CPP Coding Problem

Subject: Store

Contributor: 林鼎傑,鐘良軒

Main testing concept: Overloading.

Basics	Functions
☐ C++ BASICS	☐ SEPARATE COMPILATION AND NAMESPACES
☐ FLOW OF CONTROL	☐ STREAMS AND FILE I/O
☐ FUNCTION BASICS	☐ RECURSION
☐ PARAMETERS AND OVERLOADING	☐ INHERITANCE
☐ ARRAYS	■ POLYMORPHISM AND VIRTUAL FUNCTIONS
☐ STRUCTURES AND CLASSES	☐ TEMPLATES
☐ CONSTRUCTORS AND OTHER TOOLS	☐ LINKED DATA STRUCTURES
■ OPERATOR OVERLOADING, FRIENDS, AND REFERENCES	☐ EXCEPTION HANDLING
☐ STRINGS	☐ STANDARD TEMPLATE LIBRARY
☐ POINTERS AND DYNAMIC ARRAYS	☐ PATTERNS AND UML

Description:

In stores, sometimes there will be some combination offers, please implement a program that can generate a receipt.

- Please design a class named "Store" and implement the following methods:
  - **Store()**
    Construct empty "Store".
  - **Add(Product product)**
    Add "Product" to the "Store", "Product" is unique.
  - **Add(Combo combo)**
    Add "Combo" to "Store", "Combo" is unique.
    If the following error occurs, print the message and cancel the operation, i.e., only print the first occurred error. By the way, "Product" check is earlier than price check.
    If "Product" is in "Combo" but not in "Store", print **"Product not exist."**.
    If "Combo" has no set price, print **"The combination has no set price."**.
  - **Buy(const char* name)**
    Buy "Product" or "Combo" by string,
    If "Product" or "Combo" not in "Store", print "{name} is not in store." (e.g., Apple is not in store.)
    The name of "Product" is its name.
    The name of "Combo" contains name of "Product" and each "Product" will be separated by "+".
  - **PrintReceipt()**
    Print receipt by shopping list with 20 character width.
    Shopping list is arranged in shopping order, and then the "Product" is combined into a "Combo" with the largest discount, i.e., the lowest payment (if more than one "Product" can be combined, the first "Product" is selected), and the combo order is the highest order of combined products.

Receipt starts with " Receipt ", followed by 20 "=" after line breaks.
Receipt ends with 20 "=", followed by "Total {Total}".
Product will be printed like "{Name} {Price}".
Combo will print it contains "Product" in ascending, and how much discount is it, the output like following next paragraph.

```
Apple          10
Ball           20
Discount      -10
```

After printing the receipt, clear the shopping list.
"Name" will be left-aligned; "Price" will be right-aligned.

- Please design a class named "Product" and implement the following methods:
  - **Product(const char* name, int price)**
    Construct "Product" with name and price.
    Name characters only contain alphabet and space.
  - **Addition:**
    Implement the addition (define operator +) of two "Product" and return "Combo".

- Please design a class named "Combo" and implement the following methods:
  - **Combo()**
    Construct empty "Combo".
  - **Add(Product &product)**
    Add "Product" into "Combo", then unset price.
  - **SetPrice(int price)**
    Update "Combo" price.
  - **Addition:**
    Implements the addition (define operator +) of "Combo" and "Product", which is like "Combo.Add" but does not change the original "Combo".

Note: "Combo" does not break to multiple products. "Combo" in "Store" is not more than three. If there are the same "Product" that can be combined into the same "Combo", choose the first one.

Input:

No inputs.
**The main() function in your submission will be replaced when judging.
**You can use the main() function in "Sample Input" to test your program.

Output:

The result of executing your program with the given main function.

Sample Input / Output：

Sample Input	Sample Output

```
int main() {
    Product p1("Juice", 30), p2("Cookies", 20);
    Combo c1 = p1 + p2;
    Store store;
    store.Add(p1);
    store.Add(p2);
    store.Add(c1);
    c1.SetPrice(40);
    store.Add(c1);
    store.Buy("Cookies");
    store.Buy("Juice+Cookies");
    store.Buy("Juice");
    store.PrintReceipt();
    store.PrintReceipt();
    store.Buy("Cookies");
    store.Buy("Cookies");
    store.Buy("Juice+Cookies");
    store.Buy("Juice");
    store.PrintReceipt();
}
```

```
The combination has no set price.
      Receipt
====================
Cookies            20
Juice              30
Discount          -10
Cookies            20
Juice              30
Discount          -10
====================
Total              80
      Receipt
====================
====================
Total               0
      Receipt
====================
Cookies            20
Cookies            20
Juice              30
Discount          -10
Cookies            20
Juice              30
Discount          -10
====================
Total             100
```

☐ Eazy,Only basic programming syntax and structure are required.

☐ Medium,Multiple programming grammars and structures are required.

■ Hard,Need to use multiple program structures or more complex data types.

Expected solving time:

120 minutes

Other notes: