Tutorial 2: Data Input/Output

Overview

- Streams
- Standard I/O Reference
- Connect Input Stream in Visual Studio 2010

References

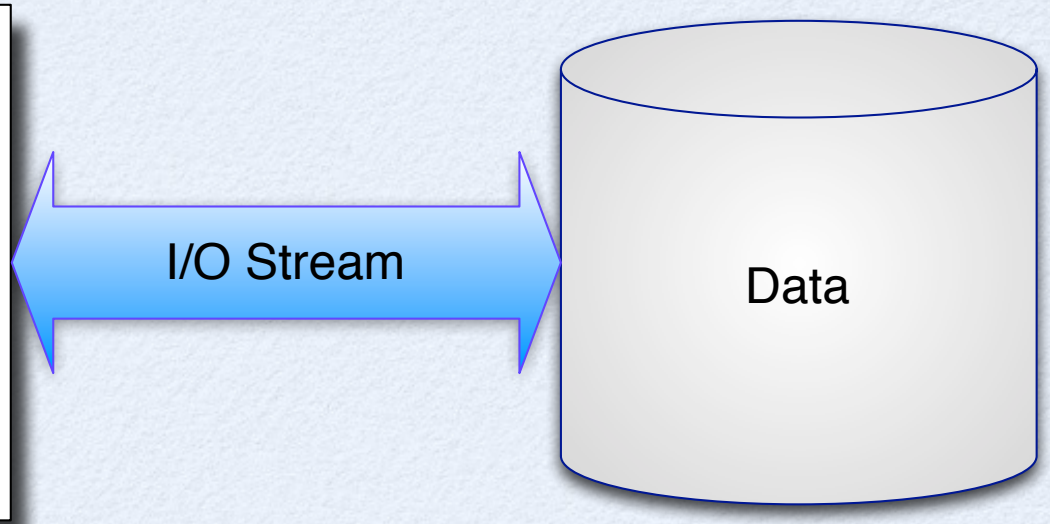
- Gary J. Bronson: C++ for Engineers and Scientists. 3rd Edition. Thomson (2010)
- Stanley B. Lippman, Josée Lajoie, and Barbara E. Moo: C++ Primer. 4th Edition. Addison-Wesley (2006)
- Bruno R. Preiss: Data Structures and Algorithms with Object-Oriented Design Patterns in C++. John Wiley & Sons, Inc. (1999)
- Gary J. Bronson: Object-Oriented Program Development Using C++ - A Class-Centered Approach. Thomson (2006)

I/O Streams

```
#include <iostream>

int main()
{


    return 0;
}
```



- In C++, input or output, independent of the type of I/O medium, are mapped into logical data streams with common properties.
- Two forms of mapping are supported: text streams and binary streams.

I/O Media

- Streams can be associated with
  - Physical devices (e.g., console - cin, cout)
  - Files (e.g., coefficients.txt, sales.dbf)
  - Structured storage (e.g., int values[10])

A Program that uses the C++ I/O library.

SimpleIO.cpp

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Enter two numbers:" << endl;
    int v1, v2;
    cin >> v1 >> v2;
    cout << "The sum of " << v1 << " and " << v2
         << " is " << v1 + v2 << endl;
    return 0;
}
```

Line: 14 Column: 1 C++ Tab Size: 4 main

C:\WINDOWS\system32\cmd.exe

```
Enter two numbers:
7
12
The sum of 7 and 12 is 19
Press any key to continue . . .
```

- cin and cout represent the standard input stream and the standard output stream in every C++ program.

The Standard Input Stream cin

- cin is an object of class istream that represents the standard input stream. It corresponds to stdin in C.
- cin is a globally visible object that is readily available to any C++ compilation unit (i.e., a .cpp-file) that includes the library iostream.
- cin receives input either from the keyboard or a stream associated with the standard input stream.
- We use the operator >> to fetch formatted data or use the methods read of get to retrieve unformatted data from the standard input stream.

The Standard Output Stream cout

- cout is an object of class ostream that represents the standard output stream. It corresponds to stdout in C.
- cout is a globally visible object that is readily available to any C++ compilation unit (i.e., a .cpp-file) that includes the library iostream.
- cout sends data either to the console (as text) or a stream associated with the standard output stream.
- We use the operator << to push formatted data or use the methods write or put to send unformatted data to the standard output stream.

Redirect I/O

program < input > output

input - Notepad

File Edit Format View Help

