

Processing strings using std::istringstream

Last Updated: 07-05-2020

The **std::istringstream** is a string class object which is used to **stream the string** into different variables and similarly files can be stream into strings. Objects of this class use a string buffer that contains a sequence of characters. This sequence of characters can be accessed as a string object.

Header File:

```
#include <sstream>
```

1. Streaming integer from a string with std::istringstream

One way to stream a string is to use an input string stream object **std::istringstream** from the header. Once a **std::istringstream** object has been created, then the string can be streamed and stored using the **extraction operator(>>)**. The extraction operator will read until whitespace is reached or until the stream fails.

Below is the illustration of the **std::istringstream**:

```
// C++ program to illustrate std::istringstream
#include <iostream>
#include <sstream>
#include <string>
using std::istringstream;
using std::string;
using std::cout;

// Driver Code
int main()
{
    // Input string
    string a("1 2 3");

    // Object class of istringstream
    istringstream my_stream(a);
```

```

// Variable to store number n
int n;

// Stream a number till white space
// is encountered
my_stream >> n;

// Print the number
cout << n << "\n";
}

```

Output:

1

The **std::istringstream** object can also be used as a boolean to determine if the last extraction operation failed. This happens if there wasn't any more of the string to the stream, **For Example**, If the stream still has more characters then we are able to stream the string again.

The extraction operator >> writes the stream to the variable on the right of the operator and returns the **std::istringstream** object, so the entire expression **my_stream >> n** is a std::istringstream object which returns a boolean i.e., true if stream is possible else return false.

Below is the implementation of using the **std::istringstream** in the above way:

Type 1

```

// C++ program to illustrate std::istringstream
#include <iostream>
#include <sstream>
#include <string>
using std::istringstream;
using std::string;
using std::cout;

// Driver Code
int main()
{
    // Input string
    string a("1 2 3");

    // Object class of istringstream
    istringstream my_stream(a);

    // Variable to store number n
    int n;

    // Testing to see if the stream was
    // successful and printing results.
    while (my_stream) {

        // Streaming untill space is
        // encountered
        my_stream >> n;
    }
}

```

```

    // If my_stream is not empty
    if (my_stream) {
        cout << "That stream was successful: "
              << n << "\n";
    }

    // Else print not successful
    else {
        cout << "That stream was NOT successful!"
              << "\n";
    }
}

return 0;
}

```

Type 2

```

// C++ program to illustrate std::istringstream
#include <iostream>
#include <sstream>
#include <string>
using std::istringstream;
using std::string;
using std::cout;

// Driver Code
int main()
{
    // Input string
    string a("1 2 3");

    // Object class of istringstream
    istringstream my_stream(a);

    // Variable to store number n
    int n;

    // Testing to see if the stream was
    // successful and printing results.
    while (my_stream >> n) {

        cout << "That stream was successful: "
              << n << "\n";
    }

    cout << "That stream was NOT successful!"
          << "\n";
    return 0;
}

```

Output:

That stream was successful: 1
That stream was successful: 2
That stream was successful: 3
That stream was NOT successful!

2. Strings with Mixed Types

In the above illustrations, the string contains only whitespaces and characters which could be converted to int. If the string has mixed types i.e., it contains more than one data type in the stream then it can be used as illustrated below.

Below is the illustration of the **std::istringstream** for the mixed types:

Program 1:

```
// C++ program to illustrate std::istringstream
// when string has integer followed by character
#include <iostream>
#include <sstream>
#include <string>
using std::istringstream;
using std::string;
using std::cout;

// Driver Code
int main()
{
    // Input string
    string str("1, 2, 3");

    // Object class of istringstream
    istringstream my_stream(str);

    // Variable to store the number n
    // and character ch
    char c;
    int n;

    // Traverse till input stream is valid
    while (my_stream >> n >> c) {

        cout << "That stream was successful: "
              << n << " " << c << "\n";
    }
    cout << "The stream has failed."
          << "\n";

    return 0;
}
```

Output:

That stream was successful: 1,
That stream was successful: 2,
The stream has failed.

Program 2:

```
// C++ program to illustrate std::istringstream
// to tokenize the string
#include <iostream>
#include <sstream>
#include <string>
using std::istringstream;
using std::string;
using std::cout;

// Driver Code
int main()
{
    // Input string
    string str("abc, def,   ghi");

    // Object class of istringstream
    istringstream my_stream(str);

    // To store the stream string
    string token;

    size_t pos = -1;

    // Traverse till stream is valid
    while (my_stream >> token) {

        // If ',' is found then tokenize
        // the string token
        while ((pos = token.rfind(','))
               != std::string::npos) {
            token.erase(pos, 1);
        }

        // Print the tokenize string
        cout << token << '\n';
    }
}
```

Output:

```
abc
def
ghi
```

Reference: <http://www.cplusplus.com/reference/ssstream/istringstream/>

Rated as one of the most sought after skills in the industry, own the basics of coding with our **C++ STL** Course and master the very concepts by intense problem-solving.