**Lab 09 C++ Classes**

**Objectives**

* Learn to design and implement C++ classes

**Review**

A ***class*** is a language feature that encourages good programming style by allowing the user to encapsulate both data and actions into a single object.

Because classes are used to encapsulate abstract data types and because the separation of the logical properties of an ADT from the implementation is so important, you should put the class declaration and the implementation in different files. The class declaration should be in a file with a "***.h***" extension (called the specification file), and the implementation should be in a file with the same name but with a "***.cpp***" extension (called the implementation file). The implementation file must use the ***#include*** directive to access the specification file. Any client programs must use the ***#include*** directive for the specification file in order to include the class in their source code.

For today's lab, the specification file is named *money.h*. Name your implementation file ***money.cpp***.

**Resources**

[Online reference material](http://www.cplusplus.com/doc/tutorial/basic_io.html)

**Today's Task**

Consider the banking community. The primary data type for the banking community is not an *int* or a *double*, but some form of *money*. You are to implement a class, called **Money**.

An object of the ***Money*** class represents amounts of U.S. money as two integers. For example, $4585.72 will be represented as 4585 and 72, with 4585 being stored in the dollars field and 72 in the cents field. The *dollars* and *cents* data members should be declared *private*.

Declare any constants that your program uses.

The class should provide the following public member functions:

Money( ) - initializes all data members to zero.

Money(double amount) - if amount is negative, the function sets the

data member to zero; otherwise it stores amount as dollars and

cents.

Money(int dollars, int cents) - although you should always validate

user input, we will omit validation of input for this function

for the sake of time. the function sets the data members to the

given values.

void set (int dollars, int cents) - again we will omit validation.

the function sets the data members to the given values.

int getDollars( ) - returns the value in the dollars field.

int getCents( ) - returns the value in the cents field.

int valueInCents( ) - returns the numeric value of the money in

cents. For example, the function should return 590 for

5 dollars and 90 cents.

double dollarsAndCents( ) - returns the total amount of money in decimal

format. For example, the function should return 2.44 for

2 dollars and 44 cents.

Money add(Money otherAmount) - returns a Money object which represents

the sum of the money in this object and otherAmount. For example,

if this object has 5 dollars and 75 cents and otherAmount has 9

dollars and 40 cents, then the newly created Money object will

have 15 dollars and 15 cents.

**Getting the files for today's lab**

1. Create a **"lab9"** directory
2. Use *cd lab9* to move to that directoyr
3. copy the necessary files to the directory using:

lab1071copy 9

Your directory should now contain the files: *main.cpp, makefile,*and *money.h*

**Procedure**

You should develop your class incrementally. Develop and test one function at a time. You can stub the other methods until the one being developed is working.

**Testing**

Test your code using.

make   
./lab9

You should add additional code to test those methods that are not tested in main.cpp

**Submission**

You are to submit two files:

1. The implementation file *money.cpp* will provide the function definitions for all the class member functions.
2. The test driver, *main.cpp* should contain code to test the additional methods of the class.

Submit your files using [Handin](https://handin.cs.clemson.edu/)