

Analyse et Programmation Orientées Objets / C++

Gestion des flux standards
(**stdin, stdout, stderr**)

Gestion des E/S via **iostream**

➔ **Besoins et problèmes posés**

- Simplifier les opérations de formatage des données
- Introduire des opérateurs génériques
- Autoriser la surcharge de ces opérateurs
- Homogénéiser les concepts avec ceux du S.E.
- E/S bufferisées
-

Opérateurs implémentés sous forme de fonctions

Contenu de IOSTREAM

➔ Description des classes

- Gestion des mémoires tampons (streambuf)
- Gestion des entrées (**istream**)
- Tous les types de base du langage C sont supportés
- Gestion des sorties (**ostream**)
- Tous les types de base du langage C sont supportés
- Réunion de **istream** et **ostream** dans **iostream**

Opérations sur les flux (1)

➔ Règles d'utilisation

- Flux stdin représenté par le mot clé *cin*
- Flux stdout représenté par le mot clé *cout*
- Opérateurs génériques >> (entrée) et << (sortie)
- Inclure le fichier iostream.h (remplace stdio.h)
- Meta syntaxe générale :

<flux cible> OPR <source ou cible>

Opérations sur les flux (2)

➔ Mise en oeuvre

```
# include <iostream.h>
```

```
void main () {
```

```
    cout << " Yes, we can ! " << endl;
```

```
}
```

Opérations sur les flux (3)

➔ Exemples d'utilisation du flux de sortie

```
int x=12;
```

```
double y=14.75;
```

```
char message[]="Valeur de Y = ";
```

```
cout << "Valeur de X = " << x << endl;
```

```
cout << message << y << endl;
```

Opérations sur les flux (4)

➡ Exemples d'utilisation du flux d'entrée

```
int x;
```

```
double y, z;
```

```
cin >> x;
```

```
cin >> y >> z;
```