CSC-202-1: Computer Science II Session 1 (Summer II, 2018)
5:00pm – 7:00pm (MTWTh)
Instructor: Dr. Laxmi P. Gewali
Lab Practice 1 (Learning simple Class design)
June 4, 2018

(Lab Practice Submissions need to be done at the end of Lab Practice around 7:00 PM)

This is a very simple exercise to get you started in designing and using a class for constructing objects. The following is a class specification and declaration for *clockType*. First, key-in this program, then compile and run the executable code. Understand the working of different methods. Add the following three methods to the class and change the client code to generate the indicated output in the next page. You must use all three new methods in the client code. Note that minutes and seconds can not be greater than 59, and hour can not be greater than 23. If the increased value of seconds is greater than 59 then the method should reset it to zero and increase the minutes. Similar rule should be applied to increased minutes. If increased hour is greater than 23 then the value of hour should be set to zero and that is all. Your modified program must generate the exact output as shown below.

After completing the program submit it to your grader (Bibek Subedi).
E-mail address of Bibek is: bibek.subedi@unlv.edu

New methods to add:

void increaseSeconds(int s); // increases seconds by s
void increaseMinutes(int m); // increases minutes by m
void increaseHours(int h); // increases hours by h

```
#include <iostream>

using namespace std;

class clockType
{
private:
	int sec;
	int min;
	int hr;
public:
	void setTime(int, int, int);
	void getTime(int&, int&, int&) const;
	void printTime() const;
	void incrementSeconds();
	void incrementMinutes();
	void incrementHours();
	bool equalTime(const clockType& otherClock) const;
};

void clockType::setTime(int h, int m, int s)
{
	if (s >= 0 && s <= 59)
	{
		sec = s;
	}
	else
	{
		sec = 0;
	}

	if (m >= 0 && m <= 59)
	{
		min = m;
	}
	else
	{
		min = 0;
	}
```

```
        if (h >= 0 && m <= 59)
        {
                hr = h;
        }
        else
        {
                hr = 0;
        }
}

void clockType::getTime(int& h, int& m, int& s) const
{
        s = sec;
        m = min;
        h = hr;
}

void clockType::printTime() const
{
        if (hr < 10)
        {
                cout<<"0";
        }
        cout<<hr<<":";
        if (min < 10)
        {
                cout<<"0";
        }
        cout<<min<<":";
        if (sec < 10)
        {
                cout<<"0";
        }
        cout<<sec<<endl;
}
```

```
void clockType::incrementSeconds()
{
	sec = sec + 1;
	if (sec > 59)
	{
		sec = 0;
		incrementMinutes();
	}
}

void clockType::incrementMinutes()
{
	min = min + 1;
	if (min > 59)
	{
		min = 0;
		incrementHours();
	}
}

void clockType::incrementHours()
{
	hr = hr + 1;
	if (hr > 23)
	{
		hr = 0;
	}
}

bool clockType::equalTime(const clockType& otherClock) const {
	return (hr == otherClock.hr && min == otherClock.min && sec == otherClock.sec);
}
```

```
int main()
{
   clockType C1;
   clockType C2, C3;
   C1.setTime(9,20,6);
   C2.setTime(9,20,7);
   C3.setTime(10,30,40);

   cout << "First time: ";
   C1.printTime();
   cout << "Second time: ";
   C2.printTime();
   cout << "Third time: ";
   C3.printTime();
   cout << endl;

   C1.incrementSeconds();
   cout << "First time after incrementing seconds:"<<endl;
   cout << "First time: ";

   C1.printTime();
   cout << endl;
   cout << "Checking equality:"<<endl;
   if (C1.equalTime(C2)) cout << "First Time and Second Time are Equal " << endl;
   else cout << "First Time and Second Time are NOT Equal " << endl;

  // call the additional methods you wrote and make
  // the output looks like the output shown below

   return 0;

}
```

Required output:

```
First time: 9:20:06
Second time: 9:20:07
Third time: 10:30:40


First Time after incrementing seconds:
First time: 9:20:7


Checking Equality:
First Time and Second Time are Equal

Checking increaseHours Method:
Increasing hours in the First Time by 11 hours
Third Time: 20:20:07


Checking increaseMinutes Method:
Increasing minutes in the Second Time by 5 minutes
Third Time: 09:25:07


Checking increaseSeconds Method:
Increasing seconds in Third Time by 45 seconds
Third Time: 10:31:25
```