

# Reference Manual

Generated by Doxygen 1.8.13

## Contents

<b>1</b>	<b>Test List</b>	<b>1</b>
<b>2</b>	<b>Hierarchical Index</b>	<b>2</b>
2.1	Class Hierarchy . . . . .	2
<b>3</b>	<b>Class Index</b>	<b>2</b>
3.1	Class List . . . . .	2
<b>4</b>	<b>File Index</b>	<b>3</b>
4.1	File List . . . . .	3
<b>5</b>	<b>Class Documentation</b>	<b>3</b>
5.1	FillnQuestion Class Reference . . . . .	3
5.1.1	Detailed Description . . . . .	7
5.1.2	Constructor & Destructor Documentation . . . . .	8
5.1.3	Member Function Documentation . . . . .	9
5.2	NumericQuestion Class Reference . . . . .	12
5.2.1	Detailed Description . . . . .	17
5.2.2	Constructor & Destructor Documentation . . . . .	18
5.2.3	Member Function Documentation . . . . .	19
5.2.4	Member Data Documentation . . . . .	22
5.3	Question Class Reference . . . . .	22
5.3.1	Detailed Description . . . . .	27
5.3.2	Constructor & Destructor Documentation . . . . .	28
5.3.3	Member Function Documentation . . . . .	29
5.3.4	Member Data Documentation . . . . .	34

<b>1</b>	<b>Test List</b>	<b>1</b>
<b>6</b>	<b>File Documentation</b>	<b>34</b>
6.1	FillInQuestion.h File Reference . . . . .	35
6.1.1	Detailed Description . . . . .	39
6.1.2	Macro Definition Documentation . . . . .	40
6.2	lab10a.cpp File Reference . . . . .	40
6.2.1	Function Documentation . . . . .	42
6.3	NumericQuestion.h File Reference . . . . .	52
6.4	question.h File Reference . . . . .	55
	<b>Index</b>	<b>59</b>

# 1 Test List

## Member **TEST\_CASE** ("Question Inheritance Test")

The `set_text()`, `set_answer()`, and `check_answer()` member functions of the base with:

**Question:** Who was the inventor of C++?

Answer (user-coded):Bjarne Stroustrup

The `set_text()`, `set_answer()`, and `check_answer()` member functions of the base with an incorrect string:

**Question:** Who was the inventor of C++?

Answer (user-coded): Jack Daniels

The `set_text()`, `set_answer()`, and `check_answer()` member functions of the derived `NumericQuestion` class with:

**Question:** Solve this math problem:  $36 + 64 = ?$

Answer (user-coded): 100

The `set_text()`, `set_answer()`, and `check_answer()` member functions of the derived `NumericQuestion` class with a value that is within 0.01 error of the answer:

**Question:** Solve this math problem:  $36 + 64 = ?$

Answer (user-coded): 99.999

The `set_text()`, `set_answer()`, and `check_answer()` member functions of the derived `NumericQuestion` class with an incorrect value:

**Question:** Solve this math problem:  $36 + 64 = ?$

Answer (user-coded): 12

The `set_text()`, `set_answer()`, and `check_answer()` member functions of the derived `FillnQuestion` class with:

**Question:** The inventor of C++ was \_\_\_\_\_.

Answer (user-coded):Bjarne Stroustrup

The `set_text()`, `set_answer()`, and `check_answer()` member functions of the derived `FillnQuestion` class with:

**Question:** The inventor of C++ was \_\_\_\_\_.

Answer (user-coded):Bjarne Stroustrup

## 2 Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

<b>Question</b>	<b>22</b>
<b>FillnQuestion</b>	<b>3</b>
<b>NumericQuestion</b>	<b>12</b>

## 3 Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">FillInQuestion</a>	3
<a href="#">NumericQuestion</a>	12
<a href="#">Question</a>	22

## 4 File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

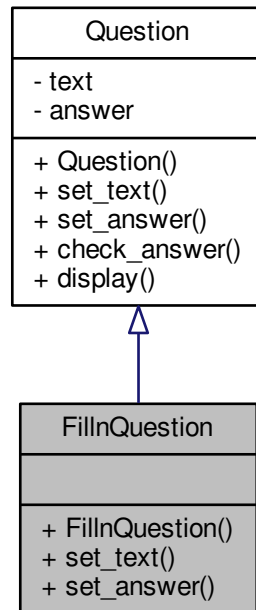
<a href="#">FillInQuestion.h</a>	35
<a href="#">lab10a.cpp</a>	40
<a href="#">NumericQuestion.h</a>	52
<a href="#">question.h</a>	55

