Reference Manual

Generated by Doxygen 1.8.13

Contents

1 Test List

Member TEST_CASE ("clear")

clear member function to reset total and item count A simulated cash register that tracks the item count and the total amount due.

```
cashregister.cpp:59: passed: cr.get_count()==0 for: 0 == 0
cashregister.cpp:60: passed: cr.get_total()==0 for: 0.0 == 0
Passed 1 test case with 2 assertions.
```

Member TEST_CASE ("output")

clear() display_all() add_item() get_total() get_count() member functions testing output of all functions A simulated cash register that tracks the item count and the total amount due.

```
cashregister.cpp:168: passed: ss.str() == "The prices for the items are :\n"
"14\n23.16\n9.82\n0.98\n80\n13.45\n" "141.41\n""6\n" for: "The prices for the
items are :
14
23.16
9.82
0.98
80
13.45
141.41
6
"
==
"The prices for the items are :
14
23.16
9.82
0.98
80
13.45
141.41
6
"
Passed 1 test case with 1 assertion.
```

Member TEST_CASE ("display_all")

display_all() member function to check if adding the three prices, 10.99, 25.25, 42.75 were inputted into the vector A simulated cash register that tracks the item count and the total amount due.

```
cashregister.cpp:132: passed: ss.str() == "The prices for the items are :\n"
"10.99\n25.25\n42.75\n" for: "The prices for the items are :
10.99
25.25
42.75
"
==
"The prices for the items are :
10.99
25.25
42.75
"
Passed 1 test case with 1 assertion.
```

Member TEST_CASE ("total")

get_total() member function to check if adding three items will add up to 85.60 A simulated cash register that tracks the item count and the total amount due.

```
cashregister.cpp:109: passed: cr.get_total()==85.60 for: 85.6 == 85.6
Passed 1 test case with 1 assertion.
```

Member TEST_CASE ("count")

get_count() member function to check if adding three items will change the count to be 3 A simulated cash register that tracks the item count and the total amount due.

```
cashregister.cpp:91: passed: cr.get_count()==3 for: 3 == 3
Passed 1 test case with 1 assertion.
```

Member TEST_CASE ("add_item")

add_item member function to check if adding an item with price of 23.25 will change the count to be 1 and total to be 23.25 A simulated cash register that tracks the item count and the total amount due.

```
cashregister.cpp:77: passed: cr.get_count()==1 for: 1 == 1
cashregister.cpp:78: passed: cr.get_total()==23.25 for: 23.25 == 23.25
Passed 1 test case with 2 assertions.
```

2 Class Index

2.1 Class List

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

3 File Index

3.1 File List

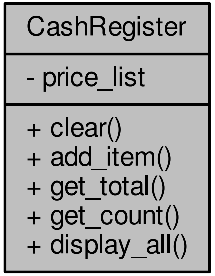
Here is a list of all files with brief descriptions:

4 Class Documentation

4.1 CashRegister Class Reference

```
#include <cashregister.h>
```

Collaboration diagram for CashRegister:



Public Member Functions

- void clear ()
- void add_item (double price)
- double get_total () const
- int get_count () const
- void display_all () const

Private Attributes

- std::vector< double > price_list

4.1.1 Detailed Description

A simulated cash register that tracks the item count and the total amount due.

Definition at line 5 of file cashregister.h.

4.1.2 Member Function Documentation

4.1.2.1 add_item()

```
void CashRegister::add_item (
             double price )
```

Adds an item to this cash register.

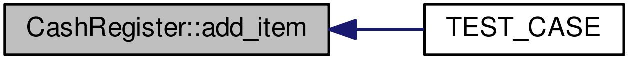
Parameters

price	the price of this item

Definition at line 33 of file cashregister.cpp.

```
{
    //item_count++;
36      //total_price += price;
37      price_list.push_back(price);
38 }
```

Here is the caller graph for this function:



4.1.2.2 clear()

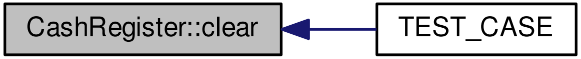
`void CashRegister::clear ( )`

Clears the item count and the total.

