



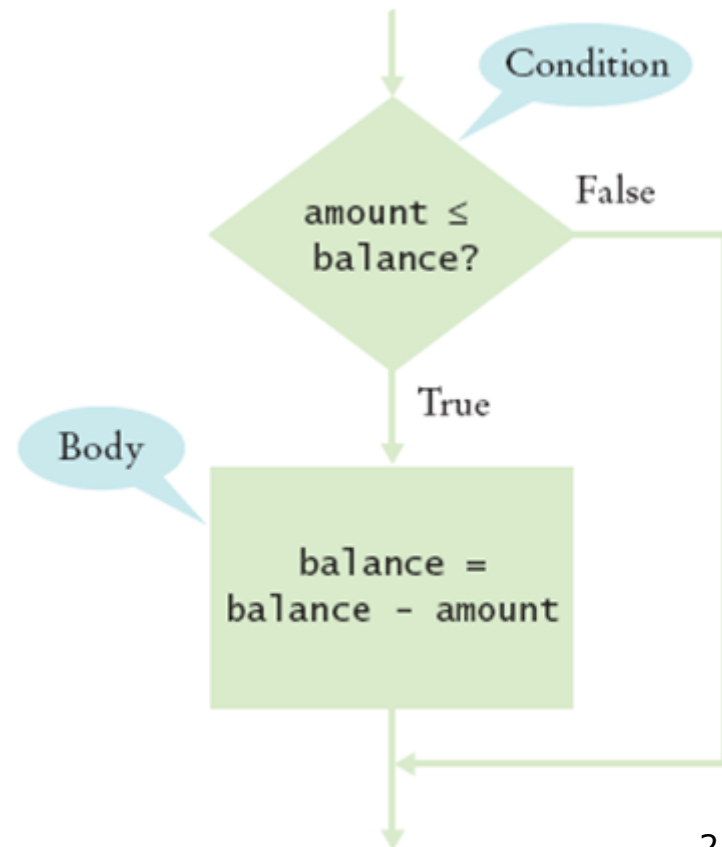
Decisioni

Istruzione if

```
if (amount <= balance)
    balance = balance - amount;
```

Sintassi:

```
if (condizione)
    istruzione
```

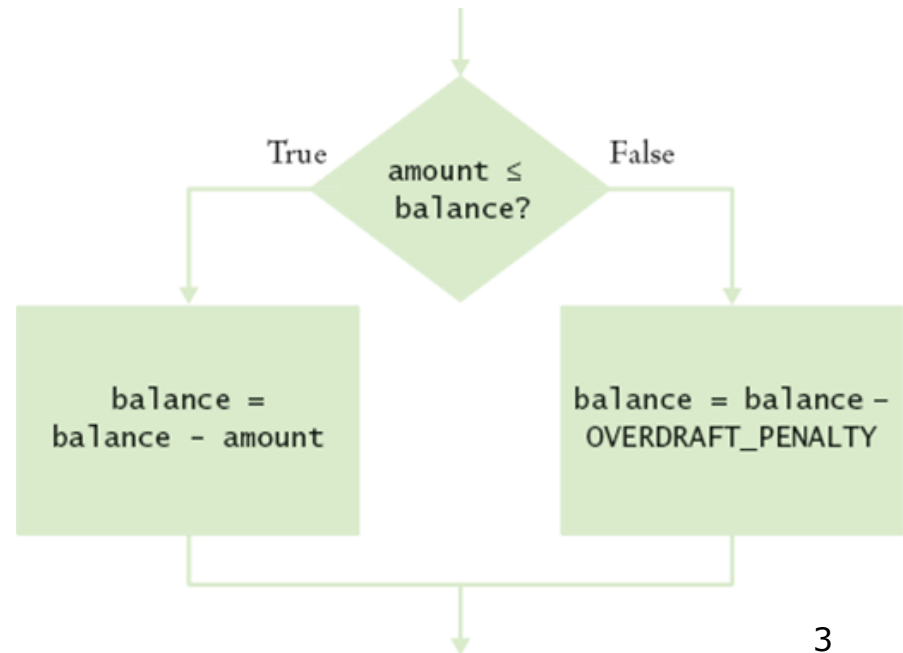


Istruzione if/else

```
if (amount <= balance)
    balance = balance - amount;
else
    balance = balance - OVERDRAFT_PENALTY;
```

Sintassi:

```
if (condizione)
    istruzione
else
    istruzione
```





Blocco di istruzioni

```
{  
    istruzione1  
    istruzione2  
    . . .  
}
```

Obiettivo:

Raggruppare più istruzioni
per formare un'unica
istruzione

Esempio:

```
if (amount <= balance)  
{  
    double newBalance =  
        balance - amount;  
    balance = newBalance;  
}
```

Se la condizione dell'**if**
è verificata vengono
eseguiti tutti gli statement
all'interno del blocco



Tipi di istruzioni

- **Semplice**

```
balance = balance - amount;
```

- **Composto**

```
if (balance >= amount)  
    balance = balance - amount;
```

- **Blocco di istruzioni**

```
{  
    double newBalance = balance - amount;  
    balance = newBalance;  
}
```



Esempio: Retribuzione dei dipendenti

○ **Presentazione del problema**

- Modellare un sistema di retribuzione per dipendenti che sono pagati con una tariffa oraria. Il sistema deve riuscire a calcolare la retribuzione di un dipendente sulla base della tariffa oraria e delle ore di lavoro effettuate e deve stampare il nome, le ore e la paga calcolata. I dipendenti che lavorano più di 40 ore ricevono una somma per gli straordinari, pagati una volta e mezzo la loro tariffa salariale normale. Se un dipendente ha 30 o più ore di straordinario nelle ultime due settimane viene emesso un messaggio d'avviso



Retribuzione

○ Scenario d' esempio

Enter employee name: **Gerald Weiss**

Enter employee rate/hour: **20**

Enter Gerald Weiss' s hours for week 1: **30**

Gerald Weiss earned \$600 for week 1

Enter Gerald Weiss' s hours for week 2: **50**

Gerald Weiss earned \$1100 for week 2

Enter Gerald Weiss's hours for week 3: **60**

Gerald Weiss earned \$1400 for week 3

*** Gerald Weiss has worked 30 hours of overtime in the last two weeks.



Retribuzione

- Oggetti primari
 - Termini chiave: dipendente, ore, nome, retribuzione oraria ...
 - **DIPENDENTE**, gli altri descrivono attributi del dipendente

```
class Employee {  
    ...  
}
```




Retribuzione

- Comportamento desiderato
 - Creare oggetti di tipo Employee (costruttore)
 - `Employee`
 - Calcolare la paga
 - `calcPay`
 - Interrogare un oggetto Employee per conoscere il nome
 - `getName`