

Lab1

[source](#)

```
#include <iostream>
using namespace std;
#define GAP std::cout<< std::endl;
class Vector {
public:
    Vector(double);
    Vector(double, double);
    Vector(double, double, double);
    void display(double);
    void display(Vector);
    Vector operator+(Vector);
    size_t dimension_;
    double data_[3] = {0,0,0};
};

Vector::Vector(double element1) {
    cout << endl << "Creating a new Vector class object in R1 space:" <<
endl;
    dimension_ = 1;
    data_[0] = element1;
}

Vector::Vector(double element1, double element2) {
    cout << endl << "Creating a new Vector class object in R2 space:" <<
endl;
    dimension_ = 2;
    data_[0] = element1;
    data_[1] = element2;
}

Vector::Vector(double element1, double element2, double element3)
{
    cout << endl << "Creating a new Vector class object in R3 space:" <<
endl;
    dimension_ = 3;
    data_[0] = element1;
    data_[1] = element2;
    data_[2] = element3;
}

void Vector::display(double vec)
{
    std::cout<<vec;
}
```

```
void Vector::display(Vector vec)
{
    for(size_t i = 0; i < vec.dimension_; i++)
    {
        std::cout<< vec.data_[i] << " ";
    }
}

Vector Vector::operator+(Vector right_vec)
{
    if(this->dimension_ == 1 && right_vec.dimension_ == 1)
    {
        Vector output_vec(this->data_[0] + right_vec.data_[0]);
        return output_vec;
    }
    if(this->dimension_ == 2 && right_vec.dimension_ == 2)
    {
        Vector output_vec(this->data_[0] + right_vec.data_[0],this->data_[1] +
right_vec.data_[1]);
        return output_vec;
    }
    if(this->dimension_ == 3 && right_vec.dimension_ == 3)
    {
        Vector output_vec(this->data_[0] + right_vec.data_[0],this->data_[1] +
right_vec.data_[1],this->data_[2] + right_vec.data_[2]);
        return output_vec;
    }
    std::cout<<"given vectors aren't the same size";
    return 0;
}

void display(double vec)
{
    std::cout<<vec;
}

void display(Vector vec)
{
    for(size_t i = 0; i < vec.dimension_; i++)
    {
        std::cout<< vec.data_[i] << " ";
    }
}

int main(void) {
    double test_double = 3.14;
    Vector r1(1.0); // R1
    Vector r2(1.0, 2.0);
    Vector r3(1.0, 2.0, 3.0);
    std::cout << "Vector class object with number of dimensions:"
        << r1.dimension_ << endl;
    r1.display(r1);
    std::cout << "\nVector class object with number of dimensions:"
        << r2.dimension_ << endl;
```

```
        r2.display(r2);
std::cout<< "\nVector class object with number of dimensions:"
    << r3.dimension_ << endl;
    r3.display(r3);
GAP
r1.display(test_double);
GAP
display(r3);
GAP
display(test_double);
GAP
Vector add_vec1 = r1 + r1;
add_vec1.display(add_vec1);
Vector add_vec2 = r2 + r2;
add_vec2.display(add_vec2);
Vector add_vec3 = r3 + r3;
add_vec3.display(add_vec3);
return EXIT_SUCCESS;
}
```