PROJECT 4 - SAMPLE TEST MAIN() FUNCTIONS

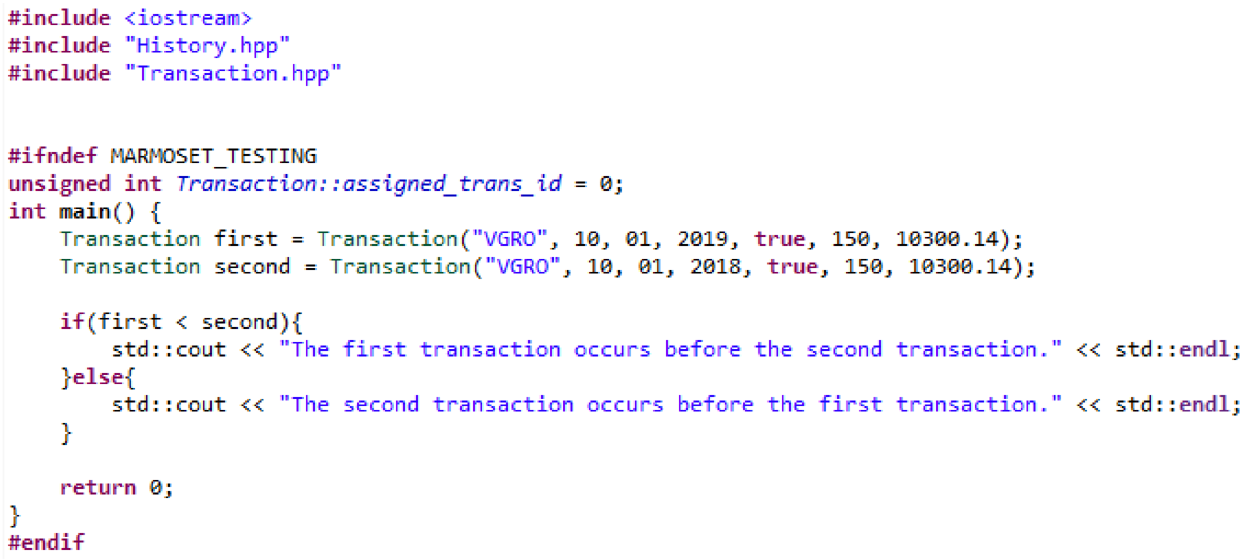
ECE 150 Fall 2020

Description

In project 4 you are required to write functions for the two classes **Transaction** and **History**. This document provides examples on how to test your functions with int main().Using these templates, you should be able to develop more complex tests to make sure your functions are operating correctly.