## 5 Class Documentation

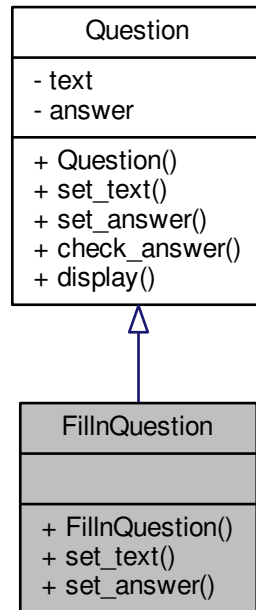
### 5.1 FillInQuestion Class Reference

```
#include <FillInQuestion.h>
```

Inheritance diagram for FillInQuestion:



Collaboration diagram for FillInQuestion:



## Public Member Functions

- [FillnQuestion](#) ()
- void [set\\_text](#) (std::string question\_text)
- void [set\\_answer](#) (std::string correct\_response)



### 5.1.1 Detailed Description

Definition at line 10 of file FillInQuestion.h.

## 5.1.2 Constructor & Destructor Documentation

### 5.1.2.1 FillnQuestion()

```
FillnQuestion::FillnQuestion ( )
```

Constructs a question with empty text and answer.

Definition at line 29 of file FillnQuestion.h.

```
30 {  
31  
32 }
```

### 5.1.3 Member Function Documentation

#### 5.1.3.1 set\_answer()

```
void FillnQuestion::set_answer (
    std::string correct_response )
```

##### Parameters

<i>response</i>	the response to check
-----------------	-----------------------

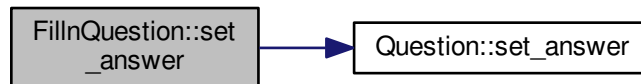
##### Returns

true if the response was correct, false otherwise

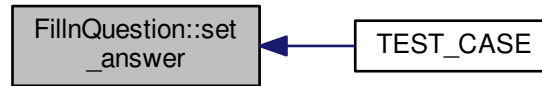
Definition at line 64 of file FillnQuestion.h.

```
65 {
66     Question::set_answer (correct_response);
67 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



#### 5.1.3.2 set\_text()

```
void FillnQuestion::set_text (  
    std::string question_text )
```

##### Parameters

<i>question_text</i>	the text of this question
----------------------	---------------------------

Definition at line 33 of file FillnQuestion.h.

```
34 {  
35     int count = 0;  
36     int count2 = 0;  
37     std::string question_buffer;  
38     bool under_score = false;
```

```
39     for(unsigned int i = 0; i < question_text.size(); i++)
40     {
41         if(question_text[i] == '_')
42         {
43             if(under_score == true)
44             {
45                 count2 = i;
46                 under_score = false;
47             }
48             else
49             {
50                 under_score = true;
51             }
52         }
53         if(under_score == true)
54         {
55             count++;
56             question_buffer += "_";
57         }
58     }
59     std::string blank_question = question_text.substr(0, count2 - count)
60 + question_buffer
61 + question_text.substr(count2, question_text.size() - count2);
62     Question::set_text(blank_question);
63 }
```

Here is the call graph for this function:



Here is the caller graph for this function:



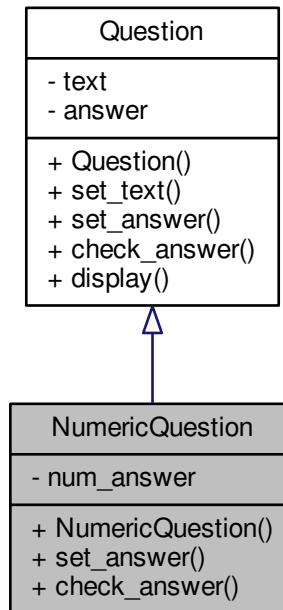
The documentation for this class was generated from the following file:

- [FillInQuestion.h](#)

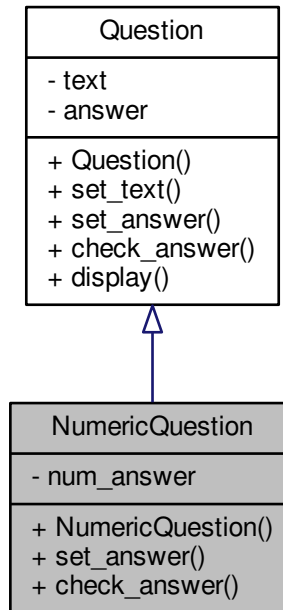
## 5.2 NumericQuestion Class Reference

