getline (string) in C++

getline() is a standard library function in C++ and is used to read a string or a line from an input stream. It is present in the <string> header.

**Prototype Syntax**:

1. **Syntax 1:**

istream& getline (istream& is, string& str, char delim);

**Parameters:**  
**is :**It is an object of istream class and tells the function about the stream from where to read the input from.  
**str :** It is a string object, the input is stored in this object after being read from the stream.  
**delim :** It is the delimitation character which tells the function to stop reading further input after reaching this character.

1. **Syntax 2:**

istream& getline (istream& is, string& str);

The second declaration is almost same as that of the first one. The only difference is, it does not accept any delimitation character. This function consider new line or (‘\n’) character as the delimitation character.

So basically, what the getline function does is extracts characters from the input stream and appends it to the string object until the delimiting character is encountered.  
**Note**: The previous stored value in the string object *str* will be replaced by the input string if any.

**Use of getline() after cin:**

#include <iostream>

#include <string>

using namespace std;

int main() {

int num =0;

string line;

cin>>num;

while(num > 0)

{

getline(cin, line); // user to enter a line of string having multiple words

cout<<line<<endl;

num--;

}

}

Input:

2

I am Saurav Kumar

Bangalore is hub for IT secotor

Output:

I am Saurav Kumar

Remark:

In output, first line is new line character (A blank line).

**Issue Description:**

cin >> leaves the newline char in the input stream. Then getline finds that newline and therefore an empty string is read. To prevent this, use cin.ignore() after cin >>. cin.ignore discards a number of chars in the stream until a delimiter is found. The arguments you should use are cin.ignore(numeric\_limits<streamsize>::max(), '\n');

**How to resolve the issue?**

Use cin.ignore(numeric\_limits<streamsize>::max(), '\n'); after cin to avoid read of \n by getline().

The numeric\_limits class template is from header file “limits”.

#include <iostream>

#include <string>

#include <limits>

using namespace std;

int main() {

int num =0;

string line;

cin>>num;

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

while(num > 0)

{

getline(cin, line); // user to enter a line of string having multiple words

cout<<line<<endl;

num--;

}

}

Input:

2

I am Saurav Kumar

Bangalore is hub for IT secotor

Output:

I am Saurav Kumar

Bangalore is hub for IT secotor

Remark:

In output, first line is now with input line but not a new line character or a blank line.

# getline() function and character array

Prerequisite: [getline(string) in C++](https://www.geeksforgeeks.org/getline-string-c/)  
In C++, stream classes support line-oriented functions, getline() and write() to perform input and output functions respectively. getline() function reads whole line of text that ends with new line or until the maximum limit is reached. getline() is the member function of istream class and has the syntax:

// (buffer, stream\_size, delimiter)

istream& getline(char\*, int size, char='\n')

// The delimiter character is considered as '\n'

istream& getline(char\*, int size)

The function does the following operations:  
1. Extracts character up to the delimiter.  
2. Stores the characters in the buffer.  
3. Maximum number of characters extracted is size – 1.  
Note that the terminator(or delimiter) character can be any character (like ‘ ‘, ‘, ‘ or any special character, etc.). The terminator character is read but not saved into a buffer, instead it is replaced by the null character.

filter\_none

edit

play\_arrow

brightness\_4

|  |
| --- |
| // C++ program to show the getline() with  // character array  #include <iostream>  using namespace std;    int main()  {      char str[20];      cout << "Enter Your Name::";        // see the use of getline() with array      // str also replace the above statement      // by cin >> str and see the difference      // in output      cin.getline(str, 20);        cout << "\nYour Name is:: " << str;      return 0;  } |

Input :

Aditya Rakhecha

Output :

Your Name is:: Aditya Rakhecha

In the above program, the statement cin.getline(str, 20) reads a string until it encounters the new line character or maximum number of characters (here 20). Try the function with different limits and see the output.

#include <iostream>

//#include <vector>

#include <string>

#include <limits>

//#include <sstream>

using namespace std;

int main() {

int num =0;

string line;

;

cin>>num;

cout<<"Total input line of string = "<<num<<endl;

cin.ignore(numeric\_limits<streamsize>::max(), '\n'); // to remove the /n created by cin so that getline will read first line properly.

while(num > 0)

{

char line[2000] ;

cin.getline(line, 2000);

cout<<line<<endl;

num--;

}

}