Formateo de Stream de Salida

Agustín J. González

Versión original de Kip Irvine

ELO-326: Seminario II

Streams de salida

- La clase ostream es derivada de la clase ios
- cout, cerr son objetos ostream predefinidos
  - cout es “buffereada”, cerr no.
- El operador << inserta caracteres y números en el stream de salida
- Todo lo mostrado aquí funciona también para archivos de salida.

Pinto fijo y Notación científica

- El manipulador fixed fuerza el uso de notación en punto fijo
- El manipulador scientific fuerza notación científica
- Los manipuladores uppercase y nouppercase cambian entre "E" y "e”
- Estos manipuladores son persistentes.

Ejemplo...

Punto fijo y notación científica

```
double X = 123456.12;
cout << fixed
     << X << '\n'
     << scientific << X << '\n'
     << uppercase  << X << endl;

//Output:
123456.120000
1.234561e+005
1.234561E+005
```

Anteponiendo signo

- Los manipuladores showpos y noshowpos determinan si el signo + es o no desplegado antes del número

```
cout << showpos << 123 << '\n'
     << noshowpos << 123 << '\n';

//Output:
+123
123
```

Desplegando Precisión

- showpoint fuerza el despliegue con punto decimal
- setprecision(n) especifica que *n* dígitos deben ser desplegados a la derecha del punto decimal.
- Estas definiciones son persistentes.

Ejemplo...

Ejemplo: Precisión

```
double Z = 21.2351;
cout << fixed << setprecision(2)
     << Z << "\n"
     << setprecision(4) << Z;

//Output:
21.24                  // rounds upward
21.2351
```

Despliegue de Valores Booleanos

Los manipuladores boolalpha y noboolalpha definen si las expresiones booleanas son desplegadas como 0/1 o false/true.

```
cout << true << '\n'
     << boolalpha << true << '\n'
     << noboolalpha << true << '\n';

//Output:
1
true
1
```

Definiendo el Ancho del Campo

- El manipulador setw(n) nos permite definir el ancho de despliegue de la siguiente expresión a ser desplegada . Éste no es persistente.
- EL manipulador setfill(ch) nos permite definir el carácter por defecto de relleno para números (el valor por defecto es el espcaio).
- Por ejemplo, usamos setfill('*') para imprimir cheques.
- Ejemplo...

Ejemplo: setw()

```
double X = 123.45;
cout << setw(10) << X << '\n'
     << setfill('*')
     << setw(10) << X << '\n';

//Output:
    123.45
****123.45
```

Left y Right

- El manipulador left justifica la salida hacia la izquierda (es el valor por defecto para strings)
- El manipulador right justifica la salida hacia la derecha (es el predefinido para números).

Ejemplo...

Left y Right

```
cout << left << setw(10) << 123
     << "*\n"
     << right << setw(10) << 123
     << "*\n";

//Output:
123       *
       123*
```

Despliegue en distinta base

- Los manipuladores dec, hex, y oct cambian la base del número a desplegar a 10, 16 y 8 respectivamente.
- Los manipuladores showbase y noshowbase nos permiten definir si la base de los números será desplegada.

- Ejemplo...

Ejemplo: *showbase*

```
cout << showbase << uppercase
        << dec << 255 << '\n'
        << hex << 255 << '\n'
        << oct << 255 << '\n';

//Output:
255
0XFF    (without uppercase, would be 0xff)
0377
```

Fin