Working Constructor

With the `Time` class we might like to have another, overloaded constructor which takes hours and minutes. This is generally known as the **working constructor**. Here is the public interface of `Time` with both of these constructors.

```
class Time
{
public:
    Time();                    // default
    Time(int h, int m);        // working
    . . .
};
```

Unfortunately, if you have **any** explicit constructors, the synthesized one **is deleted**, so you have to add an **explicit default constructor** as done here. In C++11, however, you can just add the phrase `=default;` to the end of the prototype in the class header, and the compiler will **retain** the synthesized constructor that it normally writes.

The **implementation** of the constructors goes into the `.cpp` file along with the other member functions. The job of the constructor is to **initialize the data members**, so in the `Time` class, you might have code that looks something like this.

```
Time::Time() { m_hours = m_minutes = 0; }
Time::Time(int hours, int minutes)
{
    m_hours = h;
    m_minutes = m;
}
```