

## Espressioni -- 1

- Un'espressione e' una qualsiasi **combinazione di operatori e operandi**, usati con gli opportuni tipi, che assume un valore di qualche tipo
- In generale, gli **operandi** in un'espressione possono essere:
  - i. valori (di qualche tipo), anche restituiti da una chiamata di funzione
  - ii. costanti
  - iii. variabili

## Espressioni -- 2

- Per i tipi numerici, l'operatore di **moltiplicazione** viene indicato con l'asterisco **\*** mentre la **divisione** viene indicata con la barra **/**
- **Esempi** di espressioni numeriche:
  - $\frac{2}{3.7} \Rightarrow 2/3.7$
  - $6(9-3) \Rightarrow 6*(9-3)$
  - $\frac{100-9.99}{10+0.2} \Rightarrow (100 - 9.99)/(10 + 0.2)$
  - `cout << 9*15/5+32; //stampa di una espressione`
- L'operatore di **divisione /** si puo' applicare **sia ad interi che a decimali** (`double` e `float`). Se entrambi gli operandi sono interi, viene eseguita una **divisione intera**, altrimenti viene eseguita una **divisione reale**. Ad esempio,  $1/2$  produce il valore `0`, mentre  $1.0/2$  produce il valore `0.5`

## Espressioni -- 3

- **%** e' l'operatore di **resto di una divisione intera**, detto anche **operatore di modulo**. Esempi:
  - `23%2`  $\Rightarrow$  `1`      `8%8`  $\Rightarrow$  `0`      `81%11`  $\Rightarrow$  `4`
- Valgono le usuali **regole di precedenza** tra operatori: le **parentesi** vengono prima di ogni operatore, le **divisioni e moltiplicazioni** vengono prima di **addizioni e sottrazioni**

## Istruzioni

- Il `main()` consiste di una sequenza di **dichiarazioni di variabili** e di **istruzioni** (o statements)
- Le istruzioni che considereremo si suddividono in quattro classi:
  - 1) **invocazione** di qualche funzionalita` disponibile. Considereremo solamente la funzionalita` di **stampa a video** e **input da tastiera** disponibili dalla libreria del linguaggio.
  - 2) **assegnazione**: **cambia il contenuto di una variabile** con qualche valore dello stesso tipo della variabile
  - 3) **selezione**: in base ad un **test** che puo' essere vero o falso, decide **se eseguire o meno un gruppo di istruzioni**
  - 4) **ripetizione**: **esegue un gruppo di istruzioni piu' volte** iterativamente

## Assegnazione -- 1

- Un'istruzione di assegnazione **permette di dare o cambiare valore** ad una variabile
- Dopo che una variabile e' stata dichiarata, l'istruzione di assegnazione ci permette di **darle un valore**. Lo schema generale di un'assegnazione e' il seguente:
  - **variabile = espressione;**
- Effetto dell'istruzione di assegnazione:
  1. viene **valutata l'espressione**, e
  2. il valore risultante viene **assegnato alla variabile**, ovvero memorizzato nella memoria corrispondente alla variabile. Quindi "=" puo' venire interpretato come "diventa"

## Assegnazione -- 2

- Una **dichiarazione con inizializzazione** simultanea di una variabile:  
`tipo var = val;`
- corrisponde a (cioe', ha gli stessi effetti di) una **dichiarazione con successiva assegnazione**:  
`tipo var; var = val;`
- **In generale, i tipi** dell'espressione e della variabile **devono essere gli stessi**. Ad esempio, non e' possibile assegnare un'espressione di tipo `float` ad una variabile di tipo `int`.
- **Eccezioni.** I **numeri reali comprendono gli interi**. Cio' giustifica la possibilita' di poter assegnare espressioni di tipo `int` ad una variabile di tipo `double` o `float`.
- Esempio: `double tasso = 3.5; int a = 6; tasso = a;`

## Input e Output

### Istruzioni di invocazione

- **Input da tastiera:** si usa l'operatore `>>` sull'oggetto `cin`
- **Esempio:**

```
int x; //dichiaro una variabile x di tipo intero
cin>>x; //da tastiera viene inserito il valore di x
```
- **Output a video** di stringhe, cioe' di testo, e' l'operatore di output `<<` sull'oggetto `cout` che denota lo standard output cioe' il video.
- Quindi l'istruzione  
`cout << "Ciao ragazzi!!" << endl;` ha questo significato: stampa sullo standard output `cout` la stringa "Ciao ragazzi!!" e successivamente il carattere di fine linea `endl`

## Output di variabili e espressioni

- Tramite l'operatore di output `<<` e' possibile **stampare** sul video **variabili ed espressioni** di qualsiasi tipo primitivo.
  - `cout << 4 * 5 - 2; // stampa 18`
  - `double liraEuro = 1936.27; cout << 100 * liraEuro; // stampa 193627.0`
- Come fare **per stampare piu' di un valore** oppure stringhe e numeri assieme? Si usa piu' volte l'**operatore di output a cascata**. Ad esempio:
  - `cout << "Un Euro vale " << liraEuro << " Lire" << endl; // stampa: Un Euro vale 1936.27 Lire`

## Esempio

Il seguente programma **calcola e stampa il voto finale** d'esame di un corso universitario in cui sono previsti **tre compiti** con i seguenti **pesi relativi: 20%, 30% e 50%**.

```
#include<iostream>
using namespace std;

main(){
    int voto1 = 22;
    int voto2 = 28;
    int voto3 = 30;
    int votoFinale = (voto1*20 + voto2*30 + voto3*50)/100;
    // Attenzione: e' una divisione intera
    cout << "Il voto finale e`: " << votoFinale
         << "/30" << endl;
}
// stampa: Il voto finale e`: 27/30
```

## Esempio

Il seguente programma **calcola e stampa il voto finale** d'esame di un corso universitario in cui sono previsti **tre compiti** a seconda dei **pesi relativi inseriti dalla tastiera**.

```
#include<iostream>
using namespace std;

main(){
    int voto1 = 22;
    int voto2 = 28;
    int voto3 = 30;
    int peso1, peso2, peso3;
    cin>> peso1; cin>> peso2; cin>> peso3;
    int votoFinale = (voto1*peso1 + voto2*peso2 +
voto3*peso3)/100;
    // Attenzione: e' una divisione intera
    cout << "Il voto finale e`: " << votoFinale
         << "/30" << endl;
}
// stampa: Il voto finale e`: 27/30
```