```
#include <NumericQuestion.h>
```

Inheritance diagram for NumericQuestion:



Collaboration diagram for NumericQuestion:





### Public Member Functions

- [NumericQuestion](#) ()
- void [set\\_answer](#) (double correct\_response)
- bool [check\\_answer](#) (double response) const

## Private Attributes

- double `num_answer`

### 5.2.1 Detailed Description

Definition at line 8 of file NumericQuestion.h.

## 5.2.2 Constructor & Destructor Documentation

### 5.2.2.1 NumericQuestion()

```
NumericQuestion::NumericQuestion ( )
```

Constructs a [NumericQuestion](#) with empty text and answer.

Definition at line 27 of file NumericQuestion.h.

```
28 {  
29 }
```

### 5.2.3 Member Function Documentation

#### 5.2.3.1 check\_answer()

```
bool NumericQuestion::check_answer (  
    double response ) const
```

##### Parameters

<i>response</i>	the type double response to check
-----------------	-----------------------------------

##### Returns

true if the response was correct, false otherwise

Definition at line 35 of file NumericQuestion.h.

```
36 {  
37     double max_error = 0.01;  
38     double response_error = abs(response - num_answer);  
39     if(response_error <= max_error)  
40     {  
41         std::cout << "Correct!" << std::endl;  
42         if(response != num_answer)  
43         {  
44             std::cout << "Your answer was within the allowed error (0.01)" << std::endl;  
45             std::cout << "Actual Answer: " << num_answer << std::endl;  
46         }  
47     }  
48 }
```

```
47         return true;
48     }
49     else
50     {
51         std::cout << "Incorrect." << std::endl;
52         std::cout << "Correct Answer: " << num_answer << std::endl;
53         return false;
54     }
55 }
```

Here is the caller graph for this function:



### 5.2.3.2 set\_answer()

```
void NumericQuestion::set_answer (
    double correct_response )
```

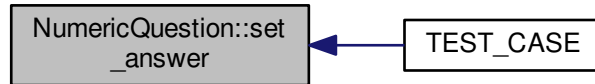
#### Parameters

<code>correct_response</code>	the double type answer for this question
-------------------------------	--

Definition at line 30 of file NumericQuestion.h.

```
31 {  
32     num_answer = correct_response;  
33 }
```

Here is the caller graph for this function:



## 5.2.4 Member Data Documentation

### 5.2.4.1 num\_answer

```
double NumericQuestion::num_answer [private]
```

Definition at line 25 of file NumericQuestion.h.

The documentation for this class was generated from the following file:

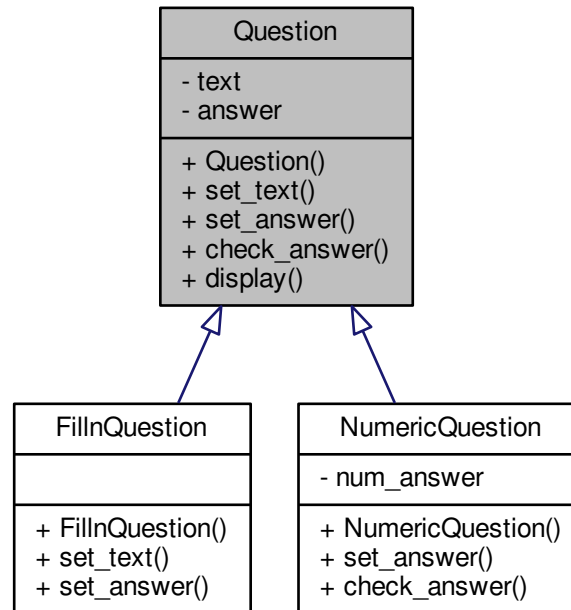
- [NumericQuestion.h](#)

## 5.3 Question Class Reference

```
#include <question.h>
```



Inheritance diagram for Question:



Collaboration diagram for Question:

Question
- text - answer
+ Question() + set_text() + set_answer() + check_answer() + display()

### Public Member Functions

- [Question](#) ()
- void [set\\_text](#) (std::string question\_text)
- void [set\\_answer](#) (std::string correct\_response)
- bool [check\\_answer](#) (std::string response) const
- void [display](#) () const

## Private Attributes

- std::string [text](#)
- std::string [answer](#)

#### 5.3.1 Detailed Description

Definition at line 7 of file question.h.

### 5.3.2 Constructor & Destructor Documentation

#### 5.3.2.1 Question()

```
Question::Question ( )
```

Constructs a question with empty text and answer.

Definition at line 35 of file question.h.

