

# COMP 132: Advanced Programming

Spring 2017

## Problem Session 9, April 27

In this PS, you are asked to implement overloaded operator functions and missing parts for a C++ class named **Weight**. We provide you C++ source/header file templates. Please put all your solutions in these files. The project includes declaration of the **Weight** class in file **Weight.h**. Note that we already provided the prototypes of some operator functions in **Weight.h**. You can use the main function (**test.cpp**) to test your implementation. You should implement the class **Weight**'s functions in file **Weight.cpp**.

The class **Weight** should include:

- a) private **integer data** to keep weight information: **kilogram** and **gram**.
- b) a **default constructor** that initializes weight to 1 kilogram and 0 gram.
- c) **overloaded addition (+) operator** function to enable the addition of two weights. It sums the kilogram and gram fields of two objects, and makes necessary conversions. Note that the value of a gram field may change outside the range (0-999) after + operation. Make sure that you correctly handle gram to kilogram conversion in your implementation. See the sample output.
- d) **overloaded output stream (<<) operator** for displaying a weight object in the format:  
    "Weight is <kilogram field> kg and <gram field> gr."
- e) **overloaded equality (=) operator** to allow comparisons of two weight objects. The function should compare kilogram and gram fields separately, and returns true if both are the same.
- f) **overloaded inequality (!=) operator** to allow comparisons of two weight objects. The function should compare kilogram and gram fields separately, and returns false if they are the same.
- g) **overloaded greater than (>) operator** to allow comparisons of two weight objects. The function should compare kilogram and gram fields separately, and returns true:
  - if current object's kilogram value is larger than the right object's kilogram value, or
  - if current object's gram value is larger than the right object's gram value when both of their kilogram values are the same.
- h) Add a **static data** named **count** to the **Weight** class. The count value should be initialized to 0, and it should keep track of the number of objects created out of **Weight** class. That is, every time a new object is created, value of count should be incremented within the class **Weight** appropriately (i.e. in the constructor). Every time an object is destroyed, value of count should be decremented within the class **Weight** (i.e. in the destructor).

You can test the above parts by using the code provided in test.cpp. See the sample output.

```
Testing class Weight
-----
y = Weight is 4 kg and 800 gr.
z = Weight is 3 kg and 300 gr.
w = Weight is 3 kg and 300 gr.
t = Weight is 3 kg and 350 gr.

Test addition operation:
x = y + z
x = Weight is 8 kg and 100 gr.
k = z + w
k = Weight is 6 kg and 600 gr.

Test boolean operations:
Weight is 3 kg and 300 gr. == Weight is 3 kg and 300 gr.
Weight is 8 kg and 100 gr. != Weight is 4 kg and 800 gr.
Weight is 4 kg and 800 gr. > Weight is 3 kg and 300 gr.
Weight is 3 kg and 350 gr. > Weight is 3 kg and 300 gr.

The count of Weight objects is: 6

Created: Weight is 10 kg and 100 gr.
Created: Weight is 5 kg and 400 gr.
The count of Weight objects after adding two objects is: 8

To be removed: Weight is 10 kg and 100 gr.
To be removed: Weight is 5 kg and 400 gr.
The count of Weight objects after removing two objects is: 6
```