

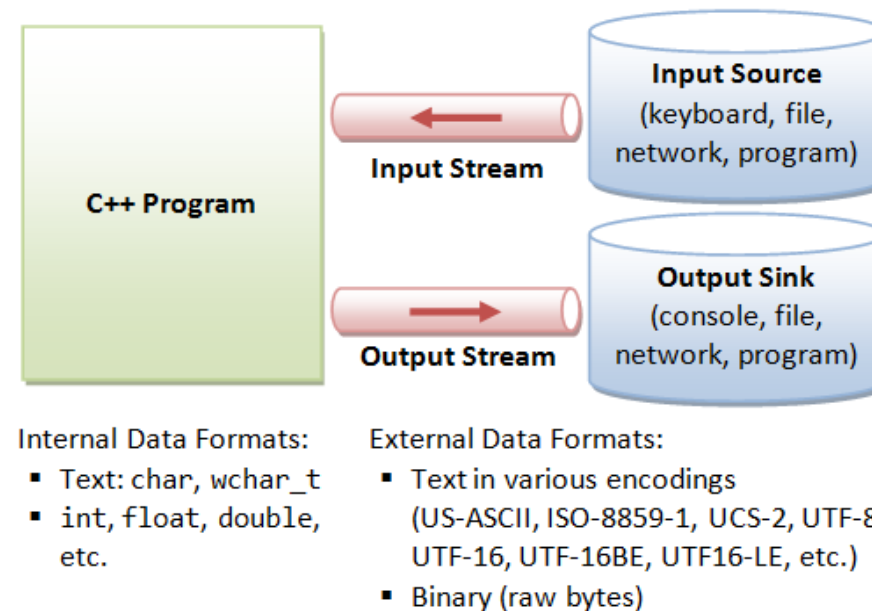


**UNIVERSITÀ
DI PARMA**

C++ input/output

Alberto Ferrari

- l'input/output del C++ è basato sugli stream
- gli stream sono sequenze di byte che rappresentano il flusso in entrata o in uscita di un programma
- gli stream fungono da intermediary fra i programmi e le periferiche di I/O liberando il programmatore dalla necessità di gestire direttamente le periferiche



- nelle operazioni di input i dati vanno da una sorgente di input verso il programma
 - una sorgente di input può essere
 - la tastiera
 - un file
 - una risorsa di rete
 - l'output di un altro programma)

- nelle operazioni di output il flusso di dati che ha come sorgente il programma può essere diretto verso:
 - la console (video)
 - un file
 - la rete
 - un altro programma

- << e >> sono operatori di flusso che operano sugli stream
- << estrazione da uno stream
- >> inserimento in uno stream
- cin, cout (e cerr) sono oggetti che rappresentano stream di input, output ed error standard

```
using namespace std;
int main()
{
    int n;
    cout << "Inserisci un valore intero in rappresentazione base 10: ";
    cin >> dec >> n;
    cout << dec << "rappresentazione decimale: " << n << " esadecimale: " << hex << n << endl;
    cout << dec << "rappresentazione decimale: " << n << " ottale      : " << oct << n << endl;
    cout << "Inserisci un valore intero in rappresentazione base 16: ";
    cin >> hex >> n;
    cout << dec << "rappresentazione decimale: " << n << " esadecimale: " << hex << n << endl;
    cout << dec << "rappresentazione decimale: " << n << " ottale      : " << oct << n << endl;
    cout << "Inserisci un valore intero in rappresentazione base 8: ";
    cin >> oct >> n;
    cout << dec << "rappresentazione decimale: " << n << " esadecimale: " << hex << n << endl;
    cout << dec << "rappresentazione decimale: " << n << " ottale      : " << oct << n << endl;
    cout << "Attenzione l'ultima base impostata rimane attiva!" << endl;
    cout << "Inserisci un valore intero : ";
    cin >> oct >> n;
    cout << dec << "rappresentazione decimale: " << n << " esadecimale: " << hex << n << endl;
    cout << dec << "rappresentazione decimale: " << n << " ottale      : " << oct << n << endl;
    return 0;
}
```

```
Inserisci un valore intero in rappresentazione base 10: 12
rappresentazione decimale: 12 esadecimale: c
rappresentazione decimale: 12 ottale      : 14
Inserisci un valore intero in rappresentazione base 16: 12
rappresentazione decimale: 18 esadecimale: 12
rappresentazione decimale: 18 ottale      : 22
Inserisci un valore intero in rappresentazione base 8: 12
rappresentazione decimale: 10 esadecimale: a
rappresentazione decimale: 10 ottale      : 12
Attenzione l'ultima base impostata rimane attiva!
Inserisci un valore intero : 12
rappresentazione decimale: 10 esadecimale: a
rappresentazione decimale: 10 ottale      : 12
```

- l'operatore << consente di *inserire* un oggetto string in un *output* stream
- es: `string message("Hello");`
`cout << message;`
- l'operatore >> consente di *estrarre* un oggetto string da un *input* stream
- ignora eventuali caratteri di spaziatura iniziali, la lettura si ferma al primo carattere di *spaziatura*
- es: `string s1, s2;`
- `cin >> s1 >> s2;`
- lettura di un'*intera linea* di input con la funzione `getline(aStream, aString)` definita nella libreria string