Definition at line 5 of file cashregister.cpp.

```
6 {
7     //item_count = 0;
8     //total_price = 0;
9     while (!price_list.empty())
10     {
11           price_list.pop_back();
12       }
13 }
```

Here is the caller graph for this function:



4.1.2.3 display_all()

`void CashRegister::display_all ( ) const`

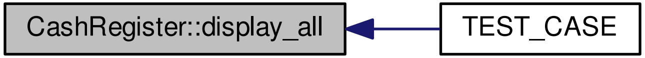
Displays all items in the total sale

Definition at line 40 of file cashregister.cpp.

```
{
    std::cout << "The prices for the items are :\n";
    for (int i = 0;i < price_list.size();i++)
   {
        std::cout << price_list[i] << std::endl;
    }

}
```

Here is the caller graph for this function:



4.1.2.4 get_count()

```
int CashRegister::get_count ( ) const
```

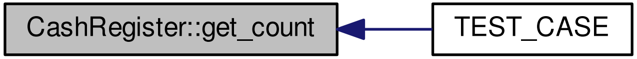
Returns

the item count of the current sale

Definition at line 26 of file cashregister.cpp.

```
27 {
28     //return item_count;
29     int size = price_list.size();
30     return size;
31 }
```

Here is the caller graph for this function:



4.1.2.5 get_total()

```
double CashRegister::get_total ( ) const
```

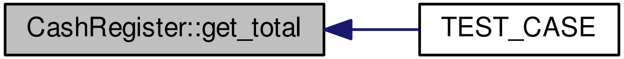
Returns

the total amount of the current sale

Definition at line 15 of file cashregister.cpp.

```
16 {
17    //return total_price;
18    double total = 0;
19    for (int i = 0;i < price_list.size();i++)
20    {
21           total += price_list[i];
22     }
23     return total;
24 }
```

Here is the caller graph for this function:



4.1.3 Member Data Documentation

4.1.3.1 price_list

```
std::vector<double> CashRegister::price_list  [private]
```

Definition at line 36 of file cashregister.h.

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

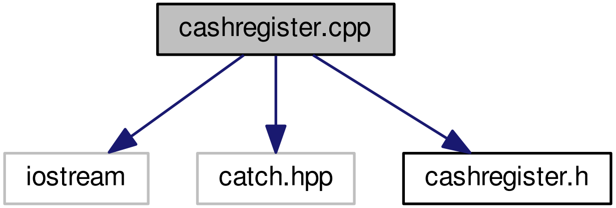
- cashregister.h
- cashregister.cpp

5 File Documentation

5.1 cashregister.cpp File Reference

```
#include <iostream>
#include "catch.hpp"
#include "cashregister.h"
```

Include dependency graph for cashregister.cpp:



Functions

- TEST_CASE ("clear")
- TEST_CASE ("add_item")
- TEST_CASE ("count")
- TEST_CASE ("total")
- TEST_CASE ("display_all")
- TEST_CASE ("output")

5.1.1 Function Documentation

5.1.1.1 TEST_CASE() [1/6]

```
TEST_CASE (
             "clear"  )
```

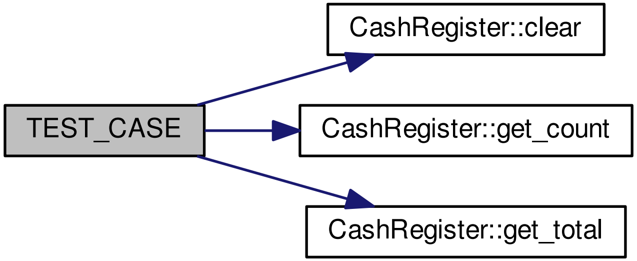
Test clear member function to reset total and item count A simulated cash register that tracks the item count and the total amount due.

```
cashregister.cpp:59: passed: cr.get_count()==0 for: 0 == 0
cashregister.cpp:60: passed: cr.get_total()==0 for: 0.0 == 0
Passed 1 test case with 2 assertions.
```

Definition at line 55 of file cashregister.cpp.

```
56 {
57           CashRegister cr;
58           cr.clear();
59           CHECK(cr.get_count()==0);
60           CHECK(cr.get_total()==0);
61
62 }
```

Here is the call graph for this function:



5.1.1.2 TEST_CASE() [2/6]

```
TEST_CASE (
          "add_item" )
```

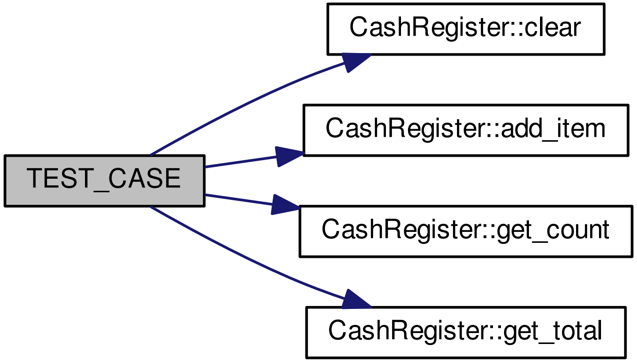
Test add_item member function to check if adding an item with price of 23.25 will change the count to be 1 and total to be 23.25 A simulated cash register that tracks the item count and the total amount due.

```
cashregister.cpp:77: passed: cr.get_count()==1 for: 1 == 1
cashregister.cpp:78: passed: cr.get_total()==23.25 for: 23.25 == 23.25
Passed 1 test case with 2 assertions.
```

Definition at line 71 of file cashregister.cpp.

```
{
    CashRegister cr;
    cr.clear();
    cr.add_item(23.25);
    CHECK(cr.get_count()==1);
    CHECK(cr.get_total()==23.25);
}
```

Here is the call graph for this function:



5.1.1.3 TEST_CASE() [3/6]

```
TEST_CASE (
            "count"  )
```

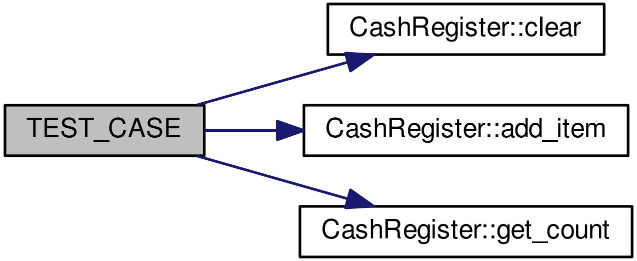
Test get_count() member function to check if adding three items will change the count to be 3 A simulated cash register that tracks the item count and the total amount due.

```
cashregister.cpp:91: passed: cr.get_count()==3 for: 3 == 3
Passed 1 test case with 1 assertion.
```

Definition at line 86 of file cashregister.cpp.

```
87 {
88     CashRegister cr;
89     cr.clear();
90     cr.add_item(23.25);
91     cr.add_item(15.00);
92     cr.add_item(5.98);
93     CHECK(cr.get_count()==3);
94 }
```

Here is the call graph for this function:



5.1.1.4 TEST_CASE() [4/6]

```
TEST_CASE (
             "total" )
```

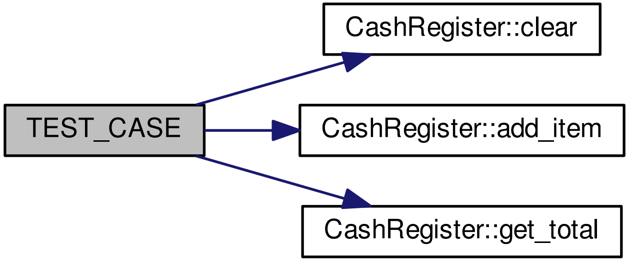
Test get_total() member function to check if adding three items will add up to 85.60 A simulated cash register that tracks the item count and the total amount due.

```
cashregister.cpp:109: passed: cr.get_total()==85.60 for: 85.6 == 85.6
Passed 1 test case with 1 assertion.
```

Definition at line 102 of file cashregister.cpp.

```
{
    CashRegister cr;
    cr.clear();
    cr.add_item(23.25);
    cr.add_item(12.35);
    cr.add_item(50.00);
    CHECK(cr.get_total()==85.60);
}
```

Here is the call graph for this function:



5.1.1.5 TEST_CASE() [5/6]

```
TEST_CASE (
             "display_all"  )
```

Test display_all() member function to check if adding the three prices, 10.99, 25.25, 42.75 were inputted into the vector A simulated cash register that tracks the item count and the total amount due.

