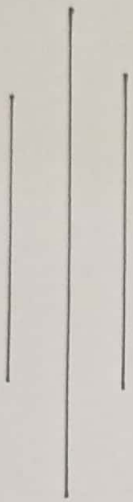
# KATHMANDU UNIVERSITY

## DHULIKHEL, KAVRE



A

Lab Report On

Object Oriented Programming {COMP116}

Lab Report No: 4


Submitted by:

Ashraya Kadel

CE Batch 2022

RollNo: 25

Submitted ~~to~~ to:

Rajani Chulyadyo

Department of Computer Science and Engineering


SUBMISSION DATE: 01 / 10 / 2023

<Q.1>: Define a class, Vector, which represents either a column vector or a row vector. A column (row) vector is a matrix consisting of a single column (row) of m elements.

Let m = 3.

Overload the unary minus operator to negative all the elements in the vector.

Ans

*) Source Code:

```
# include <iostream>
class Vector
{
    private:
        int x;
        int y;
        int z;
    public:
        Vector (double X, double Y, double Z) : x(X), y(Y), z(Z){}
        Vector () {};
        Vector operator - ()
        {
            Vector v1;
            v1.x = x;
            v2.y = y;
            v3.z = z;
            return v1;
        }
        Vector& operator = (const Vector & v1)
        {
            x = v1.x;
            y = v1.y;
            z = v1.z;
        }
```

```
    friend std::ostream& operator << (std::ostream& outputStream,
                                      const Vector& t);
    friend std::iostream& operator >> (std::istream& inputstream,
                                       const Vector& t);
};
std::ostream& operator << (std::ostream& outputStream,
                           const Vector& t)
{ return outputStream << t.x << " " << t.y << " " << t.z; }
std::istream& Operator >> (std::istream& inputstream, const
                           const Vector& t)
{return inputStream >> t.x >> t.y >> t.z; }

int main()
{
    Vector v1(2,-5,6), v2;
    std::cout << "The vector " << std::endl;
    std::cout << v1 << std::endl;
    std::cout << "The change by unary minus " << std::endl;
    v2 = -v1;
    std::cout << v2 << std::endl;
}
```

(*) Output

```
The vector
2 -5 6
The change by unary minus
-2 5 -6
```



Q.2: In the class, Vector, overload following operators.

i) operator+
ii) operator-
iii) operator*
iv) operator <<
v) operator >>
vi) operator +=
vii) operator ==
viii) operator >

Ans:

*) Source Code:

```
#include <iostream>

class Vector
{
    private:
        int x;
        int y;
        int z;
    public:
        Vector() {};
        Vector(double X, double Y, double Z): x(X), y(Y), z(Z) {}
        friend std::ostream& operator << (std::ostream& output
                                          outputStream, const Vector& t);
        friend std::istream& operator >> (std::iostream& inputstream,
                                          const Vector& t);
        Vector& operator = (const Vector& v1)
        { x = v1.x;
          y = v1.y;
          z = v1.z; }
        Vector operator+ (const Vector& v1)
        { int a,b,c
          a = x+v1.x;
          b = y+v2.y;
          c = z+v3.z; return Vector(a,b,c); }
```

```
Vector operator- (const Vector4 v2)
{   int a,b,c;
    a = x - v2.x;
    b = y - v2.y;
    c = z - v2.z;
    return Vector (a,b,c);
}
Vector operator* (const Vector4 v3)
{   int a,b,c;
    a = x * v3.x;
    b = y * v3.y;
    c = z * v3.z;
    return Vector (a,b,c); }
Vector operator += (const Vector4 v3)
{   int a,b,c;
    x = x + v3.x;
    y = y + v3.y;
    z = z + v3.z;
    return Vector (a,b,c); }
bool operator == (const Vector4 v ){
    if (x == v.x && y == v.y && z == v.z)
    { return true; }
    else
    { return false; } }
```



```
bool operator > (const Vector4 v )
{
   if (x > v.x && y > v.y && z > v.z )
   {
      return true;
   }
  else
   {
      return false;
   }
 }
};   // class definition end.

std::ostream& operator << (std::ostream& outputstream, const Vector4 t)
{
   return outputstream << t.x << " " << t.y << " " << t.z;
}

std::istream& operator >> (std::istream& outputstream,
                           const Vector4 t)
{
   return inputstream >> t.x >> t.y >> t.z;
}

int main()
{
   Vector v1, v2, v3, v4, v5, v6, v7(1,1,1), v8, v9;
   std::cout << "Please enter your vector 1" << std::endl;
   std::cin >> v1;
```

```
std:: cout <<"Please enter vector 2"<<std:: endl;
std:: cin >> v2;
std:: cout << "your two vectors are" << std:: endl;
std:: cout << v1<<std:: endl;
std:: cout<< v2 <<std:: endl;
v3 = v1+v2;
std:: cout << "The sum of two vectors "<<std:: endl;
std:: cout << v3 <<std:: endl;
v4 = v1-v2;
std:: cout << "The difference of two vectors"<<std:: endl;
std:: cout << v4 << std:: endl;
std:: ~~cin~~ cout<< "Enter column vector "<<std:: endl;
std:: cin >> v5;
v6 = v1 * v5;
std:: cout<< "Product of v1 & cv " <<std:: endl;
std:: cout << " v6 <<std:: endl;
std:: cout<< "The += of vector 1 "<<std:: endl;
v1 += v7;
std:: cout << "v1 <<std:: endl;
std:: cout << "Please enter vector 8 "<<std:: endl;
std:: cin >> v8;
std:: cout << "Please enter vector 9 "<< std:: endl;
std:: cin >> v9;
std:: cout <<"your two vectors are "<<std:: endl;
std:: cout << v8 << ~~v9 <<<~~ std:: endl;
std:: cout << v9 << std:: endl;
if (v8 == v9 ) { std:: cout<< "Vectors equal"<<std:: endl";}
else { std:: cout << "Vectors not equal " <<std:: endl; }
```



```
if (v8>v9) { std:: cout << " v8 > v9 " << std:: endl; }
else { std:: cout << "v9 > v8" std:: endl; }
}
```

*) Output:

```
Please enter your vector 1
2
2
2
Please enter your vector 2
1
1
1
Your two vectors are
 2  2  2
 1  1  1
The sum of two vectors is
3  3  3
The difference of two vector is
 1 1  1
Enter column vector
 3
 3
 3
The product of v1 and column vector is
 6    6    6
The += of vector 1
  3   3    3
Please enter vector 8
4
5
6
```

```
Please enter vector 9
2
3
4
Your two vectors are
4  5  6
2  3  4
Vectors are not equal (v8 and v9)
v8 > v9
```