Assignment no 1

Name:Muhammad Shaheer

Roll no:0480

Code:

#include<iostream>

using namespace std;

class MyClass

{

    public:

    int \*p;

    int \*q;

    int \*r;

    int arr[6]={1,2,3,4,5,6};

    int size = sizeof(arr)/sizeof(int);

    int num;

    int val;

    int pos;

    int index;

    MyClass()

    {

        this->p= arr;

        q = p;

        r = nullptr;

    }

    void get();

    void update();

    void length();

    void back();

    void Next();

    void start();

    void end();

    void Remove();

    void Add();

};

void MyClass::start()

{

    if (p == q)

    {

        cout<<"You are already at the start of array."<<endl;

    }

    else

    {

        p =arr;

        cout<<"you are at the start of the array and its value is: "<<\*p<<endl;

    }

}

void MyClass :: end()

{

    for (int i=0; i < size ; i++)

    {

        p++;

    }

    cout<<"The end value of your array is:"<<\*p<<endl;

    if (p == r)

    {

        cout<<"You are already at the end of the array."<<endl;

    }

    else

    {

        for (int j = 1; j < size; j++)

        {

            if (index == size -1)

            {

                cout<<"You are already at the end of the array."<<endl;

                break;

            }

            else

            {

                p+=1;

                index++;

            }

        }

    }

}

void MyClass::Next()

{

    // cout<<"Your current value of pointer is: "<<\*p<<endl;

    // p++;

    // cout<<"your next value is :"<<\*p<<endl;

    if((\*p = size))

    {

        cout<<"You are already at the end of the array can't go further more."<<endl;

    }

    else

    {

        cout<<"Your first position was: "<<\*p<<endl;

        p +=1;

        p++;

        cout<<"And your next position is: "<<p<<endl;

    }

}

void MyClass :: back()

{

    // cout<<"Your current value of pointer is: "<<\*p<<endl;

    // p--;

    // cout<<"your previous value is :"<<\*p<<endl;

    if (p == q)

    {

        cout<<"You are already at the start of the array can't go back anymore."<<endl;

    }

    else

    {

        cout<<"Your first position was: "<<\*p<<endl;

        // p -=1;

        index--;

        cout<<"And your previous position is: "<<\*p<<endl;

    }

}

void MyClass :: get()

{

    cout<<"your current pointer position is :"<<\*p<<endl;

}

void MyClass :: length()

{

    int size = sizeof(arr)/sizeof(int);

    cout<<"The length of your array is :"<<size<<endl;

}

void MyClass ::update()

{

    int num;

    p=arr;

    cout<<"your origional value is :"<<\*p<<endl;

    cout<<"Enter any number to update:"<<num<<endl;

    cin>>num;

    \*p=num;

    cout<<"your updated value is:"<<\*p<<endl;

    for (int i = 0; i < size-1; i++)

    {

       cout<<\*p<<" ";

    }

    cout<<endl;

}

// bool MyClass :: isfull()

// {

//     if(\*p==6)

//     return true;

//     else

//     return false;

// }

void MyClass ::Add()

{

    if(sizeof(arr)==size)

    {

        cout<<"no space for new entry"<<endl;

    }

    else

    {

        p++;

        arr[6]=val;

        //Add();

    };

}

void MyClass ::Remove()

{

    cout<<"size of array is:"<<size<<endl;

    if(sizeof(arr)==size)

    {

        size--;

    }

    else

    {

        p--;

        arr[6]=val;

        //remove()

    }

}

int main()

{

    int x;

    MyClass obj;

    do

    {

        cout<<"1. Start"<<endl;

        cout<<"2. End"<<endl;

        cout<<"3. Next"<<endl;

        cout<<"4. Back"<<endl;

        cout<<"5. Get"<<endl;

        cout<<"6. Add"<<endl;

        cout<<"7. Lenght"<<endl;

        cout<<"8. update"<<endl;

        cout<<"9. remove"<<endl;

        cout<<"0. Exit"<<endl;

        cout<<"Enter your choice:"<<endl;

        cin>>x;

        switch (x)

        {

        case 1:

            obj.start();

            break;

        case 2:

            obj.end();

            break;

        case 3:

            obj.Next();

            break;

        case 4:

            obj.back();

            break;

        case 5:

            obj.get();

            break;

        case 6:

            obj.Add();

            break;

        case 7:

            obj.length();

            break;

        case 8:

            obj.update();

            break;

        case 9:

            obj.Remove();

            break;

        default:

            break;

        }

    } while (x!=0);

    return 0;

};

Output:

