

**Object Oriented Programming**

**Assignment 4**

Name : Ganta Sowmya Kranthi

Roll no : 201210019

Year : 2nd year

Semester :4th Sem

Group : 1

**Q 1 .** Write a c++ program to implement STL Data Structure Vector using some class and methods.

Functionality of class :

It should contains following methods

1 . One constructor for initialising the vector

2. Resize method for resizing the array

3. push\_back method to insert element into array(vector).

4. Get method to get the element at desired index

5. Pop method to remove the element from vector

6. Print method to print the elements in the array along with size and capacity.

Consider array inputs as { 2, 4, 5, 8, 0, 23, 14, 45}

Perform some operations like get , print, push\_back, pop etc.

**Code:**

#include <bits/stdc++.h>

using namespace std;

template <typename T> class vect

{

    T\* arr;

    int capacity;

    int current;

public:

    // Default constructor to initialise

    // an initial capacity of 1 element and

    // allocating storage using dynamic allocation

    vect()

    {

        arr = new T[1];

        capacity = 1;

        current = 0;

    }

    // Function to add an element at the last

    void push(T data)

    {

        // if the number of elements is equal to the

        // capacity, that means we don't have space to

        // accommodate more elements. We need to double the

        // capacity

        if (current == capacity)

        {

        resize();

        }

        // Inserting data

        arr[current] = data;

        current++;

    }

       void resize( )

       {

            T\* temp = new T[2 \* capacity];

            // copying old array elements to new array

            for (int i = 0; i < capacity; i++) {

                temp[i] = arr[i];

            }

            // deleting previous array

            delete[] arr;

            capacity \*= 2;

            arr = temp;

       }

    // function to add element at any index

    void push(T data, int index)

    {

        // if index is equal to capacity then this

        // function is same as push defined above

        if (index == capacity)

            push(data);

        else

            arr[index] = data;

    }

    // function to extract element at any index

    T    get(int index)

    {

        // if index is within the range

        if (index < current)

            return arr[index];

    }

    // function to delete last element

    void pop() { current--; }

    // function to get size of the vector

    int size() { return current; }

    // function to get capacity of the vector

    int getcapacity() { return capacity; }

    // function to print array elements

    void print()

    {

        for (int i = 0; i < current; i++) {

            cout << arr[i] << " ";

        }

        cout << endl;

    }

};

// testing vector class

int main()

{

    vect<int> v;

    int choice,value,ind;

    do {

        cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

    cout<<"1.INSERT ELEMET  "<<endl;

    cout<<"2.SEARCH ELEMENT "<<endl;

    cout<<"3.POP ELEMENT  "<<endl;

    cout<<"4.UPDATE ELEMENT   "<<endl;

    cout<<"5.PRINT ARRAY "<<endl;

    cout<<"6.SIZE OF ARRAY"<<endl;

    cout<<"7.CAPACITY OF ARRAY"<<endl;

    cout<<"8.EXIT"<<endl;

    cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

    cout<<"ENTER THE CHOICE "<<endl;

    cin>>choice;

    switch(choice)

    {

        case 1:

           cout<<"enter the data to be inserted"<<endl;

           cin>>value;

           v.push(value);

           break;

        case 2:

         cout<<"enter the index to get the element"<<endl;

           cin>>ind;

          cout<<"element at index "<<ind<<" is "<< v.get(ind)<<endl;

           break;

        case 3:

               v.pop();

               break;

        case 4:

               cout<<"enter the index to be updated"<<endl;

               cin>>ind;

               cout<<"enter the index to be updated"<<endl;

               cin>>value;

               v.push(value,ind);

               break;

        case 5:

                 cout<<"array elements are"<<endl;

                v.print();

                 break;

        case 6:

              cout<<"SIZE IS "<<v.size()<<endl;

               break;

        case 7:

             cout<<"CAPACITY IS " <<v.getcapacity()<<endl;

             break;

        case 8:

             exit(1);

             break;

        default:

          cout<<"INVALID OPTION"<<endl;

    }

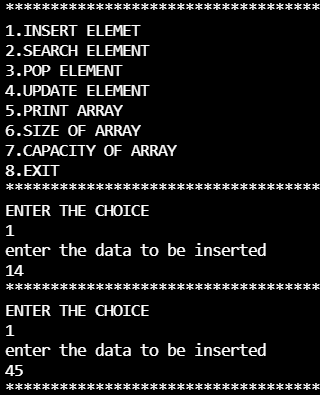
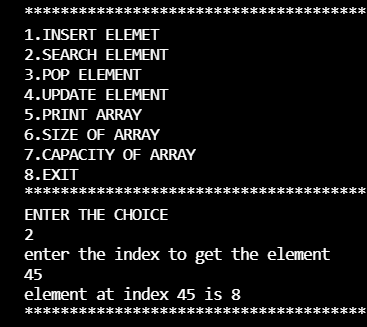
    }while(choice!=8);

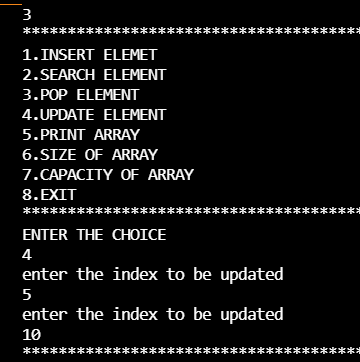
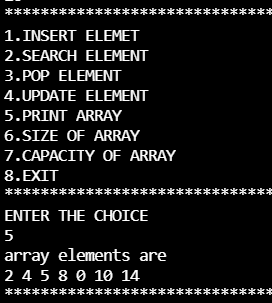
    return 0;

}

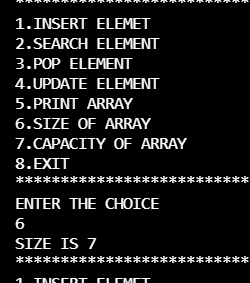
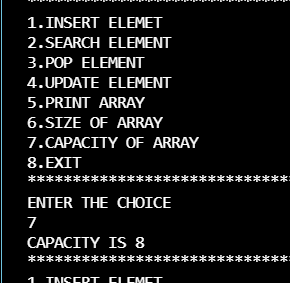
**Output:**

**Insertion: Searching:**

** **

**Updating: Printing:  
 **

**Size of Array: Capacity of array:**

** **