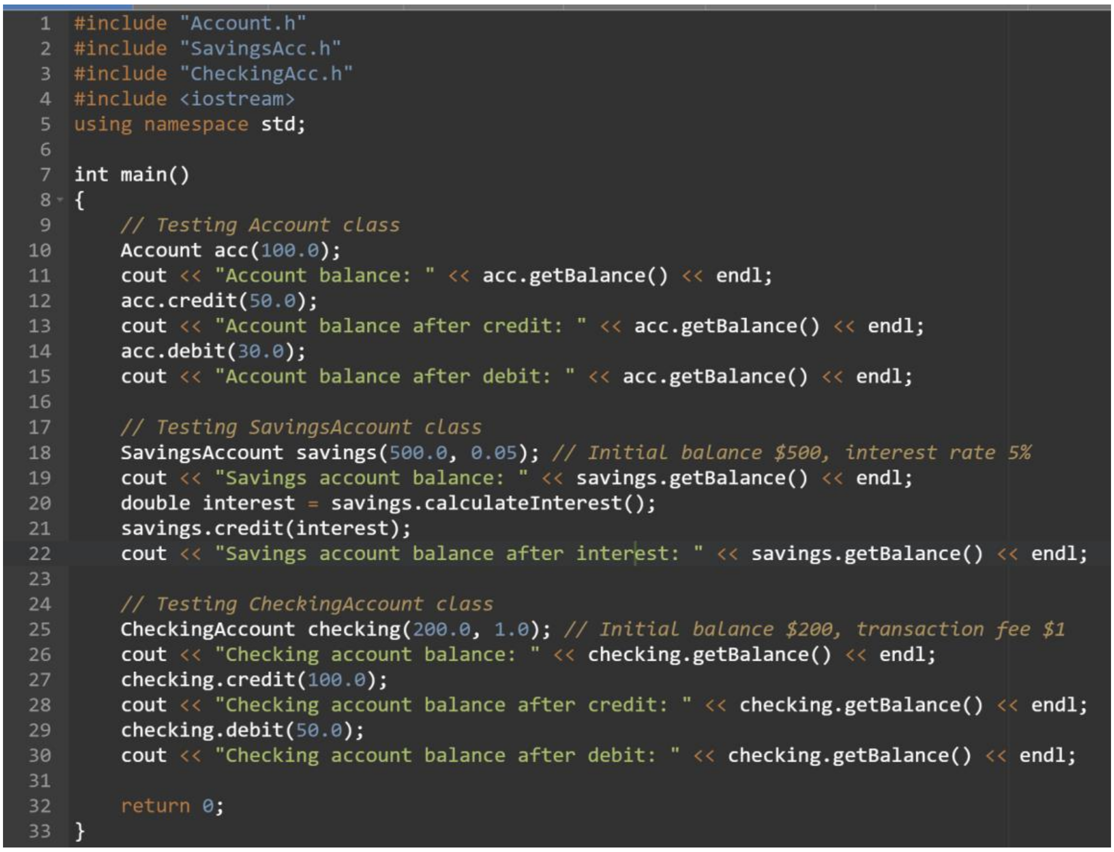
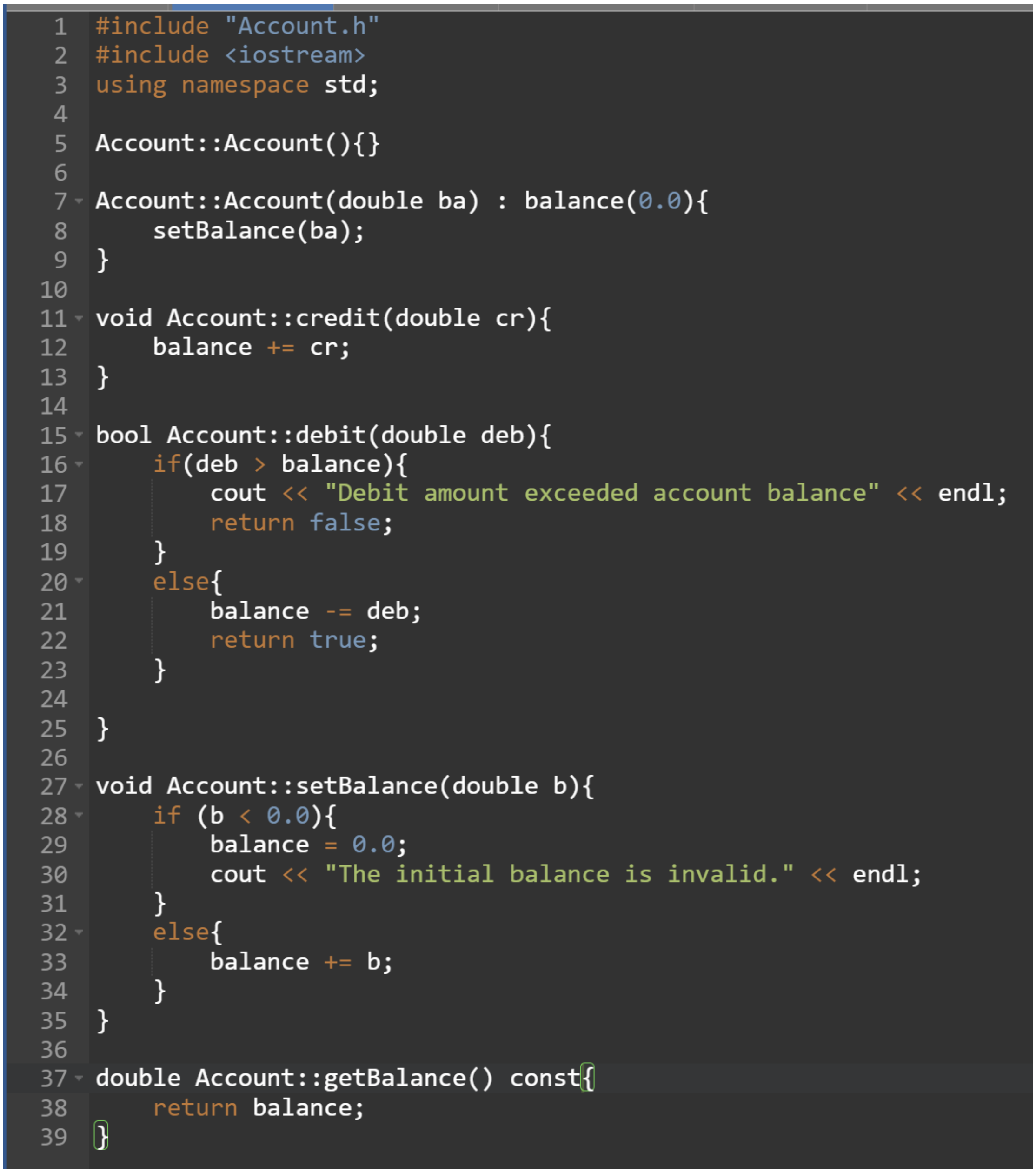
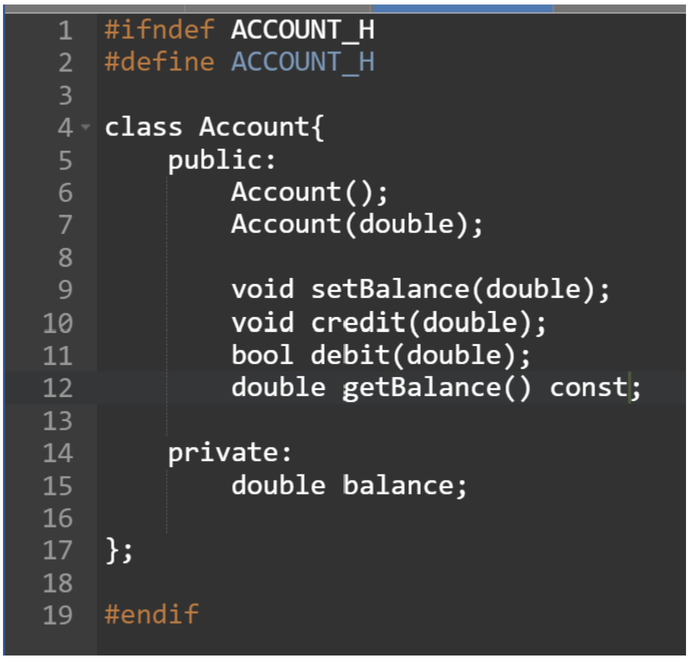


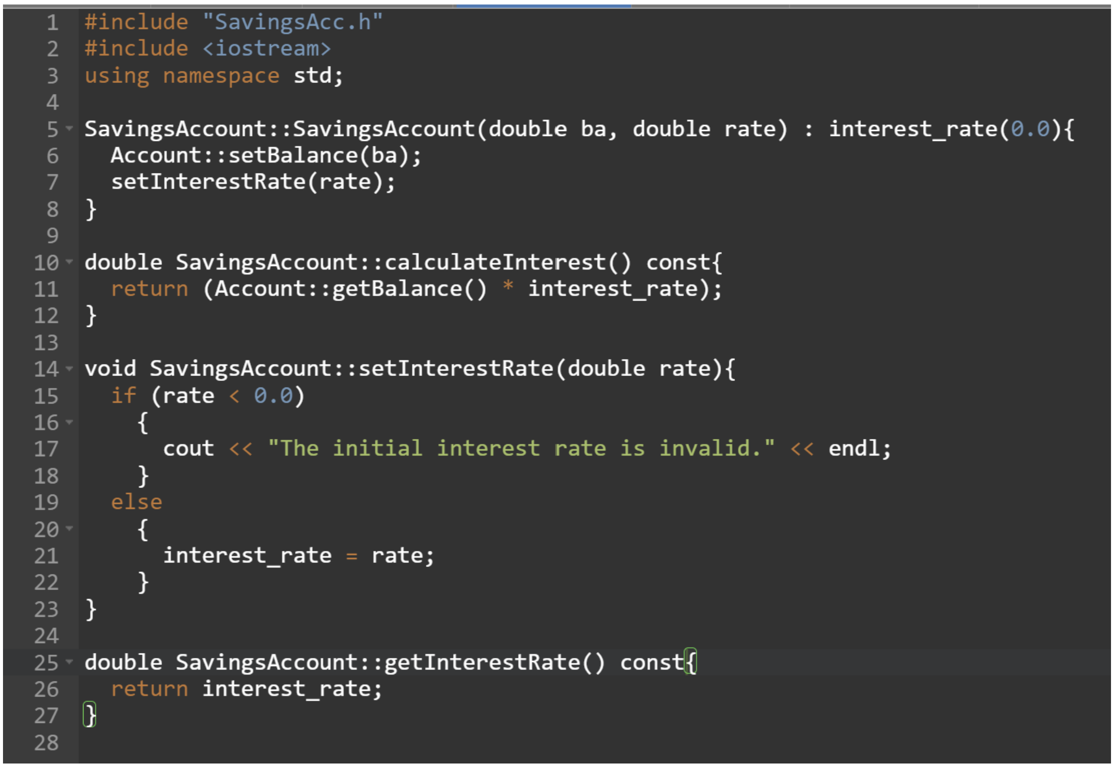
```cpp
#include "Account.h"
#include "SavingsAcc.h"
#include "CheckingAcc.h"
#include <iostream>
using namespace std;

int main()
{
    // Testing Account class
    Account acc(100.0);
    cout << "Account balance: " << acc.getBalance() << endl;
    acc.credit(50.0);
    cout << "Account balance after credit: " << acc.getBalance() << endl;
    acc.debit(30.0);
    cout << "Account balance after debit: " << acc.getBalance() << endl;

    // Testing SavingsAccount class
    SavingsAccount savings(500.0, 0.05); // Initial balance $500, interest rate 5%
    cout << "Savings account balance: " << savings.getBalance() << endl;
    double interest = savings.calculateInterest();
    savings.credit(interest);
    cout << "Savings account balance after interest: " << savings.getBalance() << endl;