```
36 {  
37 }
```

### 5.3.3 Member Function Documentation

#### 5.3.3.1 check\_answer()

```
bool Question::check_answer (
    std::string response ) const
```

##### Parameters

<i>response</i>	the response to check
-----------------	-----------------------

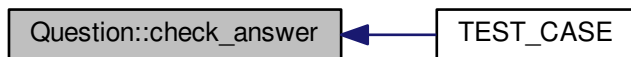
##### Returns

true if the response was correct, false otherwise

Definition at line 48 of file question.h.

```
49 {
50     if(response == answer)
51     {
52         std::cout << "Correct!" << std::endl;
53         return true;
54     }
55     else
56     {
57         std::cout << "Incorrect" << std::endl;
58         std::cout << "Correct Answer: " << answer << std::endl;
59         return false;
60     }
61 }
```

Here is the caller graph for this function:



#### 5.3.3.2 display()

```
void Question::display ( ) const
```

Displays this question.

Definition at line 63 of file question.h.

```
64 {  
65     std::cout << text << std::endl;  
66 }
```

Here is the caller graph for this function:





### 5.3.3.3 set\_answer()

```
void Question::set_answer (
    std::string correct_response )
```

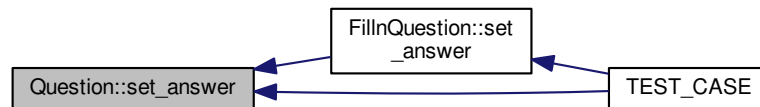
#### Parameters

<i>correct_response</i>	the answer for this question
-------------------------	------------------------------

Definition at line 43 of file question.h.

```
44 {
45     answer = correct_response;
46 }
```

Here is the caller graph for this function:



#### 5.3.3.4 set\_text()

```
void Question::set_text (
    std::string question_text )
```

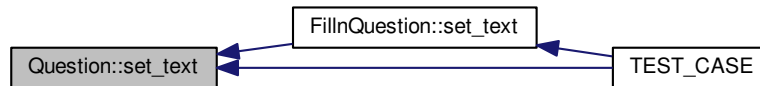
## Parameters

<i>question_text</i>	the text of this question
----------------------	---------------------------

Definition at line 38 of file question.h.

```
39 {  
40     text = question_text;  
41 }
```

Here is the caller graph for this function:



### 5.3.4 Member Data Documentation

#### 5.3.4.1 answer

```
std::string Question::answer [private]
```

Definition at line 33 of file question.h.

#### 5.3.4.2 text

```
std::string Question::text [private]
```

Definition at line 32 of file question.h.

The documentation for this class was generated from the following file:

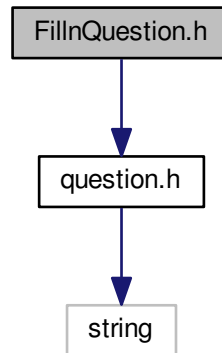
- [question.h](#)

## 6 File Documentation

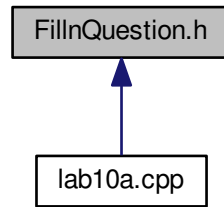
## 6.1 FillnQuestion.h File Reference

```
#include "question.h"
```

Include dependency graph for FillnQuestion.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [FillInQuestion](#)

## Macros

- #define [FILLINQUESTION\\_H](#)



### 6.1.1 Detailed Description

Such a question is constructed with a string that contains the answer, surrounded by `_`, for example, "The inventor of C++ was `_Bjarne Stroustrup_`". The question should be displayed as The inventor of C++ was \_\_\_\_\_.

## 6.1.2 Macro Definition Documentation

### 6.1.2.1 FILLINQUESTION\_H

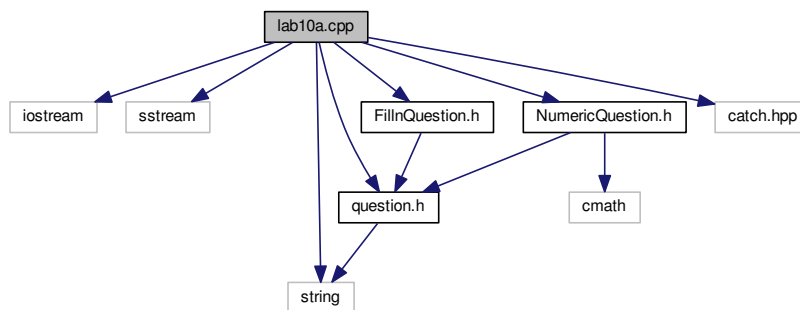
```
#define FILLINQUESTION_H
```

Definition at line 2 of file FillnQuestion.h.

## 6.2 lab10a.cpp File Reference

