

# Forum:[Hack Forums](#)

Thread: [1 small error... anyone know how to fix this?](#)

Aus10 - January 30, 2016, 2:39 pm

I've looked everywhere on how to fix this. From my understanding, the error means that the word "open" simply will not work for the line of code. I have put dashes to the right of the specific line of code that breaks the program. Any thoughts guys?

```
Code #include <iostream>
#include <fstream>
#include <iomanip>
#include <string>
```

```
using namespace std;
```

```
bool openFileIn(fstream &, string);
void read_questions(fstream &);
```

```
int main()
{
    fstream fin;

    if (openFileIn(fin, "questions.txt"))
    {
        cout << "File opened.\n";
        cout << "Now reading data from the file.\n\n";
        read_questions(fin);
        fin.close();
        cout << "\nDone.\n";
    }
    else
        cout << "File open error!" << endl;
    return 0;
}
```

```
bool openFileIn(fstream &fin, string name)
{
    fin.open(name, ios::in); <-----
    if (fin.fail())
        return false;
    else
        return true;
}

void read_questions(fstream &fin)
{
    string line;
```

```

while (fin >> line)
{
    cout << line << endl;
}

}

```

**abratar** - January 30, 2016, 2:51 pm

I tested it with your code and it is working... Only the output is only 1 word each line but you can fix that. Any Error Message?

**Aus10** - January 30, 2016, 3:46 pm

In function 'bool openFileIn(std::fstream&, std::string)':  
error: no matching function for call to 'std::basic\_fstream<char>::open(std::string&, const openmode&)|  
note: candidate is:  
note: void std::basic\_fstream<\_CharT, \_Traits>::open(const char\*, std::ios\_base::openmode) [with \_CharT = char; \_Traits = std::char\_traits<char>; std::ios\_base::openmode = std::\_ios\_Openmode]|  
note: no known conversion for argument 1 from 'std::string {aka std::basic\_string<char>}' to 'const char\*'|  
||== Build failed: 1 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ==|

This is what I get when I run it. Thanks for your help.

Also if it is only the first word on each line, how can I fix that? Maybe a for loop using getline? Little rusty on combining loops with getlines :P

**Goodies** - January 30, 2016, 4:24 pm

It compiles and runs fine for me. Although I do see the error.

fin.open() looks like it is expecting a const char\* and you're giving it a string. I'm not sure why, actually, but try this:

```

PHP Code
<?php
#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>

```

```

using namespace std;

```

```

inline bool openFileIn(fstream &, const string&);
void read_questions(fstream &);

```

```

int main(){
    fstream fin;
    if (openFileIn(fin, "questions.txt")){
        cout << "File opened.\nNow reading data from the file.\n\n";
        read_questions(fin);
        fin.close();
        cout << "\nDone.\n";
    }
}

```

```

else
    cout << "File open error!" << endl;
return 0;
}

inline bool openFileIn(fstream &fin, const string &name){
    fin.open(name.c_str(), ios::in);
    if (fin.fail())
        return false;
    return true;
}

void read_questions(fstream &fin){
    string line;
    while (fin >> line){
        cout << line << endl;
    }
}
}

```

I use K&R, sorry. Hate that newline after functions/loops/ifs/etc...

**0xDEAD10CC** - January 30, 2016, 6:05 pm

It works only if you are compiling as C++11 or newer. This prototype was only introduced in C++11: Code `std::basic_filebuf<CharT, Traits>* open( const std::string& str, std::ios_base::openmode mode );` Older standards, you can only open with a C-string (`const char *`), and not `std::string` (`std::basic_string<char>`).

Hence, the reason why it worked for abratat, he was using C++11, and I would assume that OP is using GCC, or some port of it, and not passing any command line flags to allow it to be compiled via a newer standard. `std::basic_string<char>::c_str()` is the solution here unless you're using C++11/14.

**Aus10** - January 30, 2016, 6:19 pm

You fellas are much smarter then I. Thank you for your help, it worked! My program now shows the whole txt file one word at a time lol... Trying to get it to show normally with no end lines.

of  
the  
following  
is  
NOT  
a  
reason  
for  
writing  
functions?  
modularity  
reusability  
easier  
testing  
and

verification  
reduced  
code  
size

I've been messing around with end lines and for loops, however they dont fix it lol.. scared to stray too far from this set up and losing my progress.

