Assignment – 38

A Job Ready Bootcamp in C++, DSA and IOT

list

1. List functions in C++ STL (Standard Template Library)
2. #include<iostream>
3. #include<list>
4. using namespace std;
5. void displayList(list<int> L)
6. {
7. list<int>::iterator L\_iter;
8. for (L\_iter=L.begin(); L\_iter!=L.end(); L\_iter++)
9. {
10. cout<<\*L\_iter<<" ";
11. }
12. cout<<endl;
14. }
15. int main()
16. {
17. list<int> Ilist;
18. Ilist.push\_front(12);
19. Ilist.push\_front(24);
20. Ilist.push\_front(36);
21. Ilist.push\_front(48);
22. Ilist.push\_back(13);
23. Ilist.push\_back(26);
24. Ilist.push\_back(39);
25. Ilist.push\_back(52);
26. Ilist.push\_back(65);
27. cout<<"List after PushFront and Pushback :"<<endl;
28. displayList(Ilist);
29. list<int> ::iterator it;
30. it=Ilist.begin();
31. it++;
32. Ilist.insert(it,200);
33. Ilist.insert(it,300);
34. Ilist.insert(it,400);
35. cout<<"List after Inser 200, 300 and 400"<<endl;
36. displayList(Ilist);
37. cout<<"the size of the List is :"<<Ilist.size()<<endl;
38. Ilist.sort();
39. cout<<"List after sorted Elements"<<endl;
40. displayList(Ilist);
41. Ilist.reverse();
42. cout<<"List after Reverse order"<<endl;
43. displayList(Ilist);
44. Ilist.remove(300);
45. cout<<"List after Remove 300 "<<endl;
46. displayList(Ilist);
47. cout<<"First Element of the List is:"<<Ilist.front()<<endl;
48. cout<<"Last Element of the list is :"<<Ilist.back()<<endl;
49. Ilist.pop\_front();
50. Ilist.pop\_back();
51. cout<<"List after PopFront nad Popback"<<endl;
52. displayList(Ilist);
54. cout<<"Again print the whole List:"<<endl;
55. list<int> ::iterator L\_itor;
56. for(L\_itor=Ilist.begin(); L\_itor!=Ilist.end(); L\_itor++)
57. {
58. cout<<\*L\_itor<<" ";
59. }
60. cout<<endl;
61. cout<<"Reverse the list :"<<endl;
62. list<int>:: reverse\_iterator  L\_rit;
63. for(L\_rit=Ilist.rbegin(); L\_rit!=Ilist.rend();L\_rit++ )
64. {
65. cout<<\*L\_rit<<" ";
66. }
67. cout<<endl;
68. list<int> Ilist2;
69. Ilist2.assign(5,100);
70. cout<<"Second List"<<endl;
71. displayList(Ilist2);
72. Ilist2.merge(Ilist);
73. cout<<"After Merged Both the List"<<endl;
74. displayList(Ilist2);
75. Ilist2.unique();
76. cout<<"Unique Elements In List 1"<<endl;
77. displayList(Ilist2);
78. it=Ilist2.begin();
79. cout<<"Erase begin element"<<endl;
80. Ilist2.erase(it);
81. displayList(Ilist2);
82. return 0;
83. }

2. Assign the elements to the list (different methods) - Example of list::assign() | C++ STL

//Assign the elements to the list (different methods) - Example of list::assign

#include<iostream>

#include<list>

using namespace std;

void diplatLIst(list<int> L)

{

    list<int> :: iterator itr;

    for(itr=L.begin(); itr!=L.end(); itr++)

    {

        cout<<\*itr<<" ";

    }

    cout<<endl;

}

int main()

{

    list<int> List1;

    list<int> List2;

    list <int> List3;

    list<int> List4;

    cout<<"First way of assign List Elements"<<endl;

    List1.assign(5,120);

    diplatLIst(List1);

    cout<<"Second way of assign List Elements using Begin() and End()"<<endl;

    List2.assign(List1.begin(), List1.end());

    diplatLIst(List2);

    cout<<"Third way of assign List Elements using NOrmal array"<<endl;

    int arr[]={12,23,34,434,45};

    List3.assign(arr, arr+5);

    diplatLIst(List3);

    return 0;

}

3. Iterate a list C++ STL

#include<iostream>

#include<list>

using namespace std;

int main()

{

    list<int> List1;

    List1.push\_back(23);

    List1.push\_back(24);

    List1.push\_back(25);

    List1.push\_back(28);

    List1.push\_back(27);

    List1.push\_back(29);

    list<int> :: iterator it;

    for(it=List1.begin(); it!=List1.end(); it++)

    {

        cout<<\*it<<" ";

    }

return 0;

}

4. Iterate a list in reverse order C++ STL

#include<iostream>

#include<list>

using namespace std;

int main()

{

    list<int> List1;

    List1.push\_back(23);

    List1.push\_back(24);

    List1.push\_back(25);

    List1.push\_back(28);

    List1.push\_back(27);

    List1.push\_back(29);

    list<int> :: reverse\_iterator it;

    for(it=List1.rbegin(); it!=List1.rend(); it++)

    {

        cout<<\*it<<" ";

    }

}

5. Input and add elements to a list C++ STL

//Input and add elements to a list C++ STL 6. Get the first and last element of the list C++ STL

#include<iostream>

#include<list>

#include<iterator>

using namespace std;

    void displayList(list<int> L)

    {

        list<int> :: iterator itr;

        for(itr=L.begin(); itr!= L.end(); itr++)

        {

            cout<<\*itr<<" ";

        }

        cout<<endl;

    }

int main()

