

The cover page features a large central title 'Strutture di Controllo' on two lines, with 'Dettagli e Approfondimenti' below it. Above the title is a subtitle 'Programmazione Procedurale'. At the bottom left is the text 'versione 2.7'. Below that is a small note: 'Questo lavoro è concesso in uso secondo i termini di una licenza Creative Commons (vedi ultima pagina)'. To the right is a Creative Commons license logo with the text 'SOME RIGHTS RESERVED' and 'creative commons'.

1

The page shows a table of contents with a header 'Strutture di Controllo: Dettagli e Approfondimenti >> Sommario'. Below it is a section titled 'Sommario' containing a list of topics:

- Valutazione delle Condizioni
 - ⇒ Condizioni a valori non booleani
 - ⇒ Operatore “Condizionale”
- Utilizzo delle Espressioni
- Semantica degli If Nidificati
- Altre Istruzioni
 - ⇒ Istruzione DO-WHILE
 - ⇒ Istruzione SWITCH

Strutture di Controllo: Dettagli e Approfondimenti >> Condizioni

Valutazione delle Condizioni

- Le strutture di controllo sono basate sulla valutazione di condizioni
- Condizione
 - ⇒ espressione a valori booleani
- Due particolarità del C++
 - ⇒ tutte le istruzioni sono anche espressioni

3

Strutture di Controllo: Dettagli e Approfondimenti >> Condizioni

Condizioni a Valori Non Booleani

- In generale
 - ⇒ un'espressione deve restituire vero o falso
- In C/C++
 - ⇒ qualsiasi tipo di base è compatibile con il tipo bool (per troncamento)
 - ⇒ qualsiasi espressione che ritorni un valore del tipo di base è accettabile come condizione per un'espressione di controllo
 - ⇒ valore 0->false, qualsiasi altro valore->true

4

Strutture di Controllo: Dettagli e Approfondimenti >> Condizioni

Condizioni a Valori Non Booleani

- Esempio

```
numeroEsami = 0;
sommaVoti = 0.0;
cout << "Voto (0 per fermarti)";
cin >> voto;
while (voto) {
    numeroEsami++;
    sommaVoti += voto;
    cout << "Voto (0 per fermarti)" << endl;
    cin >> voto;
}
```

5

Strutture di Controllo: Dettagli e Approfondimenti >> Utilizzo delle Espressioni

Utilizzo delle Espressioni

- Alla luce di quanto detto

- ⇒ le espressioni del C/C++ sono potenti
- ⇒ è possibile scrivere programmi fatti quasi esclusivamente di espressioni

- Un esempio: il for

- ⇒ il for prevede nella sua prima riga tre espressioni
- ⇒ queste forniscono moltissima flessibilità

6

Utilizzo delle Espressioni

- Sintassi del for

```
for (<e1>; <e2>; <e3>) <blocco>
```

⇒ non necessariamente e1 deve essere una inizializzazione

⇒ non necessariamente e2 deve essere una espressione a valori booleani

⇒ non necessariamente e3 deve essere un incremento

Utilizzo delle Espressioni

- Un altro esempio:

⇒ l'operatore “condizionale” ?

⇒ sintassi <e1> ? <e2> : <e3>

⇒ dove <e1>, <e2>, <e3> sono espressioni

- Esempio: minimo tra due numeri

```
int x, y, min;
cin >> x >> y;
min = (x <= y ? x : y);
```

Strutture di Controllo: Dettagli e Approfondimenti >> Utilizzo delle Espressioni

Utilizzo delle Espressioni

- Semantica
 - ⇒ viene valutata l'espressione e1
 - ⇒ se e1 è vera, viene valutata l'espressione e2
 - ⇒ altrimenti viene valutata l'espressione e3
- In sostanza
 - ⇒ consente di sostituire espressioni if-then-else
 - ⇒ a prezzo di rendere il codice meno chiaro

10

Strutture di Controllo: Dettagli e Approfondimenti >> Utilizzo delle Espressioni

Utilizzo delle Espressioni

- Attenzione
 - ⇒ queste funzionalità riducono la leggibilità del codice e complicano la manutenzione
 - ⇒ non dovrebbero essere usate

12

Strutture di Controllo: Dettagli e Approfondimenti >> If Nidificati

Istruzione Condizionali Nidificate

- Semantica
 - ⇒ la semantica è quella ordinaria
- Problema
 - ⇒ è possibile mischiare if-then-else ed if-then
 - ⇒ tra tanti if ed else, come si stabiliscono le associazioni (a quale if si rif. un else) ?
- Regola
 - ⇒ ogni else si riferisce al più vicino tra gli if precedenti privi di else

13

Strutture di Controllo: Dettagli e Approfondimenti >> If Nidificati

Istruzione Condizionali Nidificate

- Esempio
 - ⇒ un programma per l'attribuzione di borse di studio agli studenti meritevoli del 3 anno
- Criteri
 - ⇒ crediti conseguiti (> 140)
 - ⇒ media degli esami ($> 0 \leq 28$)
 - ⇒ vari importi per la borsa (es: 500 e 100 euro)

14

Strutture di Controllo: Dettagli e Approfondimenti >> If Nidificati

Istruzione Condizionali Nidificate

- Esempio


```
if(cd>140) if(media>28) borsa=500; else borsa=100;
```
- Il modo corretto di scrivere il codice


```
if (cd > 140) {
    if (media>28) {
        borsa=500;
    } else {
        borsa=100;
    }
}
```

l'utilizzo delle parentesi
elimina qualsiasi ambiguità

15

Strutture di Controllo: Dettagli e Approfondimenti >> If Nidificati

Istruzione Condizionali Nidificate

- Se non uso le parentesi


```
if(cd>140) if(media>28) borsa=500; else borsa=100;
```
- Due possibili interpretazioni


```
if (cd>140) {
    if (media>28)
        borsa=500;
    else
        borsa=100;
}
```

```
if (cd>140) {
    if (media>28)
        borsa=500;
} else
    borsa=100;
```

16

Strutture di Controllo: Dettagli e Approfondimenti >> If Nidificati

Istruzione Condizionali Nidificate

```
borsa=0;
if (cd>140) {
    if (media>28)
        borsa=500;
    else
        borsa=100;
}
```

Supponiamo all'inizio:

#1499	cd	120
#2500	media	25
#2501	borsa	xxx

I Interpretazione:

#1499	cd	120
#2500	media	25
#2501	borsa	0

II Interpretazione:

#1499	cd	120
#2500	media	25
#2501	borsa	100

17

Strutture di Controllo: Dettagli e Approfondimenti >> If Nidificati

Istruzione Condizionali Nidificate

- Semantica corretta

⇒ borsa solo agli studenti con più di 140 cd.

```
if (cd>140) {
    if (media>28)
        borsa=500;
    else
        borsa=100;
}
```

if esterno
(le parentesi graffe sono ridondanti)

if interno
(l'else viene assoc. all'if più vicino)

18

Strutture di Controllo: Dettagli e Approfondimenti >> If Nidificate

Istruzione Condizionali Nidificate

- Per forzare l'interpretazione alternativa
⇒ utilizzare le parentesi graffe (anche se non indispensabili) per forzare l'associazione
- Semantica alternativa

```
if (cd>140) {
    if (media>28)
        borsa=500;
    } else
        borsa=100;
```

19

Strutture di Controllo: Dettagli e Approfondimenti >> Altre Istruzioni

Altre Istruzioni

- Strutture di controllo fondamentali
 - ⇒ istruzione if
 - ⇒ ciclo while
 - ⇒ ciclo for
- Esistono altre strutture di controllo
 - ⇒ non sono fondamentali (sono sostituibili da quelle fondamentali)
 - ⇒ le elenchiamo per completezza

20

Strutture di Controllo: Dettagli e Approfondimenti >> Altre Istruzioni

Altre Istruzioni

- Altre istruzioni
 - ⇒ istruzione do while (istruzione iterativa a condizione finale)
 - ⇒ istruzione case/switch (istruzione condizionale a molte vie)

21

Strutture di Controllo: Dettagli e Approfondimenti >> Altre Istruzioni

Istruzione DO WHILE

- Ciclo while a condizione finale
 - ⇒ la condizione è alla fine e non all'inizio
- Sintassi


```
do <istruzioni> while (<condizione>);
```
- Semantica
 - ⇒ identica a quella del while
 - ⇒ il corpo viene eseguito almeno una volta prima di valutare la condizione

