Homework 06

Re-submit Assignment

Due Mar 9 by 11:59pm   **Points** 10   **Submitting** a file upload

---

Develop a class called **Weight** that is an abstract data type (ADT) for an weight.

- The complete class will include all the following member functions:
  - A constructor to set pound and ounce; ex. 2lb 12oz can be set as **Weight item1(2,12)**;
    - If users set the ounce part grater than 15, like **Weight item1(2,20)**, it should be converted to 3lb 4oz.
  - A constructor to set ounce; ex. 36oz can be set as **Weight item2(36)**;
    - It should be converted to 2lb 4oz.
  - A default constructor (takes no input) that sets 0lb 0oz.
  - Public member function, "**get_Pounds()**" that returns the pound part of weight.
  - Public member function, "**get_Ounces()**" that returns the ounce part of weight.
    - The return value must be between 0 to 15. Note 16oz=1lb.
  - Public member function, "**get_Grams()**" that returns the weight in grams.
    - use 1oz=28.35g
  - A friend function, "**Weight add(const Weight& w1, const Weight& w1)**", which adds two **Weight** objects and returns a new object of **Weight** that stores the result.
  - A friend function, "**comp compare(const Weight& w1, const Weight& w1)**", which compares two **Weight** objects and returns **comp** type (enum type) variable of:
    - GREATER if w1>w2,
    - LESS if w1<w2, or
    - EQUAL if w1=w2.
    - **comp** type is **enum** type defined by
      "**enum comp{GREATER=1, LESS=-1,EQAUL=0};**"
      see **Link**   **(http://en.cppreference.com/w/cpp/language/enum)**
  - you may define private member functions. (helper functions)
  - Add "**const**" modifier for member function that does not modify member variables.
  - Member variables must be **private**.
  - It is **your decision** how to store the information, pounds and ounces.
- Add class definition and implementation into **the code below**.
  - What is 'switch'? See **Link**   **(http://en.cppreference.com/w/cpp/language/switch)**

```cpp
#include <iostream>

// enumerate "comp" type for "compare" function
enum comp{GREATER=1, LESS=-1,EQUAL=0};
const double ounce2grams = 28.35;
```

```cpp
/////////////////////////////////
//      ADD YOUR CODE HERE         //
/////////////////////////////////

// main function
int main(int argc, const char * argv[]) {
    Weight item1(2,12);
    Weight item2(36);

    std::cout << "Item1 is " << item1.get_Pounds()
    << "(lb) " << item1.get_Ounces() << "(oz).\n";

    std::cout << "Item2 is " << item2.get_Pounds()
    << "(lb) " << item2.get_Ounces() << "(oz).\n";

    std::cout << "Item1 is ";
    switch(compare(item1,item2))
    {
        case GREATER:
            std::cout << "greater than ";
            break;
        case LESS:
            std::cout << "less than ";
            break;
        case EQUAL:
            std::cout << "equal to ";
            break;
        default:
            std::cout << "Something Wrong ";
    }
    std::cout << "Item2\n";

    Weight total = add(item1,item2);
    std::cout << "The total weight is " << total.get_Pounds()
    << "(lb) " << total.get_Ounces() << "(oz), which is equivalent to"
    << total.get_Grams() << "(g)\n";

    return 0;
}
```

Output will be:

```
Item1 is 2(lb) 12(oz).
Item2 is 2(lb) 4(oz).
Item1 is greater than Item2
The total weight is 5(lb) 0(oz), which is equivalent to 2268(g)
```