```
//
//  SPRINGDAMPER.hpp
//  SPRINGDAMPER
//
//  Created by Ben Stager on 5/11/21.
//

#ifndef SPRINGDAMPER_hpp
#define SPRINGDAMPER_hpp

#include <stdio.h>
#include <fstream>
#include <math.h>
#include <boost/numeric/odeint.hpp>

const double gravity = 9.8;
class springDamper{

public:
    springDamper();
    springDamper(double y0, double v0, double mass, double k, double c);
    springDamper (const springDamper &p);
    double getPosition() const;
    double getVelocity() const;
    double getMass() const;
    double getSpring() const;
    double getDamper() const;
    void setPosition(const double newy0);
    void setVelocity(const double newv0);
    void setMass(const double newMass);
    void setSpring(const double newSpring);
    void setDamper(const double newDamper);

    void operator()(const boost::array<double , 2> &y, boost::array<double,2>
&yprime, const double runningTime);

    friend std::ostream &operator << (std::ostream &strm, const springDamper
&frnd);
    boost::array<double,2> &vector();

private:
       double mass;
       double k;
       double c;
       boost::array<double, 2> Y;
};


#endif /* SPRINGDAMPER_hpp */
```