**Practical 7**

**Name: Saurabh Asnare (SYCOA 276)**

```
#include <iostream>
using namespace std;

// # Class Template
template <class T>
class Vector
{
private:
    // 'vec' is an array
    T *vec;
    int size;

public:
    // # Constructor
    Vector(int m)
    {
        size = m;
        vec = new T[size];
        for (int i = 0; i < size; i++)
            vec[i] = 0;
    }
    // # Create Vector
    void create()
    {
        cout << "\n# Create Vector: " << endl;
        for (int i = 0; i < size; i++)
        {
            cout << "  ";
            cout << "vec[" << i << "] = ";
            cin >> vec[i];
        }
    }
    // # Modify Vector
    void modify()
    {
        int pos;
        cout << "\n# Modify Vector: " << endl;
        cout << "  Previous Vector: ";
        display(1);
    up:
        cout << "Enter position (0-" << size - 1 << ") to make changes: ";
        cin >> pos;
        if (pos >= size)
        {
            cout << "Please enter correct position..!" << endl;
            goto up;
        }
        cout << "Enter new vector value: ";
        cin >> vec[pos];
        cout << "  New Vector: ";
    }
    // # Multiply By Scalar
    void multiply()
    {
        T sc;
        cout << "\n# Multiply By Scalar: " << endl;
        cout << "  Previous Vector: ";
        display(1);
        cout << "Enter scaler number to multiply with vector: ";
        cin >> sc;
        for (int i = 0; i < size; i++)
            vec[i] = vec[i] * sc;
        cout << "  New Vector: ";
    }
    // # Display Vector
    void display(int n)
    {
        int i;
        if (n == 0)
        {
            cout << "\n# Display Vector: " << endl;
            cout << "  Vector: ";
        }
        cout << "(";
        for (i = 0; i < size - 1; i++)
        {
            cout << vec[i] << ",";
        }
        cout << vec[i] << ")\n";
    }
};

// # Main Function
int main()
{
    int size;
    cout << "\n# Generic Vector #\n";
    cout << "\n# Enter size of the Vector: ";
    cin >> size;
    // Creating an Integer Vector
    Vector<int> vec(size); //.... 'vec' is an object of class template 'Vector'
    vec.display(0);
    vec.create();
    vec.display(0);
    vec.modify();
    vec.display(1);
    vec.multiply();
    vec.display(1);
    cout << "\n";
    return 0;
```

## output:

```
# Generic Vector #


# Enter size of the Vector: 3


# Display Vector:
  Vector: (0,0,0)


# Create Vector:
  vec[0] = 10
  vec[1] = 20
  vec[2] = 30


# Display Vector:
  Vector: (10,20,30)


# Modify Vector:
  Previous Vector: (10,20,30)
Enter position (0-2) to make changes: 1
Enter new vector value: 15
  New Vector: (10,15,30)


# Multiply By Scalar:
  Previous Vector: (10,15,30)
Enter scaler number to multiply with vector: 2
  New Vector: (20,30,60)
```