```
#include <iostream>
#include <sstream>
#include <string>
#include "question.h"
#include "NumericQuestion.h"
#include "FillnQuestion.h"
#include "catch.hpp"
```

Include dependency graph for lab10a.cpp:



## Functions

- [TEST\\_CASE](#) ("Question Inheritance Test")

## 6.2.1 Function Documentation

### 6.2.1.1 TEST\_CASE()

```
TEST_CASE (
    "Question Inheritance Test" )
```

Test the base [Question](#) class, and 2 derived classes (Numeric and Fill in type)

**Test** The `set_text()`, `set_answer()`, and `check_answer()` member functions of the base with:

**Question:** Who was the inventor of C++?

**Answer** (user-coded):Bjarne Stroustrup

#### Returns

the question and a boolean value if the answer was correct or not (1 = true and 0 = false)

This test will return:

Who was the inventor of C++?

Your answer: Bjarne Stroustrup

Correct!

1

```
lab10a.cpp:45: passed: ss.str() == "Who was the inventor of C++?\n" "Your answer: Bjarne Stroustrup\n" "Correct!\n" "1\n"
Your answer: Bjarne Stroustrup
Correct!
1
"
==
"Who was the inventor of C++?
Your answer: Bjarne Stroustrup
Correct!
1
"
Passed 1 test case with 1 assertion.
```

**Test** The `set_text()`, `set_answer()`, and `check_answer()` member functions of the base with an incorrect string:

**Question:** Who was the inventor of C++?

Answer (user-coded): Jack Daniels

### Returns

the question and a boolean value if the answer was correct or not (1 = true and 0 = false)

This test will return:

Who was the inventor of C++?

Your answer: Jack Daniels

Incorrect!

Correct Answer: Bjarne Stroustrup

0

```
lab10a.cpp:75: passed: ss.str() == "Who was the inventor of C++?\n" "Your answer: Jack Daniels\n" "Incorrect\n" "Correct\n"
Your answer: Jack Daniels
Incorrect
Correct Answer: Bjarne Stroustrup
0
"
==
"Who was the inventor of C++?
Your answer: Jack Daniels
Incorrect
Correct Answer: Bjarne Stroustrup
0
"
Passed 1 test case with 1 assertion.
```

**Test** The `set_text()`, `set_answer()`, and `check_answer()` member functions of the derived [NumericQuestion](#) class with:

**Question:** Solve this math problem:  $36 + 64 = ?$

Answer (user-coded): 100

## Returns

the question, both a string indicating correct/incorrect and a boolean value if the answer was correct or not (1 = true and 0 = false)

This test will return:

Solve this math problem:

36 + 64 = ?

Your answer: 100

Correct!

1

```
lab10a.cpp:73: passed: ss.str() == "Solve this math problem: \n36 + 64 = ?\n" "Your answer: 100\n" "Correct!\n" "1\n" for
36 + 64 = ?
Your answer: 100
Correct!
1
"
==
"Solve this math problem:
36 + 64 = ?
Your answer: 100
Correct!
1
"
Passed 1 test case with 1 assertion.
```

**Test** The `set_text()`, `set_answer()`, and `check_answer()` member functions of the derived [NumericQuestion](#) class with a value that is within 0.01 error of the answer:

**Question:** Solve this math problem: 36 + 64 = ?

**Answer (user-coded):** 99.999

## Returns

the question, both a string indicating correct/incorrect and a boolean value if the answer was correct or not (1 = true and 0 = false)

This test will return:

Solve this math problem:

36 + 64 = ?

Your answer: 99.999

Correct!

Your answer was within the allowed error (0.01)

Actual Answer: 100

1

