```cpp
#include <iostream>
#include <cmath>
#include "V2D.h"

using namespace std;


////////////////////////////////////////////////////////////////////
//                      FRIEND FUNCTION                           //
////////////////////////////////////////////////////////////////////

std::ostream & operator<<(std::ostream & os, const v2d & V) {
    os << "[" << V.x << " , " << V.y << "]";
    return os;
}


////////////////////////////////////////////////////////////////////
//                         METHODS                                //
////////////////////////////////////////////////////////////////////

// Constructor

v2d::v2d(double a, double b) {
    x = a;
    y = b;
}


// Another Constructor

v2d::v2d(const v2d & v) {
    this->x = v.x;
    this->y = v.y;
}


// Destructor

v2d::~v2d() { }


// Equality operator - Returns a reference to an object of v2d

v2d & v2d::operator=(const v2d & v) {
    this->x = v.x;
    this->y = v.y;
    return *this;
}


// Addition operator - Returns a reference to an object of v2d

v2d & v2d::operator+(const v2d & v) {
    this->x = this->x + v.x;
    this->y = this->y + v.y;
    return *this;
}
```

```cpp

// Multiplication operator - Returns a double variable
// (vectorial multiplication)

double v2d::operator*(const v2d & v) {
    return (this->x * v.x + this->y * v.y);
}


// Multiplication operator - Returns a reference to an object of v2d.
// Overloaded to provide scalar multiplication.

v2d& v2d::operator*(double k) {
    x *= k;
    y *= k;
    return *this;
}


// Method that returns the module of the vector.

double v2d::length() {
    return (sqrt(pow(x , 2) + pow(y , 2)));
}
```