CS 52 - C++ Programming

-
- Jump to...

You are here

- smcclasses
- / ► FALL12_CS52
- / ► Assignments
- / ► Problem Set 11

Directions:

All programming projects need to be completed and submitted electronically, following the Electronic Submission Guidelines discussed in class. Please cherry-pick from your solution folder and submit just the .cpp files you created as well as .exe file that Visual Studio built from your code.

Background:

This assignment deals with inheritance. Inheritance is one of the major principles of object-oriented programming. In C++, one of the biggest goals is "code reuse". Inheritance accomplishes this. In order to get inheritance working in C++, you must get both the structure of your .h files as well as the implementation of your constructor correct. Constructor implementations must use an initialization list. Please review the book and the online content on these issues so you know how it works before you begin.

Project 1: Inherited TrashCan

The purpose of this assignment is to work with exceptions and inheritance. Enhance the TrashCanclass so that the operators and other functions of the class throw subclassed exceptions when things go wrong, rather than just print out an error message. For example, if you wind up with a TrashCanwith a size value that exceeds its contents, throw a OverflowingTrashCanException, rather than just a "normal" std::logic_error. If you wind up with a negative size or contents value, throw a UnderflowingTrashCanException, rather than just a "normal" std::logic_error.

Please be sure that your subclasses call their parent class constructors.

I'd like you to enhance this class so that invoking its methods or operators potentially throw a custom exception, rather than just a `std::logic_error`. As you may recall, you can create a `logic_error` by passing a string value to its constructor. You say `#include <stdexcept>` to begin working with logic_error.

Logic Error	Hierarchy
	

Implementation Details

TrashCan

```
TrashCan();
TrashCan( int size );
TrashCan( int size, int contents
);

void setSize( intsize );
voidaddItem();

void empty( );
void cover( );
void uncover( );

void printCan( );
```

```
bool myIsCovered;
intmy_Size;
intmy_Contents;
```

TrashCan

Sample Driver

```
TrashCan t;
TrashCan yourCan( 128 );
TrashCan junk( 12 );

t.setSize( 3 );
t.addItem();
t.addItem();
t.addItem();
t.cover();

/// how can you have negative number of items
/// TrashCan???
try {
junk = yourCan - t;
cout << junk;
} catch( UnderflowingTrashCanException ) {
cout << "underflow caught..." << endl;
} catch( std::logic_error le ) {
cout << "subtraction failed" << endl;
}


/// how can a contents of 4 items in a TrashCan
/// which has only has size 3??/
try {
t.addItem( );
} catch( OverflowingTrashCanException ) {
cout << "overflow caught..." << endl;
} catch( std::logic_error le ) {
cout << "addItem failed" << endl;
}

//// t should be unchanged
cout << t << endl;

/// how can you have TrashCans with a negative
/// size??
try {
TrashCan myCan( -19 );
} catch( UnderflowingTrashCanException )
cout << "underflow caught..." << endl;
} catch( std::logic_error le ) {
cout << "constructor failed!" << endl;
}
```

```
TrashCan * ptrCan = new TrashCan( 25, 3 );
TrashCan * nullCan = NULL;

// be careful
cout << nullCan << endl;
cout << ptrCan << endl;

cin >> ptrCan;
// notice the difference
cout << ptrCan << endl;
ptrCan->addItem( );
/// notice the difference
cout << ptrCan << endl;

delete (ptrCan);
// why can't I do this???
// delete( nullCan );
```

PLEASE DON'T SEND ME THE ENTIRE VISUAL STUDIO SOLUTION FOLDER. Instead, as you'll do with every homework assignment, please cherry-pick from your hard-drive. Select just the source (.cpp) file and the executable (.exe) file. You will find these file in your Solution Folder's Debug/Release folder. Copy these two files together into a folder named with your name and student ID. Zip this folder. Then upload this file into the Assignment 16 submission area by clicking on the "Browse..." button shown below. Then click "Upload this file".

Click here if you want to review step-by-step directions on electronic submission for this class.

Available from: Friday, 6 April 2012, 08:45 PM
Due date: Monday, 3 December 2012, 10:55 PM

Submission

No files submitted yet

Upload a file (Max size: 8MB)

Upload this file

Notes

No entry

Edit

You are logged in as DAVID BROWN (Logout)
FALL12_CS52