    // Testing CheckingAccount class
    CheckingAccount checking(200.0, 1.0); // Initial balance $200, transaction fee $1
    cout << "Checking account balance: " << checking.getBalance() << endl;
    checking.credit(100.0);
    cout << "Checking account balance after credit: " << checking.getBalance() << endl;
    checking.debit(50.0);
    cout << "Checking account balance after debit: " << checking.getBalance() << endl;

    return 0;
}
```



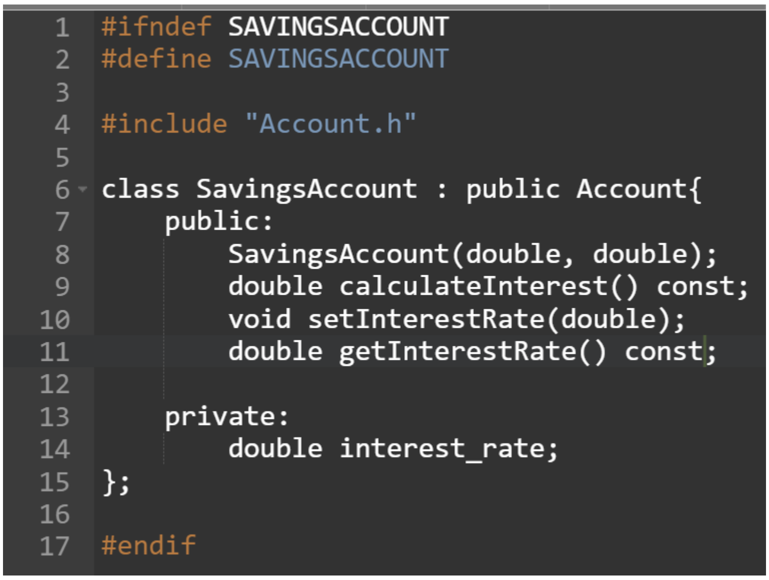
```cpp
#include "Account.h"
#include <iostream>
using namespace std;

Account::Account(){}

Account::Account(double ba) : balance(0.0){
    setBalance(ba);
}

void Account::credit(double cr){
    balance += cr;
}

bool Account::debit(double deb){
    if(deb > balance){
        cout << "Debit amount exceeded account balance" << endl;
        return false;
    }
    else{
        balance -= deb;
        return true;
    }

}

void Account::setBalance(double b){