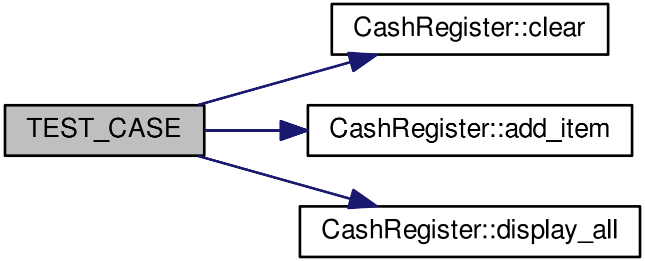
```
cashregister.cpp:132: passed: ss.str() == "The prices for the items are :\n"
"10.99\n25.25\n42.75\n" for: "The prices for the items are :
10.99
25.25
42.75
"
==
"The prices for the items are :
10.99
25.25
42.75
"
Passed 1 test case with 1 assertion.
```

Definition at line 119 of file cashregister.cpp.

```
120 {
121     CashRegister cr;
122     cr.clear();
123     cr.add_item(10.99);
124     cr.add_item(25.25);
125     cr.add_item(42.75);
126     std::streambuf *b = std::cout.rdbuf();
127     std::stringstream ss;
128     std::streambuf *sb = ss.rdbuf();
129     std::cout.rdbuf(sb);
130     // Now all output will be redirected into ss
131     cr.display_all();
132     // set output back to the terminal
133     std::cout.rdbuf(b);
134     CHECK(ss.str() ==
135     "The prices for the items are :\n"
```

```
        "10.99\n25.25\n42.75\n");
}
```

Here is the call graph for this function:



5.1.1.6 TEST_CASE() [6/6]

```
TEST_CASE (
              "output" )
```

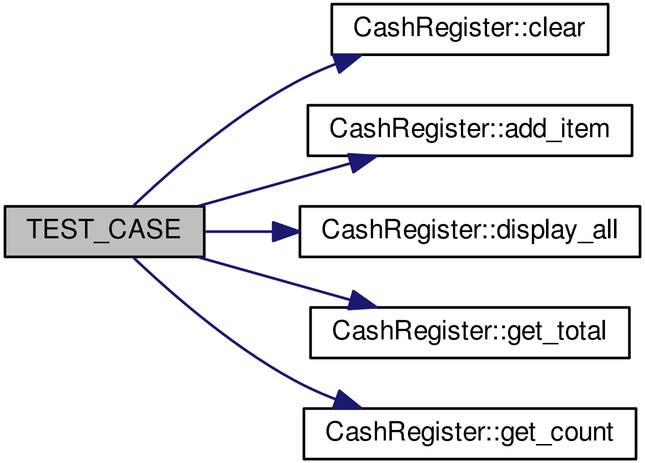
Test clear() display_all() add_item() get_total() get_count() member functions testing output of all functions A simulated cash register that tracks the item count and the total amount due.

```
cashregister.cpp:168: passed: ss.str() == "The prices for the items are :\n"
"14\n23.16\n9.82\n0.98\n80\n13.45\n" "141.41\n""6\n" for: "The prices for the
items are :
14
23.16
9.82
0.98
80
13.45
141.41
6
"
==
"The prices for the items are :
14
23.16
9.82
0.98
80
13.45
141.41
6
"
Passed 1 test case with 1 assertion.
```

Definition at line 145 of file cashregister.cpp.

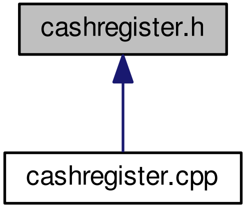
```
{
    CashRegister cr;
    cr.clear();
    cr.add_item(14.00);
    cr.add_item(23.16);
    cr.add_item(9.82);
    cr.add_item(0.98);
    cr.add_item(80.00);
    cr.add_item(13.45);
    std::streambuf *b = std::cout.rdbuf();
    std::stringstream ss;
    std::streambuf *sb = ss.rdbuf();
    std::cout.rdbuf(sb);
    // Now all output will be redirected into ss
    cr.display_all();
    std::cout << cr.get_total() << std::endl;
    std::cout << cr.get_count() << std::endl;
    // set output back to the terminal
    std::cout.rdbuf(b);
    CHECK(ss.str() ==
    "The prices for the items are :\n"
    "14\n23.16\n9.82\n0.98\n80\n13.45\n"
    "141.41\n""6\n");
}
```

Here is the call graph for this function:



5.2 cashregister.h File Reference

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



Classes

- class CashRegister

Index