İTÜ 11.04.2005
Computer Engineering Department

Object Oriented Programming 2nd Homework

In a program models of the following items are necessary:
A counter, a mod n counter and time of the day.
While designing the necessary classes, decide which relationship to use, **is-a** or **has-a**.

a) Design a class **Counter** that models *integer counter*s.
The class may have necessary **constructor**s and following abilities:
- The counter can be set to a given value.
- It has ++ (increment) and --(decrement) operators.
- It has an **operator >** to compare values of two counters.
- It has a function call operator that returns the value of the counter.

b) Design a class **ModN_Counter** that models *modulus n counter*s. For example a mod 3 counter counts as follows: 0, 1, 2, 0, 1…
The class will have a **constructor**, which can be called with one or two arguments. First argument is the modulus number n. The second argument is the initial value of the counter. If the second argument is not given the initial value is zero. If the second argument is greater than n-1 then the initial value will be n-1.
It has ++ (increment) and --(decrement) operators. Here, mod n arithmetic is used. If the value runs over (reaches the modulus number), the method will return true. If the value does not reach the modulus number the method returns false.
It will have an **operator >** to compare values of two counters.
It has a function call operator that returns the value of the counter.
The value of a mod_n_counter may not be changed (set) during the operation.
This class does not include other methods.

c) Design and implement a class **TimeOfDay** that models the *time of the day* (hour(0-23), minute, second).
To test the class TimeOfDay a main function is given below. This main function must run correctly on your TimeOfDay class.
Methods incS(), incM(), incH() are increment methods, which increment the time 1 second, 1 minute and 1 hour, respectively. Note that incrementing the time by one second can also affect the minute and the hour of the day.
Do not insert UNNECESSARY methods into class TimeOfDay.

```
int main()
{
    TimeOfDay time1(7, 45, 3);          // Hour: 7, minute: 45, second: 3
    TimeOfDay time2,time4;              // Hour: 0, minute: 0, second: 0
    TimeOfDay time3 = time1;
    time2.incS();                       // Increment the time by 1 second.
    time2.incM();                       // Increment the time by 1 minute.
```

```
    time2.incH();                          // Increment the time by 1 hour.
    time4 = time2;                         // Assignment
    if (time3 > time2)  cout << "time3 is greater (later) than time2" << endl;
       else  cout << "time3 is smaller (earlier) than time2" << endl;
    cout << "Time3=  Hour:" << time3.getH() << " Minute:" << time3.getM();
    return 0;
}
```

SUBMISSION:

- Read **C++ Programming Standards**, which can be found on the course web page. Write your program according to these rules. Don't forget to include sufficient explanation into your code.
- The name of the source file should be your number. For example: 040000601.cpp
- Copy the source file via anonymous ftp to akasya.cs.itu.edu.tr in to directory oop/ between **9.00** and **17.30** on **18th APRIL 2005**.
- List files and check the size of your file.
- You cannot view, delete, modify, overwrite or get files from this directory (Even your homework).
- Late submitted homeworks are not accepted.
- **Cheating** will not be tolerated. If cheating is discovered, all responsible students will be punished. Punishment for cheating is a score of **–50/100** points.
  It is allowed to discuss how to solve a problem with your classmates; however, **homeworks are not group homeworks. All actual programming should be an independent effort**.

I strongly recommend you to deal with this assignment before the 2nd midterm exam.