

WEB DEVELOPER
Fondamenti di Programmazione
Massimo PAPA

Librerie definite dall'utente

- I MODULI -

Esempio

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

double computeTaxes(double income)
{
    if (income<5000) return 0.0;
    return 0.07*(income-5000.0);
}

double getIncome(string prompt)
{
    double reddito;
    cout << prompt;
    cin >> reddito;
    return reddito;
}

void printTaxes(double tasse)
{
    cout << "The taxes is $" << tasse
    << endl;
}
```

```
int main()
{
    // Get the income;
    double reddito;
    double tasse;

    reddito = getIncome("Please enter
        the employee income: ");

    // Compute Taxes
    tasse = computeTaxes(reddito);

    // Print employee taxes
    printTaxes(tasse);

    return 0;
}
```

Firma della funzione

- La firma della funzione è in realtà simile all'intestazione della funzione tranne per due aspetti:
 - ◆ I nomi dei parametri potrebbero non essere specificati nella firma della funzione
 - ◆ La firma della funzione deve essere terminata con un punto e virgola
- Esempio:

Parametro
senza nome

Punto e
virgola

```
double computeTaxes(double) ;
```

Perché abbiamo bisogno della firma della funzione?

- Per nascondere le informazioni
 - ◆ Se vuoi creare la tua libreria e condividerla con i tuoi clienti senza far conoscere loro i dettagli di implementazione, dovresti dichiarare tutte le firme delle funzioni in un file di intestazione (.h) e distribuire il codice binario del file di implementazione
- Per l'astrazione delle funzioni
 - ◆ Condividendo solo le firme delle funzioni, abbiamo la libertà di modificare i dettagli di implementazione di volta in volta per
 - Migliorare le prestazioni delle funzioni
 - fare in modo che i clienti si concentrino sullo scopo della funzione, non sulla sua implementazione

Esempio con le firme

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

// Function Signature
double getIncome(string);
double computeTaxes(double);
void printTaxes(double);

int main()
{
    // Get the income;
    double reddito;
    double tasse;

    reddito = getIncome("Please enter
        the employee income: ");

    // Compute Taxes
    tasse = computeTaxes(reddito);

    // Print employee taxes
    printTaxes(tasse);

    return 0;
}
```

```
double computeTaxes(double income)
{
    if (income<5000) return 0.0;
    return 0.07*(income-5000.0);
}

double getIncome(string prompt)
{
    double income;
    cout << prompt;
    cin >> income;
    return income;
}

void printTaxes(double taxes)
{
    cout << "The taxes is $" << taxes <<
        endl;
}
```