22

Strutture di Controllo: Dettagli e Approfondimenti >> Altre Istruzioni

Altre Istruzioni

- Esempio: calcolo della media


```
numeroEsami = 0;
sommaVoti = 0.0;
do {
    cout << "Voto (0 per fermarti)" << endl;
    cin >> voto;
    if (voto) {
        numeroEsami++;
        sommaVoti += voto;
    }
} while (voto);
```

23

Strutture di Controllo: Dettagli e Approfondimenti >> Altre Istruzioni

Istruzione CASE

- A volte
 - ⇒ è necessario verificare caso per caso una condizione
 - ⇒ es: riconoscimento dei mesi dell'anno
- In questi casi
 - ⇒ molte istruzioni if-then-else una dopo l'altra
 - ⇒ lunghe e poco leggibili
 - ⇒ in questi casi è possibile utilizzare il case

24

Strutture di Controllo: Dettagli e Approfondimenti >> Altre Istruzioni

Istruzione CASE

○ Esempio: riconoscimento del mese

```
int main() {
    int numero;
    string mese;
    cout << "Inserisci il mese (da 1 a 12): ";
    cin >> numero;
    if (numero == 1)
        mese="gennaio";
    else if (numero == 2)
        mese="febbraio";
    else if (numero == 3)
        mese="marzo";
    ...
}
```

25

Strutture di Controllo: Dettagli e Approfondimenti >> Altre Istruzioni

Istruzione CASE

```
#include <iostream>
using namespace std;
int main() {
    int numero; string mese;
    cout << "Inserisci il mese (da 1 a 12): ";
    cin >> numero;
    switch (numero) {
        case 1: mese = "gennaio"; break;
        case 2: mese = "febbraio"; break;
        case 3: mese = "marzo"; break;
        case 4: mese = "aprile"; break;
        ...
        case 12: mese = "dicembre";
    }
    return 0;
}
```

26

Strutture di Controllo: Dettagli e Approfondimenti >> Altre Istruzioni

Istruzione CASE

- Sintassi


```
switch (<espr>) {
    case <val1>: <istruzioni1>;
    case <val2>: <istruzioni2>;
    ...
    default: <istruzioniN>; // opzionale
}
```
- Dove
 - ⇒ <espr> è un'espressione a valori di tipo enumerabile
 - ⇒ <val1>, <val2>, ... sono possibili valori dell'espressione

27

Strutture di Controllo: Dettagli e Approfondimenti >> Altre Istruzioni

Istruzione CASE

- Semantica
 - ⇒ viene valutata l'espressione
 - ⇒ viene cercato il ramo case con il valore corrispondente; se viene trovato, vengono eseguite le istruzioni relative e quelle di tutti i rami successivi
 - ⇒ altrimenti, se è specificato il ramo default, vengono eseguite le istruzioni relative
 - ⇒ si prosegue con l'istruzione successiva

28

Strutture di Controllo: Dettagli e Approfondimenti >> Altre Istruzioni

Istruzione CASE

- Nota sulla semantica
 - ⇒ vengono eseguite le istruzioni del ramo case selezionato e tutte le istruzioni dei rami successivi
 - ⇒ in molti casi, è necessario eseguire le istruzioni di un solo ramo
 - ⇒ per interrompere l'esecuzione subito dopo il ramo selezionato: istruzione `break`

29

Strutture di Controllo: Dettagli e Approfondimenti >> Sommario

Riassumendo

- Valutazione delle Condizioni
 - ⇒ Condizioni a valori non booleani
 - ⇒ Operatore “Condizionale”
- Utilizzo delle Espressioni
- Semantica degli If Nidificati
- Altre Istruzioni
 - ⇒ Istruzione DO-WHILE
 - ⇒ Istruzione SWITCH

30

Termini della Licenza

○ This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

○ Questo lavoro viene concesso in uso secondo i termini della licenza “Attribution-ShareAlike” di Creative Commons. Per ottenere una copia della licenza, è possibile visitare <http://creativecommons.org/licenses/by-sa/1.0/> oppure inviare una lettera all’indirizzo Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.