```
7
12
```

```
1:59 PM    <DIR>          .
1:59 PM    <DIR>          ..
3:15 PM                 7 input.txt
4:12 PM            38,912 SimpleIO.exe
4:12 PM           380,976 SimpleIO.ilk
4:12 PM           560,128 SimpleIO.pdb
               4 File(s)    980,023 bytes
               2 Dir(s)   3,242,401,792 bytes free

C:\Uni\HIT3303\SimpleIO\Debug>SimpleIO.exe < input.txt > output.txt

C:\Uni\HIT3303\SimpleIO\Debug>dir
 Volume in drive C has no label.
 Volume Serial Number is 1075-0DEA

 Directory of C:\Uni\HIT3303\SimpleIO\Debug

03/08/2013  02:00 PM    <DIR>          .
03/08/2013  02:00 PM    <DIR>          ..
08/20/2012  03:15 PM                 7 input.txt
03/08/2013  02:00 PM                47 output.txt
02/25/2013  04:12 PM
02/25/2013  04:12 PM
02/25/2013  04:12 PM
               5 File(s)
               2 Dir(s)

C:\Uni\HIT3303\SimpleIO\De
```

output - Notepad

File Edit Format View Help

```
Enter two numbers:
The sum of 7 and 12
```

Chaining Input and Output

output - Notepad

File Edit Format View Help

```
Enter two numbers:
The sum of 7 and 12
```

```
#include <iostream>
using namespace std;

int main()
{
    cout << "Enter two numbers:" << endl;
    int v1, v2;
    cin >> v1 >> v2;
    cout << "The sum of " << v1 << " and " << v2
         << " is " << v1 + v2 << endl;
    return 0;
}
```

14 Column: 1 Tab Size: 4 main

input - Notepad

File Edit Format View Help

