

Homework #4 – Minimizing Compile-time Dependencies

(10 points) Minimize the compile-time dependencies of the following header file:

```
// CheckingAccount.h
#ifndef CHECKINGACCOUNT_H
#define CHECKINGACCOUNT_H

#include <iostream>
#include <list>
#include <string>
#include <vector>
using namespace std;

#include "AccountExceptions.h" // class OverdrawException
#include "BankAccount.h"      // class BankAccount
#include "Check.h"            // class Check

class CheckingAccount : public BankAccount
{
public:
    CheckingAccount(long accountNumber,
                    double initialBalance,
                    const string &firstName,
                    const string &lastName);
    ~CheckingAccount();
    double getBalance() const;
    void depositCheck(const Check &);
    void depositCash(double);
    void withdrawCash(double);
    virtual ostream &print(ostream &) const;

protected:
    void processAccount();

private:
    long accountNumber_;
    double balance_;
    string firstName_;
    string lastName_;
    vector<double> deposits_;
    vector<double> withdrawals_;
    list<Check> checks_;
};

ostream &operator<<(ostream &, const CheckingAccount &);

#endif
```

Your solution should show:

- CheckingAccount.h converted to have minimized compile-time dependencies
- The portion of CheckingAccount.cpp showing the #includes and the CheckingAccountImpl class definition
 - You do not need to show anything else from CheckingAccount.cpp

Hint: See Items 26 through 29 from the book for details on how to solve this problem.

Place all source code in a single PDF document. Submit this document.