

## Problem 1

Meek 215

2.72

Implement a Point class for 3-D points  $(x, y, z)$ . Include a default constructor, a `negate()` function to transform the point into its negative, a `norm()` function to return the point's distance from the origin  $(0,0,0)$ , and a `print()` function.

```
#include <iostream.h>
#include <math.h>

class Point {
public:
    Point(float x=0, float y=0, float z=0)
        { x_=x; y_=y; z_=z; }
    void negate()
        { x_ *= -1; y_ *= -1; z_ *= -1; }
    double norm()
        { return sqrt(x_*x_ + y_*y_ + z_*z_); }
    void print()
        { cout << "(" << x_ << ", " << y_ << ", "
          << z_ << ")"; }
private:
    float x_, y_, z_;
};

int main()
{
    Point p(1,2,3);
    cout << "\np = ";
    p.print();
    Point q = p;
    cout << "\nq = ";
    q.print();
    q.negate();
    cout << "\nq = ";
    q.print();
    cout << "\nq.norm() = " << q.norm() << endl;
    return 0;
}
```

## Problem 2

Implement a Matrix class for 2x2 matrices:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Include a default constructor, an `inverse()` function that returns the inverse of the matrix, a `det()` function that returns the determinant of the matrix, a Boolean function `isSingular()` that returns 1 or 0 according to whether the determinant is zero, and a `print()` function that writes out the matrix.