```
7
12
```

Standard Streams Summary

- In C++, there are three standard text streams:
  - cin - text input stream
  - cout - text output stream
  - cerr - secondary error text output stream

Input Stream Variations

- istream:
  - istream objects can be used to read and interpret input from sequences of characters. istream provides functions to perform input operations divided in two main groups: formatted input and unformatted (raw) input.
- ifstream:
  - ifstream provides an interface to read data from files as input streams. objects of type ifstream maintain private memory to perform buffered I/O.
- istringstream:
  - istringstream provides an interface to manipulate strings as input streams. istringstream objects provide a memory stream to exchange data between two stream-based data endpoints.

Output Stream Variations

- ostream:
  - ostream objects can be used to write and format output as sequences of characters. ostream provides functions to perform output operations divided in two main groups: formatted output and unformatted (raw) output.
- ofstream:
  - ofstream provides an interface to write data to files as output streams. objects of type ofstream maintain private memory to perform buffered I/O.
- ostringstream:
  - ostringstream provides an interface to manipulate strings as output streams. ostringstream objects provide a memory stream to exchange data between two stream-based data endpoints.

Reference - C++ Reference

www.cplusplus.com/reference/

The T-Syste...are Archive | German <->...U Chemnitz | Apple | Amazon | eBay | Yahoo! | News | Blackboard ...demic Suite

cplusplus.com | Search: Go | Not logged in | register | log in

Reference

C++

- Information
- Documentation
- Reference
- Articles
- Forum

Reference

- *C library:*
- *Containers:*
- *Input/Output:*
- *Other:*

Reference

Reference of the C++ Language Library, with detailed descriptions of its elements and examples on how to use its functions

The standard C++ library is a collection of functions, constants, classes, objects and templates that extends the C++ language providing basic functionality to perform several tasks, like classes to interact with the operating system, data containers, manipulators to operate with them and algorithms commonly needed.

It can be divided into:

C Library

The elements of the C language library are also included as a subset of the C++ Standard library. These cover many aspects, from general utility functions and macros to input/output functions and dynamic memory management functions:

<cassert> (assert.h)	C Diagnostics Library (header)
<cctype> (ctype.h)	Character handling functions (header)
<cerrno> (errno.h)	C Errors (header)
<cfloat> (float.h)	Characteristics of floating-point types (header)
<ciso646> (iso646.h)	ISO 646 Alternative operator spellings (header)
<climits> (limits.h)	Sizes of integral types (header)
<clocale> (locale.h)	C localization library (header)
<cmath> (math.h)	C numerics library (header)
<codecvt>	Unicode conversion facets (header)
<csetjmp> (setjmp.h)	Non local jumps (header)
<csignal> (signal.h)	C library to handle signals (header)

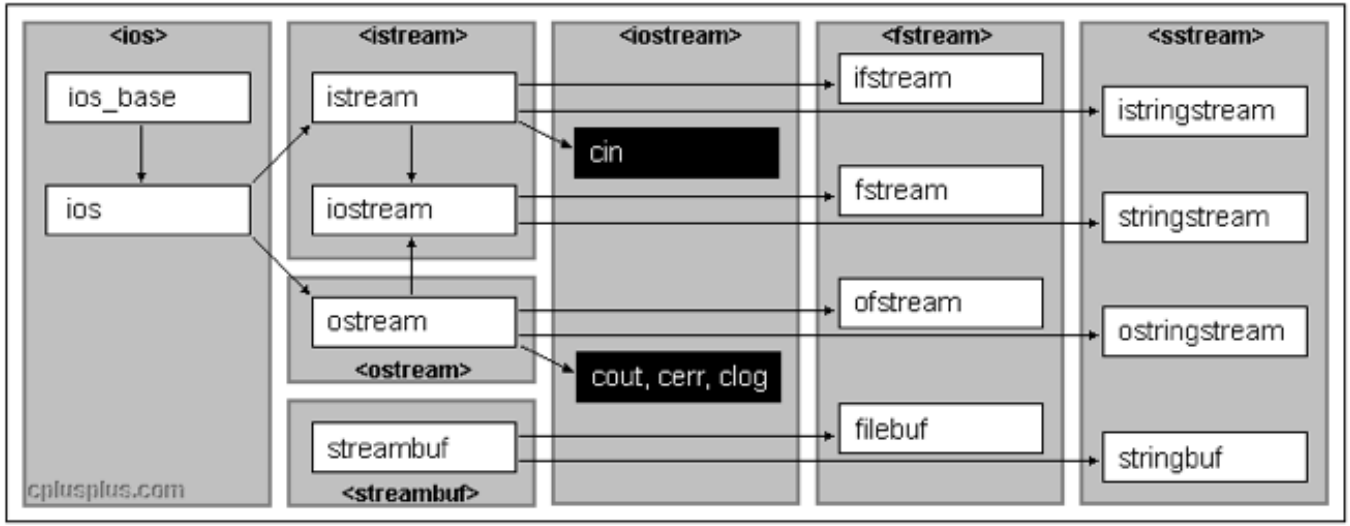
Reference - C++ Reference

www.cplusplus.com/reference/

<unordered_map>	Unordered map header (header)
<unordered_set>	Unordered set header (header)
<vector>	Vector header (header)

Input/Output Stream Library

Provides functionality to use an abstraction called *streams* specially designed to perform input and output operations on sequences of character, like files or strings.
This functionality is provided through several related classes, as shown in the following relationship map, with the corresponding header file names on top:



Atomics and threading library

[under construction]

Miscellaneous headers

<algorithm>	Standard Template Library: Algorithms (library)
<chrono>	Time library (header)
<complex>	Complex numbers library (header)
<exception>	Standard exceptions (header)

istream - C++ Reference

www.cplusplus.com/reference/istream/istream/

The T-Syste...are Archive | German <->...U Chemnitz | Apple | Amazon | eBay | Yahoo! | News | Blackboard ...demic Suite

Reference

- *C library:*
- *Containers:*
- *Input/Output:*
  - <fstream>
  - <iomanip>
  - <ios>
  - <iosfwd>
  - <iostream>
  - <istream>
  - <ostream>
  - <sstream>
  - <streambuf>
- *Other:*

<istream>

- *class templates:*
  - basic_iostream
  - basic_istream
- *classes:*
  - iostream
  - istream
  - wiostream
  - wistream
- *manipulators:*
  - ws

istream

- istream::istream
- istream::~istream
- *member classes:*
  - istream::sentry
- *member functions:*
  - istream::gcount
  - istream::get
  - istream::getline
  - istream::ignore
  - istream::operator>>
  - istream::peek
  - istream::putback
  - istream::read
  - istream::readsome

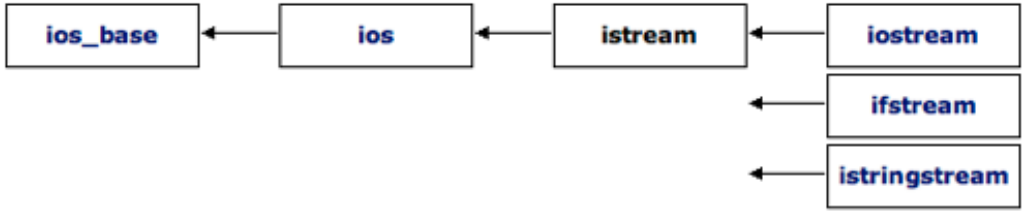
class

std::istream

<istream> <iostream>