**Goodies** - January 30, 2016, 8:21 pm

(Jan 30, 2016 07:19 PM)Aus10 Wrote: [\[CITE\]](#) You fellas are much smarter then I. Thank you for your help, it worked! My program now shows the whole txt file one word at a time lol... Trying to get it to show normally with no end lines.

of  
the  
...  
code  
size

I've been messing around with end lines and for loops, however they dont fix it lol.. scared to stray too far from this set up and losing my progress.

You can use something like (in your read\_questions function):

PHP Code

```
<?php  
std::string line;  
line.erase(std::remove(line.begin(), line.end(), '\n'), line.end());
```

It's probably not the best way, but it will definitely work and I think it's what you have to use with the way you have it.

If you don't want that, use std::getline instead of fstream >> std::string.

**0xDEAD10CC** - January 30, 2016, 9:19 pm

Show normally with no endlines? What does that mean? Extracting from a stream by default should skip whitespace.

**Goodies** - January 30, 2016, 10:14 pm

(Jan 30, 2016 10:19 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Show normally with no endlines? What does that mean? Extracting from a stream by default should skip whitespace.

@OP this. You're printing it with:  
cout << line << endl; omit that endl.

**Aus10** - January 31, 2016, 2:06 am

I tried that Goodies :P I took away the endl; and just ended it after line, but then it puts it all down on the same line from left to right in the prompt.( Which is bad) Because the user would not be able to see all of it without scrolling right. :) By show normally, I just meant output in the same format as the .txt file that it's in. If we can imagine taking a multiple choice test, with 4 answers choices underneath each question. ^^ To your first post Goodies, I will have to try that now. Will post back when I have tried it. Thanks!

I found one proper way to use the getline function, the problem with white space can be solved by the getline function. The function reads a line of data, including white space characters. Here is an example of the function call:

## PHP Code

```
<?php
getline(fin, str, '\n');
```

The 3 arguments in this statement are explained as follows:

**fin** : This is the name of the file stream object, it specifies the stream object from which the data is to be read.

**str** : This is the name of the string object. the data read from the file will be stored here.

**'\n'** : This is a delimiter character of choice. If this delimiter is encountered, it will cause the function to stop reading. (This is optional. If it's left out, '\n' is the default).

**0xDEAD10CC** - January 31, 2016, 3:37 pm

Quote: I just meant output in the same format as the .txt file that it's in.

Well then you need to keep track of where each newline, tab, space, etc... are within that file otherwise there's no way to tell. Why not just read the file and print out the file contents rather than retrieving individual tokens delimited by the whitespace? You can use `std::istream::read()` to read back the exact bytes of the file without using anything as a delimiter, not even a newline.

**Aus10** - January 31, 2016, 4:23 pm

I was able to read everything into the program this way.

## PHP Code

```
<?php
#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>

using namespace std;

void read_questions(fstream &);

int main()
{
    fstream fin;
    string input;
    fin.open("questions.txt", ios::in);
    read_questions(fin);
    return 0;
}

void read_questions(fstream &fin)
{
    string input;

    if (fin)
    {
        getline(fin, input);

        while (fin)
        {
```

```

        cout << input << endl;
        getline(fin, input);
    }
    fin.close();
}
}

```

It outputted like this:

Where does a C++ program begin its execution?

The first function in the file

The first function by alphabetic order of names

The function main( )

The function specified in the #DEFINE FirstFunction

How many values can be returned from a function by means of the "return" statement?

1

2

3

Lots and Lots

Which loop type is guaranteed to execute at least once?

for

while

do...while

all the above

Arrays are passed to functions by:

reference

value

either method

neither method

Global variables should be used...

in place of function parameters

only when absolutely needed

never

only during the full moon

When storing a floating point value to an integer type variable, the fractional part is:

rounded up

rounded down

branched

truncated

Which of the following is NOT a reason for writing functions?

modularity

reusability

easier testing and verification

reduced code size

Process returned 0 (0x0) execution time : 0.035 s

Press any key to continue.

Now I need to figure out how to "A" "B" "C" "D" Before each answer choice for every question. I'm thinking a basic for loop will work. Will update when I figure it out. =)

Update:

Changed EVERYTHING. UGH.

Couldn't get ABCD to show for each question using a while loop, so i switched to for loop. Here is what I did.

