```
/*
 * 3D Vector Class in PseudoCode C++
 * ENGS 65 Homework1
 * Author: Brett Nicholas
 * 1/12/2015
 */
class Vector {
    private:
        double x, y, z;
    public:
        /* constructors */
        Vector(); /*init to 0*/
        Vector(double x, double y, double z);
        ~Vector();

        /* Get/setters  */
        double get_components() {return new double[3] = {this.x, this.y, this.z}; // returns array holding components

        /* class methods */
        float magnitude() { return sqrt(this.x^2 + this.y^2 + this.z^2) }; // returns magnitude of vector
        double dot(Vector v); // returns dot product scalar
        Vector cross(Vector v); // returns new cross product vector
        Vector proj(Vector v); // returns vector projection onto Vector v

        /* overloaded operators */
        Vector& operator + (Vector LHS, Vector RHS); // redefine addition for vectors
        Vector& operator - (Vector LHS, Vector RHS); // redefine subtraction for vectors
};

/* default constructor */
Vector::Vector() { /* default constructor*/
    this->components = {0,0,0};
};

/* Constructor to initialize components */
Vector::Vector(double x, double y, double z) {
    this.x = x;
    this.y = y;
    this.z = z;
};

/* calculates and returns dot product scalar */
float Vector::dot(Vector v) {
    return ( (this.x * v.x) + (this.y * v.y) + (this.z * v.z) );
};

/* calculates and returns cross product vector */
Vector Vector::cross(Vector v) {
    /*
     * compute cross product here
     */
    return new_vector;
};

Vector operator + (vector LHS, vector RHS) {
    /*
     * redefine addition to define vector addition
     */
}

Vector operator - (vector LHS, vector RHS) {
    /*
     * overload subtraction to define vector subtraction
     */
};

}; // end class Vector
```