```
lab10a.cpp:105: passed: ss.str() == "Solve this math problem: \n36 + 64 = ?\n" "Your answer: 99.999\n" "Correct!\n" "Your
36 + 64 = ?
Your answer: 99.999
Correct!
Your answer was within the allowed error (0.01)
Actual Answer: 100
1
"
==
"Solve this math problem:
36 + 64 = ?
Your answer: 99.999
Correct!
Your answer was within the allowed error (0.01)
Actual Answer: 100
1
"
Passed 1 test case with 1 assertion.
```

**Test** The `set_text()`, `set_answer()`, and `check_answer()` member functions of the derived [NumericQuestion](#) class with an incorrect value:

**Question:** Solve this math problem:  $36 + 64 = ?$

**Answer (user-coded):** 12

## Returns

the question, both a string indicating correct/incorrect and a boolean value if the answer was correct or not (1 = true and 0 = false)

This test will return:

Solve this math problem:

$36 + 64 = ?$

Your answer: 12

Incorrect

Correct Answer: 100

0

```

lab10a.cpp:135: passed: ss.str() == "Solve this math problem: \n36 + 64 = ?\n" "Your answer: 12\n" "Incorrect.\n" "Correct.\n"
36 + 64 = ?
Your answer: 12
Incorrect.
Correct Answer: 100
0
"
==
"Solve this math problem:
36 + 64 = ?
Your answer: 12
Incorrect.
Correct Answer: 100
0
"
Passed 1 test case with 1 assertion.

```

**Test** The `set_text()`, `set_answer()`, and `check_answer()` member functions of the derived [FillnQuestion](#) class with:

**Question:** The inventor of C++ was \_\_\_\_\_.

**Answer (user-coded):** Bjarne Stroustrup

## Returns

the question and a boolean value if the answer was correct or not (1 = true and 0 = false)

This test will return:

The inventor of C++ is\_\_\_\_\_.

Your answer: Bjarne Stroustrup

Correct!

1

```

lab10a.cpp:198: passed: ss.str() == "The inventor of C++ was _____.\n" "Your answer: Bjarne Stroustrup\n" "
Your answer: Bjarne Stroustrup
Correct!
1
"
==
"The inventor of C++ was _____.
Your answer: Bjarne Stroustrup

```



```
Correct!
1
"
Passed 1 test case with 1 assertion.
```

**Test** The `set_text()`, `set_answer()`, and `check_answer()` member functions of the derived [FillnQuestion](#) class with:

**Question:** The inventor of C++ was \_\_\_\_\_.

**Answer (user-coded):** Bjarne Stroustrup

### Returns

the question and a boolean value if the answer was correct or not (1 = true and 0 = false)

This test will return:

The inventor of C++ is \_\_\_\_\_.

Your answer: Bjarne Stroustrup

Correct!

1

```
lab10a.cpp:226: passed: ss.str() == "The inventor of C++ was _____.\\n" "Your answer: Morgan Freeman\\n" "Inc
Your answer: Morgan Freeman
Incorrect
Correct Answer: Bjarne Stroustrup
0
"
==
"The inventor of C++ was _____.
Your answer: Morgan Freeman
Incorrect
Correct Answer: Bjarne Stroustrup
0
"
Passed 1 test case with 1 assertion.
```

Definition at line 11 of file lab10a.cpp.

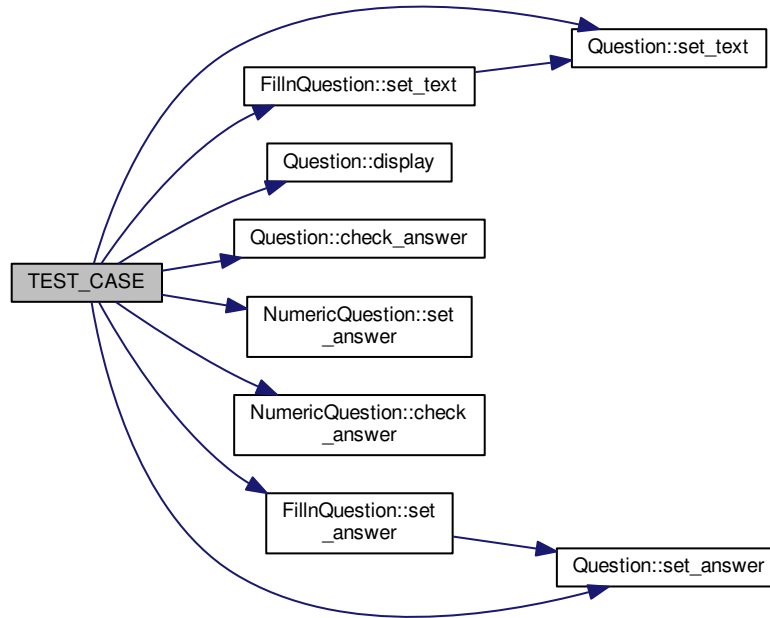
```
12 {
16     std::string response;
17     std::streambuf *b = std::cout.rdbuf();
18     std::stringstream ss;
19     std::streambuf *sb = ss.rdbuf();
20     std::cout.rdbuf(sb);
21     SECTION("Question Base Class Test")
22     {
35         Question q1;
36         q1.set_text("Who was the inventor of C++?");
37         q1.set_answer("Bjarne Stroustrup");
38         q1.display();
39         //~ getline(std::cin, response);
40         response = "Bjarne Stroustrup";
41         std::cout << "Your answer: " << response << std::endl;
42         std::cout << q1.check_answer(response) << std::endl;
43         CHECK(ss.str() == "Who was the inventor of C++?\n"
44               "Your answer: Bjarne Stroustrup\n"
45               "Correct!\n"
46               "1\n");
47     }
48     SECTION("Question Base Class Test (Incorrect)")
49     {
63         Question q1;
64         q1.set_text("Who was the inventor of C++?");
65         q1.set_answer("Bjarne Stroustrup");
66         q1.display();
67         //~ getline(std::cin, response);
68         response = "Jack Daniels";
69         std::cout << "Your answer: " << response << std::endl;
70         std::cout << q1.check_answer(response) << std::endl;
```