```
typedef basic_istream<char> istream;
```

Input stream



Input stream objects can read and interpret input from sequences of characters. Specific members are provided to perform these input operations (see functions below).

The standard object `cin` is an object of this type.

This is an instantiation of `basic_istream` with the following template parameters:

template parameter	definition	comments
charT	char	Aliased as member char_type
traits	char_traits<char>	Aliased as member traits_type

Objects of these classes keep a set of internal fields inherited from `ios_base` and `ios`:

	field	member functions	description
Formatting	format flags	flags setf unsetf	A set of internal flags that affect how certain input/output operations are interpreted or generated. See member type fmtflags.
	field width	width	Width of the next formatted element to insert.
	display precision	precision	Decimal precision for the next floating-point value inserted.
	locale	getloc imbue	The locale object used by the function for formatted input/output operations affected by localization properties.
	fill character	fill	Character to pad a formatted field up to the *field width* (width).
State	error state	rdstate setstate clear	The current error state of the stream. Individual values may be obtained by calling good, eof, fail and bad. See member type iostate.
	exception mask	exceptions	The state flags for which a failure exception is thrown. See member type iostate.

istream::operator>> - C++ Reference

www.cplusplus.com/reference/istream/istream/operator%3E%3E/

The T-Syste...are Archive | German <->...U Chemnitz | Apple | Amazon | eBay | Yahoo! | News | Blackboard ...demic Suite

Reference

- *C library:*
- *Containers:*
- *Input/Output:*
  - <fstream>
  - <iomanip>
  - <ios>
  - <iosfwd>
  - <iostream>
  - <istream>
  - <ostream>
  - <sstream>
  - <streambuf>
- *Other:*

<istream>

- *class templates:*
  - basic_iostream
  - basic_istream
- *classes:*
  - iostream
  - istream
  - wiostream
  - wistream
- *manipulators:*
  - ws

istream

- istream::istream
- istream::~istream
- *member classes:*
  - istream::sentry
- *member functions:*
  - istream::gcount
  - istream::get
  - istream::getline
  - istream::ignore
  - istream::operator>>
  - istream::peek
  - istream::putback
  - istream::read
  - istream::readsome

public member function

std::istream::operator>>

<istream> <iostream>

C++98 | C++11

arithmetic types (1)	`istream& operator>> (bool& val);`
	`istream& operator>> (short& val);`
	`istream& operator>> (unsigned short& val);`
	`istream& operator>> (int& val);`
	`istream& operator>> (unsigned int& val);`
	`istream& operator>> (long& val);`
	`istream& operator>> (unsigned long& val);`
	`istream& operator>> (float& val);`
	`istream& operator>> (double& val);`
	`istream& operator>> (long double& val);`
	`istream& operator>> (void*& val);`

stream buffers (2)	`istream& operator>> (streambuf* sb );`

manipulators (3)	`istream& operator>> (istream& (*pf)(istream&));`
	`istream& operator>> (ios& (*pf)(ios&));`
	`istream& operator>> (ios_base& (*pf)(ios_base&));`

Extract formatted input

This operator (>>) applied to an input stream is known as *extraction operator*:

(1) arithmetic types

Extracts and parses characters sequentially from the stream to interpret them as the representation of a value of the proper type, which is stored as the value of *val*.
Internally, the function accesses the input sequence by first constructing a `sentry` object (with *noskipws* set to `false`). Then (if `good`), it calls `num_get::get` (using the stream's *selected locale*) to perform both the extraction and the parsing operations, adjusting the stream's *internal state flags* accordingly. Finally, it destroys the `sentry` object before returning.

(2) stream buffers

Extracts as many characters as possible from the stream and inserts them into the output sequence controlled by the *stream buffer* object pointed by *sb* (if any), until either the input sequence is exhausted or the function fails to insert into the object pointed by *sb*.

C++98 | C++11

The function is considered to perform *formatted input*: Internally, the function accesses the input sequence by first constructing a `sentry` object (with *noskipws* set to **`false`**). Then (if `good`), it extracts characters from its associated *stream buffer* object as if calling its member functions `sbumpc` or `sgetc`, and finally destroys the `sentry` object before returning.

(3) manipulators

Calls `pf(*this)`, where *pf* may be a *manipulator*.
Manipulators are functions specifically designed to be called when used with this operator.

failbit	appropriate type. For *(2)*, it is set when no characters are inserted in the object pointed by *sb*, or when *sb* is a *null pointer*.
badbit	Error on stream (such as when this function catches an exception thrown by an internal operation). When set, the integrity of the stream may have been affected.

Multiple flags may be set by a single operation.

If the operation sets an *internal state flag* that was registered with member `exceptions`, the function throws an exception of member type `failure`.

Example

