```
#ifndef _ACCOUNT_HPP_
#define _ACCOUNT_HPP_

#include <vector>
#include <algorithm>
#include <iterator>

#include "transaction.hpp"

#define BEGIN_ACCOUNT "BACCT"

/**
 * This class defines a money account which stores financial transactions
 * and maintains a balance.
 */
class Account {
  std::string name;
  std::vector<Transaction> transactions;

  public :
    Account() : name({}), transactions({}) {}
    Account(const std::string& name) : name(name), transactions({}) {}

    const std::string& getName() const {
      return name;
    }

    const std::vector<Transaction>& getTransactions() const {
      return transactions;
    }

    void addTransaction(Transaction& t) {
      transactions.push_back(t);
    }

    long double balance() const {
      long double sum = 0;
      sum = std::accumulate(std::begin(transactions), std::end(transactions), sum,
          [](const long double& a, const Transaction& b) {
            return a + b.getAmount();
          });

      return sum;
    }

    friend std::ostream& operator<<(std::ostream& os, const Account& a);
    friend std::istream& operator>>(std::istream& is, Account& a);
};

std::ostream& operator<<(std::ostream& os, const Account& a) {
  os << BEGIN_ACCOUNT << " "
    << a.name << "\n";

  return os;
}

std::istream& operator>>(std::istream& is, Account& a) {
  is >> a.name;
  return is;
}

#endif
```