Failed" );
} // fine di else
```

☺ *Buona pratica di programmazione*

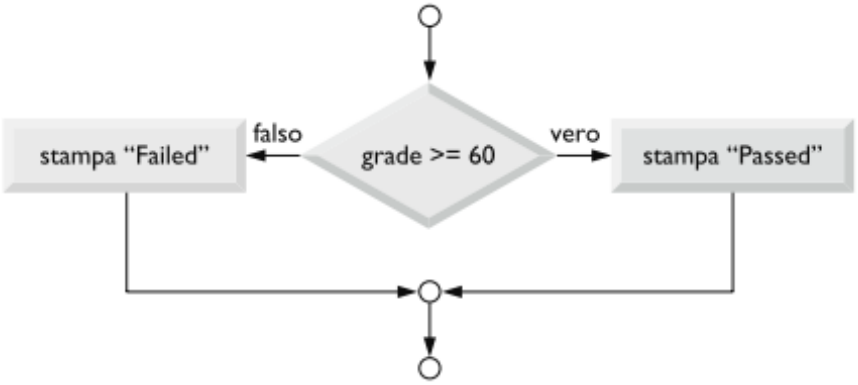
Indentare ambedue le istruzioni del corpo di un'istruzione if...else (sia in pseudocodice che in C).

☺ *Buona pratica di programmazione*

Se vi sono diversi livelli di indentazione, ciascuno di essi deve essere indentato con la stessa quantità di spazio.

Il seguente diagramma di flusso illustra il flusso di controllo nell'istruzione if...else.

Oltre a cerchietti e frecce, gli unici simboli nel diagramma di flusso sono rettangoli (per le azioni) e un rombo (per una decisione).



2. Operatore condizionale ?:

Il C fornisce l'operatore condizionale (**?:**), che è strettamente correlato all'istruzione if...else. L'operatore condizionale è il solo operatore *ternario* in C, ovvero si applica a tre operandi. Questi, insieme all'operatore condizionale, formano un'espressione condizionale.

- Il primo operando è una condizione.
- Il secondo operando è il valore dell'intera espressione condizionale se la condizione è vera.
- il terzo operando è il valore dell'intera espressione condizionale se la condizione è falsa.

Ad esempio, l'istruzione puts:

```
puts( grade >= 18 ? "Passed" : "Failed" );
```

contiene come suo argomento un'espressione condizionale che assume come valore la stringa "Passed" se la condizione grade >= 18 è vera e la stringa "Failed" se la condizione è falsa.

L'istruzione puts ha essenzialmente lo stesso comportamento della precedente istruzione if...else.

Il secondo e il terzo operando di un'espressione condizionale possono essere anche azioni da eseguire. Ad esempio, l'espressione condizionale

```
grade >= 18 ? puts( "Passed" ) : puts( "Failed" );
```

si legge: "Se grade è maggiore o uguale a 18, allora puts("Passed"), altrimenti puts ("Failed")."

Questa è pure paragonabile alla precedente istruzione if...else.

Gli operatori condizionali possono essere usati in alcuni punti dove non è possibile ricorrere all'istruzione if...else, includendo espressioni e argomenti a funzioni (come printf).

😊 *Prevenzione di errori*

Utilizzate espressioni dello stesso tipo per il secondo e il terzo operando dell'operatore condizionale (?:) per evitare errori insidiosi.

3. Istruzioni annidate if...else

Le istruzioni annidate if...else effettuano test su casi multipli attraverso il piazzamento di istruzioni if...else all'interno di altre istruzioni if...else.

Ad esempio, la seguente istruzione in pseudocodice stamperà A per i voti dell'esame maggiori o uguali a 27, B per i voti maggiori o uguali a 24 (ma meno di 27), C per i voti maggiori o uguali a 21 (ma meno di 24), D per i voti maggiori o uguali a 18 (ma meno di 21) e F per tutti gli altri voti.

Se il voto dello studente è maggiore o uguale a 27

Stampa "A"

altrimenti

Se il voto dello studente è maggiore o uguale a 24

Stampa "B"

altrimenti

Se il voto dello studente è maggiore o uguale a 21

Stampa "C"

altrimenti

Se il voto dello studente è maggiore o uguale a 18

Stampa "D"

altrimenti

Stampa "F

Questo pseudocodice può essere scritto in C come