```
// example on extraction
#include <iostream>     // std::cin, std::cout, std::hex

int main () {
  int n;

  std::cout << "Enter a number: ";
  std::cin >> n;
  std::cout << "You have entered: " << n << '\n';

  std::cout << "Enter a hexadecimal number: ";
  std::cin >> std::hex >> n;          // manipulator
  std::cout << "Its decimal equivalent is: " << n << '\n';

  return 0;
}
```

This example demonstrates the use of some of the overloaded `operator>>` functions shown above using the standard `istream` object `cin`.

Data races

Modifies *val* or the object pointed by *sb*.
Modifies the stream object.
Concurrent access to the same stream object may cause data races, except for the standard stream object `cin` when this is *synchronized with stdio* (in this case, no data races are initiated, although no guarantees are given on the order in which extracted characters are attributed to threads).

Exception safety

Basic guarantee: if an exception is thrown, the object is in a valid state.

IO manipulators

C++
Information
Documentation
Reference
Articles
Forum

Reference
C library:
Containers:
Input/Output:
<fstream>
<iomanip>
<ios>
<iosfwd>
<iostream>
<istream>
<ostream>
<sstream>
<streambuf>
Other:

<iomanip>
get_money
get_time
put_money
put_time
resetiosflags
setbase
setfill
setiosflags
setprecision
setw

header

<iomanip>

IO Manipulators

Header providing parametric manipulators:

fx Parameterized manipulators

setiosflags	Set format flags (function)
resetiosflags	Reset format flags (function)
setbase	Set basefield flag (function)
setfill	Set fill character (function)
setprecision	Set decimal precision (function)
setw	Set field width (function)
get_money C++11	Get monetary value (function)
put_money C++11	Put monetary value (function)
get_time C++11	Get date and time (function)
put_time C++11	Put date and time (function)

Notice that non-parametric manipulators are declared directly in <ios>.

setw - C++ Reference

www.cplusplus.com/reference/iomanip/setw/

Reader

The T-Syste...are Archive | German <->...U Chemnitz | Apple | Amazon | eBay | Yahoo! | News | Blackboard ...demic Suite

Reference

- *C library:*
- *Containers:*
- *Input/Output:*
  - <fstream>
  - <iomanip>
  - <ios>
  - <iosfwd>
  - <iostream>
  - <istream>
  - <ostream>
  - <sstream>
  - <streambuf>
- *Other:*

<iomanip>

- get_money
- get_time
- put_money
- put_time
- resetiosflags
- setbase
- setfill
- setiosflags
- setprecision
- setw

function

std::setw

<iomanip>

