5. (10 points) Overload the **ostream** operator for a movie class so that when a movie type is printed it looks similar to:

```
Trilogy: True
Duration: 136
Year: 1999
Budget: 63000000
Director: The Wachowskis
Title: The Matrix
```

You can assume the class has data members to match the output labels (trilogy,duration,year,etc.).

Solution:

```
class Movie {
private:
   // stuff
public:

    // Declare the operator<< as a friend. The signature must match the definition if you
    // do define the method outside of the class.
    friend std::ostream& operator<<(std::ostream& os, const Movie& m);
};

std::ostream& operator<<(std::ostream& os, const Movie& m) {
    // write everything to the os instance we passed in:
    os << "Trilogy:"<<m.trilogy<<std::endl;
    os << "Duration:"<<m.duration<<std::endl;
    os << "Year:"<<m.year<<std::endl;
    os << "Budget:"<<m.budget<<std::endl;
    os << "Director:"<<m.director<<std::endl;
    os << "Title:"<<m.title<<std::endl;
    // then return os to be printed to stdout.
    return os;
}
```