```
#ifndef _TRANSACTION_HPP_
#define _TRANSACTION_HPP_

#include <string>
#include <ctime>
#include <iostream>

#define BEGIN_TRANSACTION "BTRANS"

/**
 * This is a simple helper functions that will trim off the leading and
 * trailing white space characters from a string
 * @param[in,out] s - The string to trim
 * @return - s
 */
std::string& trim(std::string& s) {
  s.erase(0, s.find_first_not_of(" \t\n"));
  s.erase(s.find_last_not_of(" \t\n") + 1);
  return s;
}

/**
 * This class defines a single financial transaction.
 */
class Transaction {
  std::string accountName;
  long double amount;
  std::string memo;
  std::time_t transactionDate;

  public :
    Transaction() : accountName({}), amount{}, memo({}), transactionDate(time(0)) {}

    Transaction(const std::string& accountName, long double amount, const std::string&
 memo) :
      accountName(accountName), amount(amount), memo(memo),
      transactionDate(std::time(0)) {}

    const std::string& getAccountName() const {
      return accountName;
    }

    long double getAmount() const {
      return amount;
    }

    const std::string& getMemo() const {
      return memo;
    }

    time_t getTransactionDate() const {
      return transactionDate;
    }

    friend std::ostream& operator<<(std::ostream& os, const Transaction& t);
    friend std::istream& operator>>(std::istream& is, Transaction& t);

};

std::ostream& operator<<(std::ostream& os, const Transaction& t) {
  os << BEGIN_TRANSACTION << " "
    << t.accountName << " "
    << t.transactionDate << " "
    << t.amount << " "
    << t.memo << "\n";

  return os;
}
```

```
std::istream& operator>>(std::istream& is, Transaction& t) {
  is >> t.accountName;
  is >> t.transactionDate;
  is >> t.amount;
  std::getline(is, t.memo);

  trim(t.memo);

  return is;
}

#endif
```