```
/*undefined*/ setw (int n);
```

Set field width

Sets the *field width* to be used on output operations.

Behaves as if member `width` were called with *n* as argument on the stream on which it is inserted/extracted as a manipulator (it can be inserted/extracted on *input streams* or *output streams*).

This manipulator is declared in header `<iomanip>`.

Parameters

n
: Number of characters to be used as field width.

Return Value

Unspecified. This function should only be used as a stream manipulator (see example).

Example

```
// setw example
#include <iostream>     // std::cout, std::endl
#include <iomanip>      // std::setw

int main () {
  std::cout << std::setw(10);
  std::cout << 77 << std::endl;
  return 0;
}
```

Output:

```
        77
```

Data races

The stream object on which it is inserted/extracted is modified.
Concurrent access to the same stream object may introduce data races.

showpos	Show positive signs (function)
skipws	Skip whitespaces (function)
unitbuf	Flush buffer after insertions (function)
uppercase	Generate upper-case letters (function)

Independent flags (switch off):

noboolalpha	No alphanumerical bool values (function)
noshowbase	Do not show numerical base prefixes (function)
noshowpoint	Do not show decimal point (function)
noshowpos	Do not show positive signs (function)
noskipws	Do not skip whitespaces (function)
nounitbuf	Do not force flushes after insertions (function)
nouppercase	Do not generate upper case letters (function)

Numerical base format flags ("basefield" flags):

dec	Use decimal base (function)
hex	Use hexadecimal base (function)
oct	Use octal base (function)

Floating-point format flags ("floatfield" flags):

fixed	Use fixed floating-point notation (function)
scientific	Use scientific floating-point notation (function)

Adustment format flags ("adjustfield" flags):

internal	Adjust field by inserting characters at an internal position (function)
left	Adjust output to the left (function)
right	Adjust output to the right (function)

fx Other functions

iostream_category C++11	Return iostream category (function)

Notice that not all standard manipulators are defined in this header. Input streams also support `ws`, and output streams `endl`, `ends` and `flush`. Streams also support an additional set of manipulators, which are parametric and defined apart in header <iomanip>. These are: setiosflags, resetiosflags, setbase, setfill, setprecision, setw

hex – C++ Reference

www.cplusplus.com/reference/ios/hex/

The T-Syste...are Archive | German <->...U Chemnitz | Apple | Amazon | eBay | Yahoo! | News | Blackboard ...demic Suite

Reference

- *C library:*
- *Containers:*
- *Input/Output:*
  - <fstream>
  - <iomanip>
  - <ios>
  - <iosfwd>
  - <iostream>
  - <istream>
  - <ostream>
  - <sstream>
  - <streambuf>
- *Other:*

<ios>

- *types:*
  - basic_ios
  - fpos
  - ios
  - ios_base
  - io_errc
  - streamoff
  - streampos
  - streamsize
  - wios
  - wstreampos
- *manipulators:*
  - boolalpha
  - dec
  - defaultfloat
  - fixed
  - hex
  - hexfloat
  - internal
  - left
  - noboolalpha
  - noshowbase
  - noshowpoint
  - noshowpos
  - noskipws
  - nounitbuf

function

std::hex

<ios> <iostream>