Transaction Example

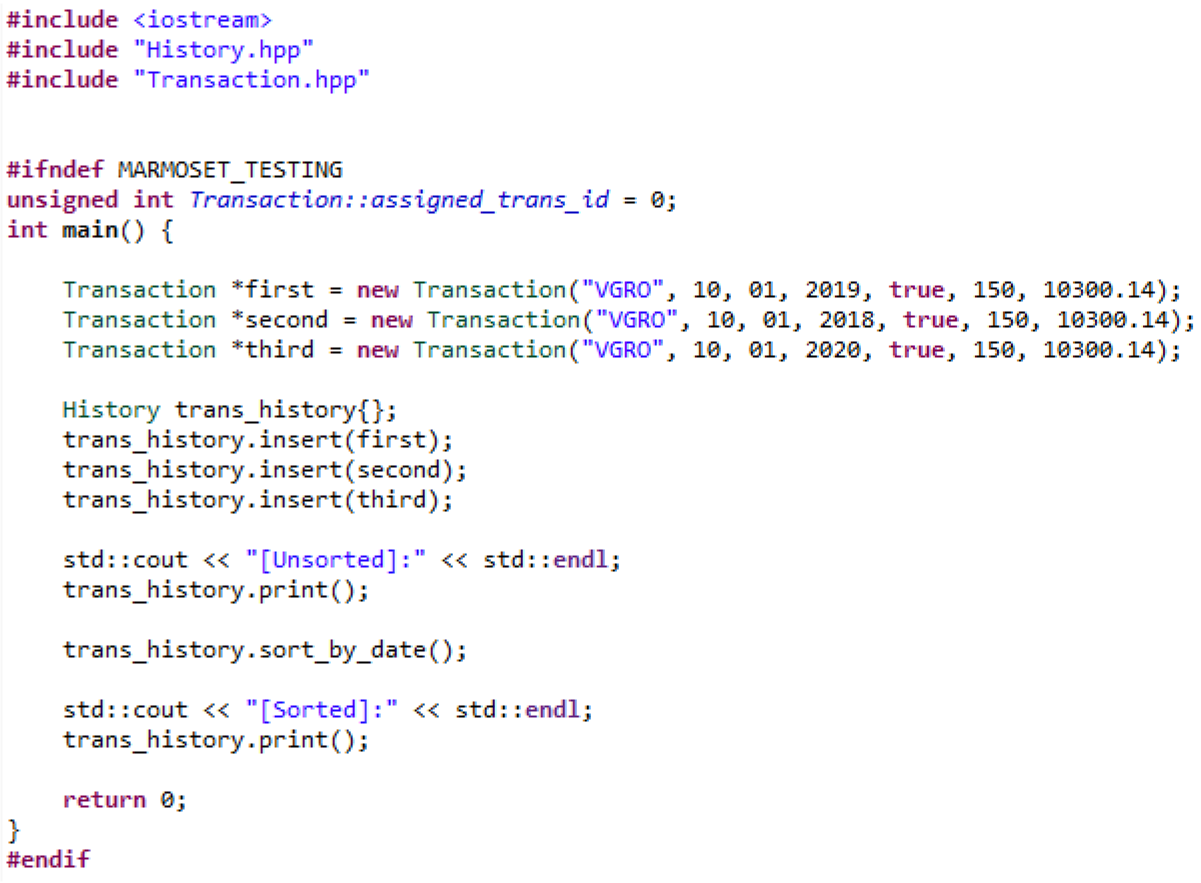
The program below compares two different transactions to determine which one occurs first. Firstly, the transactions are created with different years (2019 and 2018). Afterwards the overloaded < operator is used to compare the different objects. The result will then be printed to the console.



```
#include <iostream>
#include "History.hpp"
#include "Transaction.hpp"


#ifndef MARMOSET_TESTING
unsigned int Transaction::assigned_trans_id = 0;
int main() {
    Transaction first = Transaction("VGRO", 10, 01, 2019, true, 150, 10300.14);
    Transaction second = Transaction("VGRO", 10, 01, 2018, true, 150, 10300.14);

    if(first < second){
        std::cout << "The first transaction occurs before the second transaction." << std::endl;
    }else{
        std::cout << "The second transaction occurs before the first transaction." << std::endl;
    }

    return 0;
}
#endif
```

History Example

The program below is used to show the transaction history before and after sorting by date. Three different transactions are created with different years (2019, 2018 and 2020). The three transactions are inserted into the history out of order. The transaction history is printed (unsorted) then the transaction history is sorted and printed.



```cpp
#include <iostream>
#include "History.hpp"
#include "Transaction.hpp"


#ifndef MARMOSET_TESTING
unsigned int Transaction::assigned_trans_id = 0;
int main() {

    Transaction *first = new Transaction("VGRO", 10, 01, 2019, true, 150, 10300.14);
    Transaction *second = new Transaction("VGRO", 10, 01, 2018, true, 150, 10300.14);
    Transaction *third = new Transaction("VGRO", 10, 01, 2020, true, 150, 10300.14);

    History trans_history{};
    trans_history.insert(first);
    trans_history.insert(second);
    trans_history.insert(third);

    std::cout << "[Unsorted]:" << std::endl;
    trans_history.print();

    trans_history.sort_by_date();

    std::cout << "[Sorted]:" << std::endl;
    trans_history.print();

    return 0;
}
#endif
```