

**Computer Science Department  
San Francisco State University  
CSC 340  
Spring 2016**

## **Assignment 4 - Inheritance and Polymorphism**

### **Due Date**

Wednesday, April 6, at midnight.

### **Overview**

The purpose of this project is to begin to make use of inheritance in our budget application by encapsulating deposit and withdrawal functionality into a base class named **BudgetItem**, and beginning to create an inheritance hierarchy to leverage and refine this behavior, starting with the **Account** class.

### **Submission**

See the submission guidelines posted on iLearn.

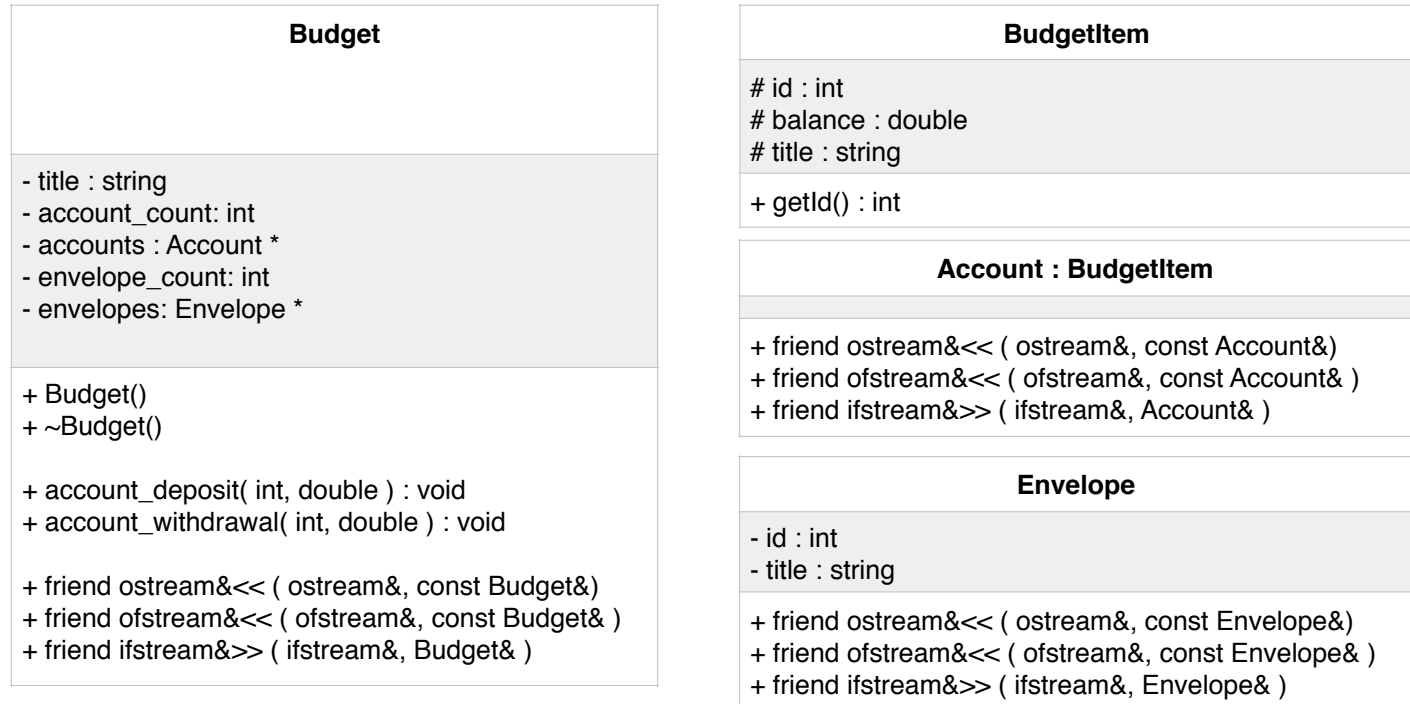
### **Requirements**

1. Create **BudgetItem** as an abstract base class that requires the following methods:  
**deposit**, and **withdraw**, and manages the **balance** (field formerly in **Account**)
  - 1.1. **void deposit( double amount )** - increments the object's **balance**
  - 1.2. **virtual bool withdraw( double amount ) = 0** - will do different things in derived classes.
2. Update **Account** to be a derived class of **BudgetItem**.
  - 2.1. Implement the required virtual method, whose implementation decrements the object's **balance**. If the withdrawal results in a negative **balance**, return **false**, otherwise return **true**. Note that the withdrawal should still be performed! (You can overdraw your account.)
  - 2.2. *Note: Next project will see the addition of another type that will need different withdrawal behavior.*
3. Update **Budget** with the following methods:
  - 3.1. **void account\_deposit( int account\_id, double amount )** - Performs a deposit on the account **with the specified id** (note that the id is not the index in the accounts array!).
  - 3.2. **bool account\_withdrawal( int account\_id, double amount )** - Performs a withdrawal on the account **with the specified id** (note that the id is not the index in the accounts array!).
4. Update the main method, and any processing methods, to make use of the new **Account** abilities (via the **Budget**'s **account\_deposit** and **account\_withdrawal**). (See the output for requirements.)