```
ios_base& hex (ios_base& str);
```

Use hexadecimal base

Sets the `basefield` format flag for the *str* stream to `hex`.

When `basefield` is set to `hex`, integral numerical values inserted into the stream are expressed in hexadecimal base (i.e., radix 16). For input streams, extracted values are also expected to be expressed in hexadecimal base when this flag is set.

The `basefield` format flag can take any of the following values (each with its own manipulator):

flag value	effect when set
dec	read/write integral values using decimal base format.
hex	read/write integral values using hexadecimal base format.
oct	read/write integral values using octal base format.

Notice that the `basefield` flag only affects the insertion/extraction of integer values (floating-point values are always interpreted in decimal base).

Notice also that no base prefix is implicitly prepended to the number unless the `showbase` format flag is set.

For standard streams, the `basefield` flag is set to `dec` on initialization.

Parameters

`str`

Stream object whose `basefield` *format flag* is affected.
Because this function is a manipulator, it is designed to be used alone with no arguments in conjunction with the *insertion* (<<) and *extraction* (>>) operations on streams (see example below).

Return Value

Argument *str*.

Example

```
// modify basefield
#include <iostream>     // std::cout, std::dec, std::hex, std::oct
```

CharacterCounter - Microsoft Visual Studio

File Edit View Project Build Debug Team Data Tools Test Analyze Window Help

Debug | Win32

Solution Explorer

- Solution 'CharacterCounter' (1 pr
- CharacterCounter
  - External Dependencies
  - Header Files
    - CharacterCounter.h
  - Resource Files
  - Source Files

CharacterCounter.h

(Global Scope)

```
#pragma once

#include <iostream>

class CharacterCounter
{
private:
    int fTotalNumberOfCharacters;
    int fCharacterCounts[256];

public:
  CharacterCounter();

  void count( int aCharacter );      // We count all 256 byte values

  friend std::ostream& operator<<( std::ostream& aOStream, const CharacterCounter&
};
```

100 %

Output

Show output from: Build

Code Definition Window | Output | Error List

Item(s) Saved Ln 19 Col 1 Ch 1 INS

CharacterCounter - Microsoft Visual Studio

File Edit View Project Build Debug Team Data Tools Test Analyze Window Help

Debug | Win32

Solution Explorer

Solution 'Char
Charact
Exter
Head
Reso
Sourc

CharacterCounter Property Pages

Configuration: Active(Debug) | Platform: Active(Win32) | Configuration Manager...

- Common Properties
- Configuration Properties
  - General
  - Debugging
  - VC++ Directories
  - C/C++
  - Linker
  - Manifest Tool
  - XML Document Generator
  - Browse Information
  - Build Events
  - Custom Build Step
  - Code Analysis

Debugger to launch:

Local Windows Debugger

Command	$(TargetPath)
Command Arguments	< **$(ProjectDir)/Main.cpp**
Working Directory	$(ProjectDir)
Attach	No
Debugger Type	Auto
Environment	
Merge Environment	Yes
SQL Debugging	No

Command
The debug command to execute.

OK | Cancel | Apply

Toolbox

S... C... P... T... Code Definition Window Output Error List

Ready Ln 54 Col 1 Ch 1 INS

CharacterCounter - Microsoft Visual Studio

File Edit View Project Build Debug Team Data Tools Test Analyze Window Help

Debug | Win32

Solution Explorer

- Solution 'CharacterCounter' (1 pr
- CharacterCounter
  - External Dependencies
  - Header Files
    - CharacterCou
  - Resource Files
  - Source Files
    - Main.cpp

Main.cpp | CharacterCounter.h

(Global Scope)

```
#include "CharacterCounter.h"

using namespace std;
```

C:\WINDOWS\system32\cmd.exe

```
Character counts for 713 characters:
!:      1
":      8
#:      1
&:      3
(:      10
):      10
+:      8
,:      1
.:      5
0:      6
2:      2
5:      2
6:      2
::      6
;:      17
<:      22
=:      5
>:      2
C:      40
N:      3
O:      7
S:      4
T:      3
[:      4
```

Code Definition Window | Output | Error List

Ready | Ln 54 | Col 1 | Ch 1 | INS