CSCI 135 Software Design and Analysis, C++
Lab 7

Saad Mneimneh
Hunter College of CUNY

Lab A: PiggyBank
Consider the following class:

```
class PiggyBank {
  int a[4];
  //a[0] quarters
  //a[1] dimes
  //a[2] nickels
  //a[3] pennies

  bool breakable; //false by default
  bool emptied; //initially false,
                //set to true the first time piggy bank is emptied

 public:
  PiggyBank() {...}
  PiggyBank(int cents) {...}
  PiggyBank(int cents, bool breakable) {...}
  void add(int cents) {...}
  PiggyBank add(PiggyBank& b) {...} //by reference to empty b
  int empty() {...}
  void print() {...}
};
```

(a) Implement the default constructor to create an empty piggy bank, i.e. with 0 cents.

(b) Implement the second and third constructors to create a piggy bank with the specified amount of cents by first using as many quarters as possible, then as many dimes as possible, then as many nickels as possible, then pennies. We call this the fewest coins condition. In addition, the third constructor determines whether the piggy bank must be broken in order to get the money.

(c) Implement the add function to add the specified amount of cents to the piggy bank and rearrange to reestablish the fewest coins condition.

(d) Implement the empty function to return the number of cents contained in the piggy bank and reset the piggy bank to empty. If the piggy bank must be broken, the piggy bank cannot be used later, i.e. not matter what functions are called it will always contain 0 cents (used emptied to keep track).

(e) Implement the second add function that takes another piggy bank as parameter and consolidate the two into a third one while reestablishing the fewest coins condition. In addition, the two piggy banks should be emptied (see above).

(f) Implement the print function to output information about the piggy bank, i.e. the number of quarters, the number of dimes, the number of nickels, and the number of pennies.