## Resources

The makefile for this assignment can be found at <https://gist.github.com/jrob8577/446caf0e2dbd500f5e2676774c6496cb>.

## Appendix A: UML Diagrams



## Appendix B: Budget File Format

File Format	Sample Data
TITLE	My Budget
ENVELOPE_COUNT	2
ENVELOPE_ID ENVELOPE_TITLE	1 Rent
// Repeated up to ENVELOPE_COUNT	2 Groceries
ACCOUNT_COUNT	2
ACCOUNT_ID ACCOUNT_TITLE	345 Checking 10.00
ACCOUNT_BALANCE	876 Savings 100.00
// Repeated up to ACCOUNT_COUNT	

Note the output format of the decimals (precision is 2, decimal point shown).

## Sample Output

User input is *italicized, in bold*.

```
➤ make clean; make
rm -f *.o driver
g++ -std=c++11 -c -Wall main.cpp
g++ -std=c++11 -c -Wall budget.cpp
```

```
g++ -std=c++11 -c -Wall budget_item.cpp
g++ -std=c++11 -c -Wall account.cpp
g++ -std=c++11 -c -Wall envelope.cpp
g++ -std=c++11 main.o budget.o budget_item.o account.o envelope.o -o
driver
```

▷ **cat budget**

```
NoSpacesInTitleBudget
2
123 WoW
456 Groceries
2
789 Checking 100.00
120 Savings 70.68
```

▷ **./driver**

```
1. Show Budget
2. Make Deposit
3. Make Withdrawal
0. EXIT
Enter your selection: 1
```

----- NoSpacesInTitleBudget -----

```
2 Accounts:
789: Checking - 100.00
120: Savings - 70.68
```

```
2 Envelopes:
123: WoW
456: Groceries
```

```
1. Show Budget
2. Make Deposit
3. Make Withdrawal
0. EXIT
Enter your selection: 2
```

```
Enter account ID: 789
Enter deposit amount: 123.45
```

```
1. Show Budget
2. Make Deposit
3. Make Withdrawal
0. EXIT
Enter your selection: 1
```

----- NoSpacesInTitleBudget -----

```
2 Accounts:
789: Checking - 223.45
120: Savings - 70.68
```

```
2 Envelopes:
123: WoW
456: Groceries
```

1. Show Budget
2. Make Deposit
3. Make Withdrawal
0. EXIT

Enter your selection: **3**

Enter account ID: **120**

Enter withdrawal amount: **100**

1. Show Budget
2. Make Deposit
3. Make Withdrawal
0. EXIT

Enter your selection: **1**

----- NoSpacesInTitleBudget -----

2 Accounts:

789: Checking - 223.45

120: Savings - -29.32

2 Envelopes:

123: WoW

456: Groceries

1. Show Budget
2. Make Deposit
3. Make Withdrawal
0. EXIT

Enter your selection: **0**

▷ **cat budget**

NoSpacesInTitleBudget

2

123 WoW

456 Groceries

2

789 Checking 223.45

120 Savings -29.32