```
71         CHECK(ss.str() == "Who was the inventor of C++?\n"
72                     "Your answer: Jack Daniels\n"
73                     "Incorrect\n"
74                     "Correct Answer: Bjarne Stroustrup\n"
75                     "0\n");
76     }
77     SECTION("Numeric Question Derived Class Test (Correct Answer)")
78     {
79         NumericQuestion NQ1;
80         NQ1.set_text("Solve this math problem: \n36 + 64 = ?");
81         NQ1.set_answer(100.0);
82         NQ1.display();
83         //~ getline(std::cin, response);
84         //~ double num_response = std::stod(response, nullptr);
85         double num_response = 100.0;
86         std::cout << "Your answer: " << num_response << std::endl;
87         std::cout << NQ1.check_answer(num_response) << std::endl;
88         CHECK(ss.str() == "Solve this math problem: \n36 + 64 = ?\n"
89                     "Your answer: 100\n"
90                     "Correct!\n"
91                     "1\n");
92     }
93     SECTION("Numeric Question Derived Class Test (Within 0.01)")
94     {
95         NumericQuestion NQ1;
96         NQ1.set_text("Solve this math problem: \n36 + 64 = ?");
97         NQ1.set_answer(100.0);
98         NQ1.display();
99         //~ getline(std::cin, response);
100        //~ double num_response = std::stod(response, nullptr);
101        double num_response = 99.999;
```

```
132         std::cout << "Your answer: " << num_response << std::endl;
133         std::cout << NQ1.check_answer(num_response) << std::endl;
134         CHECK(ss.str() == "Solve this math problem: \n36 + 64 = ?\n"
135             "Your answer: 99.999\n"
136             "Correct!\n"
137             "Your answer was within the allowed error (0.01)\n"
138             "Actual Answer: 100\n"
139             "1\n");
140     }
141     SECTION("Numeric Question Derived Class Test (Incorrect)")
142     {
143         NumericQuestion NQ1;
144         NQ1.set_text("Solve this math problem: \n36 + 64 = ?");
145         NQ1.set_answer(100.0);
146         NQ1.display();
147
148         //~ getline(std::cin, response);
149         //~ double num_response = std::stod(response, nullptr);
150         double num_response = 12;
151         std::cout << "Your answer: " << num_response << std::endl;
152         std::cout << NQ1.check_answer(num_response) << std::endl;
153         CHECK(ss.str() == "Solve this math problem: \n36 + 64 = ?\n"
154             "Your answer: 12\n"
155             "Incorrect.\n"
156             "Correct Answer: 100\n"
157             "0\n");
158     }
159     SECTION("Fill in Question Derived Class Test")
160     {
161         FillInQuestion FnQ1;
162         FnQ1.set_text("The inventor of C++ was _Bjarne Stroustrup_.");
```