PHP Code

```
<?php
#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>

using namespace std;
// Function Prototype
void read_questions();

int main()
{
    // fstream fin; // File stream input
    // string question; // To hold file input
    // Opening the file in input mode
    // fin.open("questions.txt", ios::in);
    read_questions();
    return 0;
}

void read_questions()
{
    string question;
    string choice1;
    string choice2;
    string choice3;
    string choice4;
    string space;
    fstream fin;

    fin.open("questions.txt");

    for (int x = 0; x < 7; x++)
    {
        getline(fin, question);
        getline(fin, choice1);
        getline(fin, choice2);
        getline(fin, choice3);
        getline(fin, choice4);
        getline(fin, space);
    }
}
```

```
cout << question << endl;
cout << "A: " << choice1 << endl;
cout << "B: " << choice2 << endl;
cout << "C: " << choice3 << endl;
cout << "D: " << choice4 << endl;
cout << space << endl;

}
```

Here is the result thus far:

Where does a C++ program begin its execution?

- A: The first function in the file
- B: The first function by alphabetic order of names
- C: The function main( )
- D: The function specified in the #DEFINE FirstFunction

How many values can be returned from a function by means of the "return" statement?

- A: 1
- B: 2
- C: 3
- D: Lots and Lots

Which loop type is guaranteed to execute at least once?

- A: for
- B: while
- C: do...while
- D: all the above

Arrays are passed to functions by:

- A: reference
- B: value
- C: either method
- D: neither method

Global variables should be used...

- A: in place of function parameters
- B: only when absolutely needed
- C: never
- D: only during the full moon

When storing a floating point value to an integer type variable, the fractional part is:

- A: rounded up
- B: rounded down
- C: branched
- D: truncated

Which of the following is NOT a reason for writing functions?

- A: modularity
- B: reusability
- C: easier testing and verification
- D: reduced code size

Process returned 0 (0x0) execution time : 0.028 s

Press any key to continue.

UPDATE: Now I put it through an array to handle the ABCD and have it stored into a 2D array.

PHP Code

```
<?php
#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>

using namespace std;
// Function Prototype
void read_questions();

int main()
{
    read_questions();
    return 0;
}

void read_questions()
{
    string question;
    string choice1;
    string choice2;
    string choice3;
    string choice4;
    string space;
    fstream fin;
    string input_arr[50][50];
    fin.open("questions.txt");

    for (int q = 0; q < 7; q++)
    {
        getline(fin, question);
        getline(fin, choice1);
        getline(fin, choice2);
        getline(fin, choice3);
        getline(fin, choice4);
        getline(fin, space);

        input_arr[q][0] = question;
        input_arr[q][1] = "A. " + choice1;
        input_arr[q][2] = "B. " + choice2;
        input_arr[q][3] = "C. " + choice3;
        input_arr[q][4] = "D. " + choice4;
```

```

        input_arr[q][5] = space;

        cout << input_arr[q][0] << endl;
        cout << input_arr[q][1] << endl;
        cout << input_arr[q][2] << endl;
        cout << input_arr[q][3] << endl;
        cout << input_arr[q][4] << endl;
        cout << input_arr[q][5] << endl;

    }

}

```

**Russ.** - January 31, 2016, 9:36 pm

^^ That looks like a thread. Problem solved mate?

**Aus10** - January 31, 2016, 9:42 pm

So I've basically made this a blog lol.

New update, last update for a day or two.

Added the read\_answers function, soon going to have the program compare the users answer choice to the correct answers in the .txt file.

PHP Code

```

<?php
#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>

using namespace std;
// Function Prototype
double read_questions();
double read_answers();

int main()
{
    read_questions();
    read_answers();
    return 0;
}

double read_questions()
{
    string question;
    string choice1;
    string choice2;
    string choice3;
    string choice4;
    string space;
    fstream fin;

```

```

string input_arr[50][50];
fin.open("questions.txt");

for (int q = 0; q < 7; q++)
{
    getline(fin, question);
    getline(fin, choice1);
    getline(fin, choice2);
    getline(fin, choice3);
    getline(fin, choice4);
    getline(fin, space);

    input_arr[q][0] = question;
    input_arr[q][1] = "A. " + choice1;
    input_arr[q][2] = "B. " + choice2;
    input_arr[q][3] = "C. " + choice3;
    input_arr[q][4] = "D. " + choice4;
    input_arr[q][5] = space;

    cout << input_arr[q][0] << endl;
    cout << input_arr[q][1] << endl;
    cout << input_arr[q][2] << endl;
    cout << input_arr[q][3] << endl;
    cout << input_arr[q][4] << endl;
    cout << input_arr[q][5] << endl;

}

}

double read_answers()
{
    fstream fin;
    char answer_arr[50];
    char answer;
    fin.open("answers.txt");

    for (int y = 0; y < 7; y++)
    {
        fin >> answer;
        answer_arr[y] = answer;
    }
}

```

Yep lol problem solved. Thanks guys !! :) Just excited that the program is taking off. Don't really know why im updating it so much lol.