

```
#ifndef TIMER_H
#define TIMER_H

class Timer {
public:
    // constructors
    Timer(double s=0.0, double e=0.0, double t=0.0) : \
        ts(s), \
        te(e), \
        tt(t) { std::cout << "in constructor\n"; }
    Timer(const Timer& rhs) : \
        ts(rhs.ts), \
        te(rhs.te), \
        tt(rhs.tt) { std::cout << "in copy constructor\n"; }

    // desctructors (default ok)
    ~Timer()

    // assignment operator
    const Timer& operator=(const Timer& rhs)
    {
        std::cout << "in assignment operator\n";

        if(this != &rhs) { // standard alias test
            ts = rhs.ts;
            te = rhs.te;
            tt = rhs.tt;
        }
    }

    // friends
    friend ostream& operator<<(ostream& s, const Timer& rhs);
    friend ostream& operator<<(ostream& s, Timer *rhs)
        { return(s << (*rhs)); }

    // member functions
    void start();
    void end();
    double elapsed_us() { return tt; }
    double elapsed_ms() { return tt/1000.0; }
    double elapsed_s() { return tt/1000000.0; }
    double stamp_us();

private:
    double ts;
    double te;
    double tt;
};

#endif
```