    if (b < 0.0){
        balance = 0.0;
        cout << "The initial balance is invalid." << endl;
    }
    else{
        balance += b;
    }
}

double Account::getBalance() const{
    return balance;
}
```



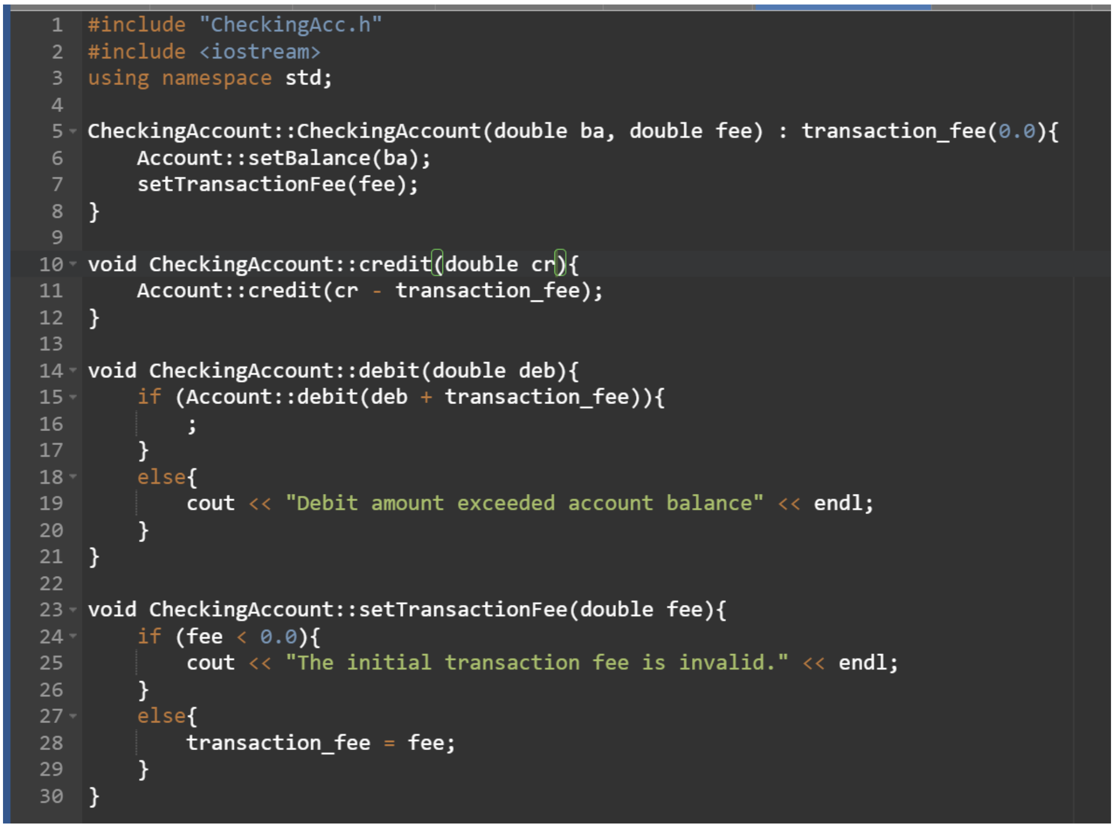
```cpp
#ifndef ACCOUNT_H
#define ACCOUNT_H

class Account{
    public:
        Account();
        Account(double);

        void setBalance(double);
        void credit(double);
        bool debit(double);
        double getBalance() const;

    private:
        double balance;

};

#endif
```



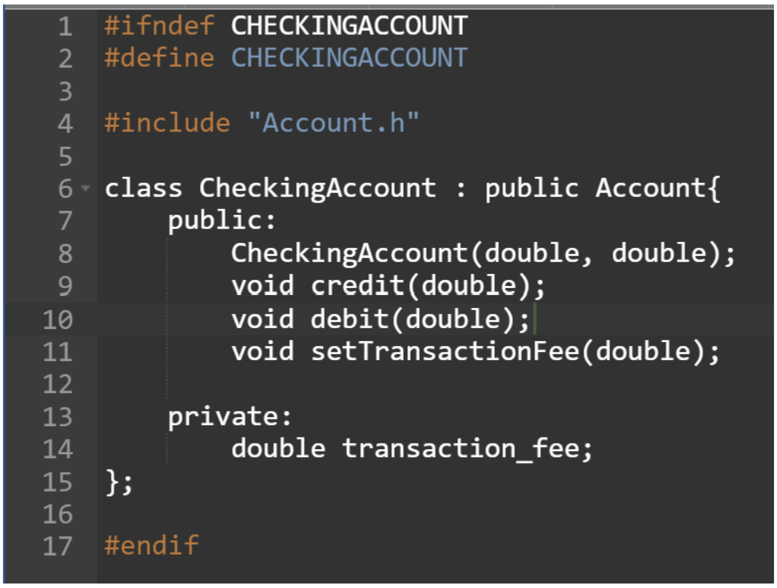
```cpp
#include "SavingsAcc.h"
#include <iostream>
using namespace std;

SavingsAccount::SavingsAccount(double ba, double rate) : interest_rate(0.0){
  Account::setBalance(ba);
  setInterestRate(rate);
}

double SavingsAccount::calculateInterest() const{
  return (Account::getBalance() * interest_rate);
}

void SavingsAccount::setInterestRate(double rate){
  if (rate < 0.0)
    {
      cout << "The initial interest rate is invalid." << endl;
    }
  else
    {
      interest_rate = rate;
    }
}

double SavingsAccount::getInterestRate() const{
  return interest_rate;
}

```



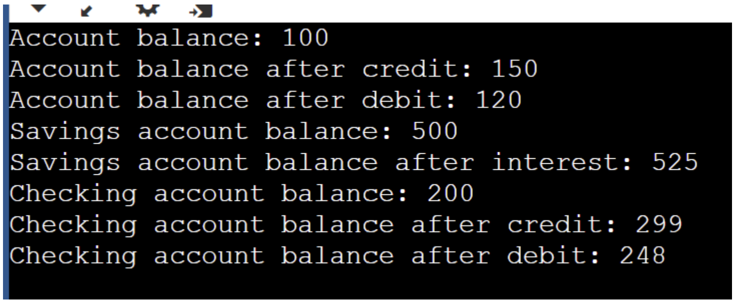
```cpp
#ifndef SAVINGSACCOUNT
#define SAVINGSACCOUNT

#include "Account.h"

class SavingsAccount : public Account{
    public:
        SavingsAccount(double, double);
        double calculateInterest() const;
        void setInterestRate(double);
        double getInterestRate() const;

    private:
        double interest_rate;
};

#endif
```



```cpp
#include "CheckingAcc.h"
#include <iostream>
using namespace std;

CheckingAccount::CheckingAccount(double ba, double fee) : transaction_fee(0.0){
    Account::setBalance(ba);
    setTransactionFee(fee);
}

void CheckingAccount::credit(double cr){
    Account::credit(cr - transaction_fee);
}

void CheckingAccount::debit(double deb){
    if (Account::debit(deb + transaction_fee)){
        ;
    }
    else{
        cout << "Debit amount exceeded account balance" << endl;
    }
}

void CheckingAccount::setTransactionFee(double fee){
    if (fee < 0.0){
        cout << "The initial transaction fee is invalid." << endl;
    }
    else{
        transaction_fee = fee;
    }
}
```



```cpp
#ifndef CHECKINGACCOUNT
#define CHECKINGACCOUNT

#include "Account.h"

class CheckingAccount : public Account{
    public:
        CheckingAccount(double, double);
        void credit(double);
        void debit(double);
        void setTransactionFee(double);

    private:
        double transaction_fee;
};

#endif
```



```
Account balance: 100
Account balance after credit: 150
Account balance after debit: 120
Savings account balance: 500
Savings account balance after interest: 525
Checking account balance: 200
Checking account balance after credit: 299
Checking account balance after debit: 248
```