```
189         FnQ1.set_answer("Bjarne Stroustrup");
190         FnQ1.display();
191         //~ getline(std::cin, response);
192         response = "Bjarne Stroustrup";
193         std::cout << "Your answer: " << response << std::endl;
194         std::cout << FnQ1.check_answer(response) << std::endl;
195         CHECK(ss.str() == "The inventor of C++ was _____.\\n"
196                        "Your answer: Bjarne Stroustrup\\n"
197                        "Correct!\\n"
198                        "1\\n");
199     }
200     SECTION("Fill in Question Derived Class Test (Incorrect)")
201     {
202         FillnQuestion FnQ1;
203         FnQ1.set_text("The inventor of C++ was _Bjarne Stroustrup_.");
204         FnQ1.set_answer("Bjarne Stroustrup");
205         FnQ1.display();
206         //~ getline(std::cin, response);
207         response = "Morgan Freeman";
208         std::cout << "Your answer: " << response << std::endl;
209         std::cout << FnQ1.check_answer(response) << std::endl;
210         CHECK(ss.str() == "The inventor of C++ was _____.\\n"
211                        "Your answer: Morgan Freeman\\n"
212                        "Incorrect\\n"
213                        "Correct Answer: Bjarne Stroustrup\\n"
214                        "0\\n");
215     }
216 }
```

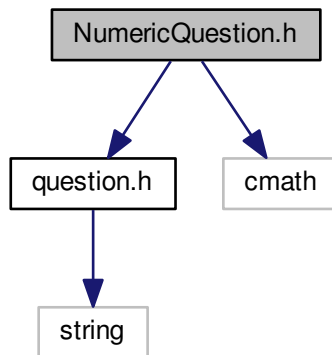
Here is the call graph for this function:



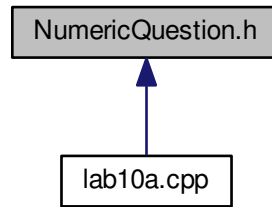
### 6.3 NumericQuestion.h File Reference

```
#include "question.h"  
#include <cmath>
```

Include dependency graph for NumericQuestion.h:



This graph shows which files directly or indirectly include this file:





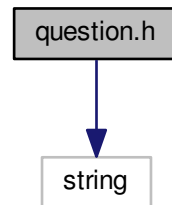
## Classes

- class [NumericQuestion](#)

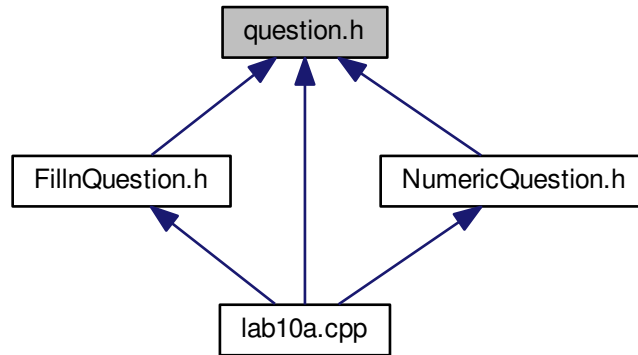
## 6.4 question.h File Reference

```
#include <string>
```

Include dependency graph for question.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Question](#)



## Index

- answer
  - Question, [34](#)
- check\_answer
  - NumericQuestion, [19](#)
  - Question, [29](#)
- display
  - Question, [30](#)
- FILLINQUESTION\_H
  - FillInQuestion.h, [40](#)
- FillInQuestion, [3](#)
  - FillInQuestion, [8](#)
  - set\_answer, [9](#)
  - set\_text, [10](#)
- FillInQuestion.h, [35](#)
  - FILLINQUESTION\_H, [40](#)
- lab10a.cpp, [40](#)
  - TEST\_CASE, [42](#)
- num\_answer
  - NumericQuestion, [22](#)
- NumericQuestion, [12](#)
  - check\_answer, [19](#)
  - num\_answer, [22](#)
  - NumericQuestion, [18](#)
  - set\_answer, [20](#)
- NumericQuestion.h, [52](#)
- Question, [22](#)
  - answer, [34](#)
  - check\_answer, [29](#)
  - display, [30](#)
  - Question, [28](#)
  - set\_answer, [31](#)
  - set\_text, [31](#)
  - text, [34](#)
- question.h, [55](#)
- set\_answer
  - FillInQuestion, [9](#)
  - NumericQuestion, [20](#)
  - Question, [31](#)
- set\_text
  - FillInQuestion, [10](#)
  - Question, [31](#)
- TEST\_CASE
  - lab10a.cpp, [42](#)
- text
  - Question, [34](#)