```
if ( grade >= 27 ) {
   puts( "A" );
} // fine di if
else {
   if ( grade >= 24 ) {
      puts( "B" );
   } // fine di if
   else {
      if ( grade >= 21 ) {
         puts( "C" );
      } // fine di if
      else {
         if ( grade >= 18 ) {
            puts( "D" );
         } // fine di if
         else {
            puts( "F" );
         } // fine di else
      } // fine di else
   } // fine di else
} // fine di else
```

Se la variabile grade è maggiore o uguale a 27, tutte e quattro le condizioni saranno vere, ma soltanto l'istruzione puts dopo il primo test sarà eseguita.

Dopo l'esecuzione di puts, la parte else dell'istruzione if...else "esterna" viene saltata.

Si potrebbe preferire scrivere la precedente istruzione if come:

```
if ( grade >= 27 ) {
   puts( "A" );
} // fine di if
else if ( grade >= 24 ) {
   puts( "B" );
} // fine di else if
else if ( grade >= 21 ) {
   puts( "C" );
} // fine di else if
else if ( grade >= 18 ) {
   puts( "D" );
} // fine di else if
else {
   puts( "F" );
} // fine di else
```

Per quanto riguarda il compilatore C, le due forme sono equivalenti.

L'ultima forma è la più comune, dal momento che evita la profonda indentazione a destra del codice.

Tale indentazione lascia spesso poco spazio su una riga, facendo sì che le righe siano spezzate con conseguente riduzione della leggibilità del programma.

L'istruzione di selezione if si aspetta solo un'istruzione nel suo corpo (se avete solo un'istruzione nel corpo di un if, non dovete necessariamente includerla fra parentesi).

Per includere diverse istruzioni nel corpo di un if, dovete includere l'insieme delle istruzioni fra parentesi graffe ({ e }).

L'insieme delle istruzioni contenute dentro una coppia di parentesi graffe è chiamato *istruzione composta* o *blocco*.

L'esempio seguente include un'istruzione composta nella parte else di un'istruzione if...else

```
if ( grade >= 18 ) {
   puts( "Passed." );
} // fine di if
else {
   puts( "Failed." );
   puts( "You must take this course again." );
} // fine di else
```

In questo caso, se il voto è inferiore a 18, il programma esegue entrambe le istruzioni puts nel corpo dell'else e stampa

```
Failed.
You must take this course again.
```

Le parentesi graffe attorno alle due istruzioni nella clausola else sono importanti.

Senza di esse l'istruzione

```
puts( " You must take this course again." );
```

sarebbe al di fuori del corpo della parte else dell'if e verrebbe eseguita a prescindere dal fatto che il voto sia inferiore o meno a 18, quindi anche uno studente che supera la prova dovrebbe ripetere il corso!

Un errore di sintassi è individuato dal compilatore.

Un errore logico ha il suo effetto al momento dell'esecuzione.

Un errore logico irreversibile fa sì che il programma fallisca e termini prematuramente.

Un errore logico non irreversibile permette al programma di continuare l'esecuzione ma gli fa produrre risultati scorretti.

Un'istruzione composta può essere collocata ovunque si possa collocare un'istruzione singola

È anche possibile non avere affatto istruzioni, ossia avere un'istruzione vuota.

L'istruzione vuota è rappresentata mettendo un punto e virgola (;) dove ci sarebbe normalmente un'istruzione.

😊 *Prevenzione di errori*

Includete sempre il corpo delle istruzioni di controllo tra parentesi ({ e }), anche se contiene solo una singola istruzione. Questo risolve il problema dell'else sospeso.

☹ *Errore comune di programmazione*

L'inserimento di un punto e virgola dopo la condizione in un'istruzione if come in

```
if (voto >= 18);
```

provoca un errore logico nelle istruzioni if a selezione singola e un errore di sintassi nelle istruzioni if a selezione doppia.

😊 *Prevenzione di errori*

L'inserimento di parentesi graffe iniziali e finali nelle istruzioni composte prima di scrivere le singole istruzioni dentro le parentesi aiuta a evitare l'omissione di una o di entrambe le parentesi,

prevenendo errori di sintassi ed errori logici (dove entrambe le parentesi graffe sono davvero necessarie).

Bibliografia

- Paul Deitel, Harvey Deitel, "Il linguaggio C – Fondamenti e tecniche di programmazione", Libro edito da Pearson Italia. Include anche utili esercizi di autovalutazione.