Review: C++ classes

private vs public:

- private members can only be accesses from within the class itself (i.e. via a member function)
- public members can be accessed from outside of the class

Accessing private data:

Constructor:

Multiple-file compilation:

time12.hpp

```
class Time12
{
private:
    int hour;
    int minute;
    string mer; // am or pm
public:
    //constructors:
    Time12( int h, int m, string me );
    Time12( ){};
    //display function
    void printTime();
    // mutator (setter) functions:
    void setHour( int h );
    void setMinute( int m );
    // accessor functions
    int getHour();
    int getMinute();
};
```

overloadSB

Review: C++ classes - definitions file

time12.cpp

```
#include <iostream>
#include <string>
using namespace std;
#include "Time12.hpp"

// note the overloaded constructor definition
Time12::Time12( ){
    hour = 12;
    minute = 59;
    mer = "AM";
}

Time12::Time12( int h, int m, string me ){
    if ( h >= 0 && h <= 12 )
        hour = h;
    else
        cout << "bad choice " << endl;
    if( m >=0 && m <60 )
        minute = m;
    else
        cout << "wrong choice for minute " << endl;
    mer = me;
}
void Time12::printTime(){
    cout << hour << ":" << minute << mer << endl;
}
void Time12::setHour( int h ){ // mutator
    if ( h >= 0 && h <= 12 )
        hour = h;
    else
        cout << "bad choice " << endl;
}
void Time12::setMinute( int m ){
    if( m >=0 && m <60 )
        minute = m;
    else
        cout << "wrong choice for minute " << endl;
}
// accessor methods
int Time12::getHour() {
    return hour;
}
...
```

Review: C++ classes - driver file

driver.cpp

```
// example class: 12-hour time clock
#include <iostream>
#include <string>
using namespace std;
#include "Time12.hpp"
int main(){
    Time12 t; // (10,30,"AM");
     Time12 t0(10,30,"AM");
     t.hour = 9;
    t.printTime();
    t.setHour( 10 );
    t.setMinute( 15 );
    t.printTime();
    Time12 t2(10,30,"AM"); // with constructor
    t2.printTime();

    cout << "hour " << t.getHour() << endl;
    cout << "minute " << t.getMinute() << endl;

    return 0;
}
```