**Unidad 2 – Manejo de Buffer de Teclado   
Escanear una cadena con C++ y getline**

Cuando se utiliza cin y se encuentra un espacio o separador en lo que el usuario introduzca, C++ infiere que son variables separadas. Es decir, si se escribe “Jose Luis ” no se toma el nombre completo, sino que se toma una variable como “jose” y otra como “luis”.

Es por eso que para escanear o solicitar una cadena en C++ no se usa el flujo de cin, mejor se utiliza la función getline.

#include <iostream>

using namespace std;

int main() {

string nombre = "";

cout << "Escribe tu nombre: \n";

getline(cin, nombre);

cout << "Hola, " << nombre;

}

En este caso usamos getline, que va a escanear lo que haya dentro del stream o flujo (cin) dentro de la variable nombre, **hasta que se encuentre el carácter de nueva línea \n** o el carácter que se especifique como tercer argumento a getline.

Código que reproduce el error:

#include <iostream>

// Lectura cin vs getline()

using namespace std;

int main() {

string nombre = "";

cout << "Escribe tus nombres: \n";

cin >> nombre;

cout << "Hola, " << nombre << endl;

cout << "\n";

// vemos ahora la diferencia con getline

cout << "ahora con getline ;-) \n";

cout << "Escribe tu nombre te lo leo con getline: \n";

getline(cin, nombre);

cout << "Hola, " << nombre;

cout << "\n";

cout << "\n";

cout << "Pero que a pasado aqui? \n";

cout << "Como el Buffer (memoria del teclado) quedo cargada por que trunco el segundo nombre al encontrar un espacio en Blanco, ese valor que quedo despues del espacio en blanco del primer nombre, lo guardo para la proxima variable" ;

cout << ", debemos hacer un cin.Sync para que borre el Buffer, Pero que ha pasado aqui otra vez? \n";

cin.sync();

cout << "\n";

cout << "Escribe tu nombre te lo leo con getline: \n";

getline(cin, nombre);

cout << "\n";

cout << "Hola (con cin.sync y luego getline) vuala!!! \n" ;

cout << " Tus nombres: " << nombre;

cout << "\n";

}

Apéndice Leer enteros

Leer un entero con C++

Para leer un entero primero declaramos la variable, y después redirigimos el flujo de cin hacia nuestra variable. Más tarde, nuestra variable tendrá el valor que el usuario haya introducido por teclado.

Veamos el siguiente ejemplo que solicita la edad de una persona:

#include <iostream>

using namespace std;

int main() {

int edad = 0;

cout << "Escribe tu edad: \n";

cin >> edad;

cout << "Tu edad es " << edad;

}

Primero desplegamos un mensaje para que se le solicite a una persona su edad, redirigimos lo que escriba hacia la variable edad y después de eso ya podemos imprimirla.

Por cierto, con esto tenemos la seguridad de que la variable será entera (suponiendo que el usuario escribe un número válido) y no una cadena.