1. Write a class, called Coins, that is a holder for a variety of coins (similar to a pocket or coin purse). This has no way to split coins or to receive change back from a transaction, so it must pay with exact change. I wrote a simple main, put in a file called simple\_main.cpp, for you to use to test this first part of the homework assignment.
// Coins.h /// The name of this file.
#include <iostream>

using namespace std;
const int CENTS\_PER\_QUARTER = 25, CENTS\_PER\_DIME = 10, CENTS\_PER\_NICKEL = 5;
class Coins
{
public:
 Coins( int q, int d, int n, int p );
 void deposit\_coins( Coins & c ); // **note remove the coins from c and put them into this Coins object**
 bool has\_exact\_change\_for\_cents( int amount\_in\_cents );
 Coins extract\_exact\_change\_for\_cents( int amount\_in\_cents );
 int total\_value\_in\_cents();
 void print( ostream & out ); // “6 quarters, 2 dimes, …”
private:
 **int quarters, dimes, nickels, pennies;**
};
ostream & operator << ( ostream & out, Coins c );
// simple\_main.cpp /// The name of this file
#include "Coins.h"
const int CENTS\_FOR\_CANDYBAR = 482;
int main()
{
/// The first line creates a Coins object called 'pocket.'
 Coins pocket( 100, 10, 10, 100 );
 cout << "I started with " << pocket << " in my pocket" << endl;
/// This line creates a Coins object called payForCandy and initializes it.
 Coins payForCandy = pocket.extract\_exact\_change\_for\_cents( CENTS\_FOR\_CANDYBAR );
 cout << "I bought a candy bar for " << CENTS\_FOR\_CANDYBAR
 << " cents using " << payForCandy << endl;
 cout << "I have " << pocket << " left in my pocket" << endl;
 return 0;
}

2. Write a better main, in a file named coins\_main.cpp, that does the following scenario:

* + Create two Coins objects:
	pocket: 5 q, 3 d, 6 n, 8 p
	piggyBank: 50 q, 50 d, 50 n, 50 p
	+ Buy a bag of chips that costs 68 cents taking the coins from your pocket. Display what's left in your pocket.
	+ Transfer $2.05 from your piggyBank to your pocket. Display how much you now have in both.
	+ While vacuuming, you find loose change in your sofa 10 q, 10d, 10n, 10p. Put it in your piggyBank. Display how much you now have in your piggyBank. Use the format $xx.xx
	+ To enter the amount found in your sofa, create an object with the number of coins found, then call deposit\_coins.

3. Write a third main, in a file called coins\_menu.cpp, that presents a (text-based) menu interface to your program that allows the user to deposit change, extract change, and to print the balance in a single coins object called myCoins. Be sure you don't allow someone to extract more money than they have.