{

    int n;

    list<int> list1;

    cout<<"Enter Elements"<<endl;

    while (true)

    {

        cin>>n;

        list1.push\_back(n);

        if (n==0)

        {

            break;

        }

    }

    cout<<"List after insertion"<<endl;

    displayList(list1);

    return 0;

}

6. Get the first and last element of the list C++ STL

//Input and add elements to a list C++ STL 6. Get the first and last element of the list C++ STL

#include<iostream>

#include<list>

#include<iterator>

using namespace std;

    void displayList(list<int> L)

    {

        list<int> :: iterator itr;

        for(itr=L.begin(); itr!= L.end(); itr++)

        {

            cout<<\*itr<<" ";

        }

        cout<<endl;

    }

int main()

{

    int n;

    list<int> list1;

    cout<<"Enter Elements"<<endl;

    while (true)

    {

        cin>>n;

        if (n==0)

        {

            break;

        }

        list1.push\_back(n);

    }

    cout<<"List after insertion"<<endl;

    displayList(list1);

    cout<<"First Elements of the list is:"<<list1.front()<<endl;

    cout<<"Last Elements of the list is:";

    cout<<list1.back();

    return 0;

}

7. Insert the element at beginning and end of the list | C++ STL

//Insert the element at beginning and end of the list

#include<iostream>

#include<list>

using namespace std;

int main()

{

    list<int> List1;

    int n;

    list<int> list1;

    cout<<"Enter Elements"<<endl;

    while (true)

    {

        cin>>n;

        if (n==0)

        {

            break;

        }

        list1.push\_back(n);

    }

    int a,b;

    cout<<"Insert at begning"<<endl;

    cin>>a;

    list1.push\_front(a);

    cout<<"Insert at end"<<endl;

    cin>>b;

    list1.push\_back(b);

    list<int> ::iterator it;

    for(it=list1.begin(); it!=list1.end(); it++)

    {

        cout<<\*it<<" ";

    }

    return 0;

}

8. Remove all occurrences of an element and remove set of some specific from the list C++ STL

//Remove all occurrences of an element and remove set of some specific from the list

#include<iostream>

#include<list>

using namespace std;

int main()

{

    list<int> List1;

    int n;

    list<int> list1;

    cout<<"Enter Elements"<<endl;

    while (true)

    {

        cin>>n;

        if (n==0)

        {

            break;

        }

        list1.push\_back(n);

    }

    list<int> ::iterator it;

    for(it=list1.begin(); it!=list1.end(); it++)

    {

        cout<<\*it<<" ";

    }

    cout<<endl;

    int a;

    cout<<"Enter a number"<<endl;

    cin>>a;

    list1.remove(a);

     list<int> ::iterator itr;

    for(itr=list1.begin(); itr!=list1.end(); itr++)

    {

        cout<<\*itr<<" ";

    }

    return 0;

}

9. Remove all consecutive duplicate elements from the list | C++ STL

//Remove all occurrences of an element and remove set of some specific from the list

#include<iostream>

#include<list>

using namespace std;

int main()

{

    list<int> List1;

    int n;

    list<int> list1;

    cout<<"Enter Elements"<<endl;

    while (true)

    {

        cin>>n;

        if (n==0)

        {

            break;

        }

        list1.push\_back(n);

    }

    list<int> ::iterator it;

    for(it=list1.begin(); it!=list1.end(); it++)

    {

        cout<<\*it<<" ";

    }

    cout<<endl;

    list1.unique();

     list<int> ::iterator itr;

    for(itr=list1.begin(); itr!=list1.end(); itr++)

    {

        cout<<\*itr<<" ";

    }

    return 0;

}

10. Merge two lists C++ STL

#include<iostream>

#include<list>

using namespace std;

int main()

{

    list<int> List1 = {1,3,4,5,6};

    list<int> list2 ={23,34,45,56,55};

    List1.merge(list2);

     list<int> ::iterator it;

    for(it=List1.begin(); it!=List1.end(); it++)

    {

        cout<<\*it<<" ";

    }

    return 0;

}

11. Creating a list by assigning the all elements of another list C++ STL

#include<iostream>

#include<list>

using namespace std;

int main()

{

    list<int> List1 = {1,3,4,5,6};

    list<int> list2;

    list2.assign(List1.begin(), List1.end());

     list<int> ::iterator it;

    for(it=list2.begin(); it!=list2.end(); it++)

    {

        cout<<\*it<<" ";

    }

    return 0;

}

12. Assign a list with array elements C++ STL

#include<iostream>

#include<list>

using namespace std;

int main()

{

    //list<int> List1;

    list<int> list2;

    int arr[]={1,3,4,5,6};

    list2.assign(arr, arr+5);

     list<int> ::iterator it;

    for(it=list2.begin(); it!=list2.end(); it++)

    {

        cout<<\*it<<" ";

    }

    return 0;

}

13. Push characters in a list and print them separated by space in C++ STL

//13. Push characters in a list and print them separated by space in C++ STL

#include<iostream>

#include<list>

using namespace std;

int main()

{

    list<char> clist;

    for (char i = 'A'; i <= 'Z'; ++i)

    {

        clist.push\_back(i);

    }

   for(char x: clist)

   {

    cout<<x<<" ";

   }

   cout<<endl;

    return 0;

}

14. Access elements of a characters list using const\_iterator in C++ STL

//13. Push characters in a list and print them separated by space in C++ STL

#include<iostream>

#include<list>

using namespace std;

int main()

{

    list<char> clist;

    for (char i = 'A'; i <= 'Z'; ++i)

    {

        clist.push\_back(i);

    }

    list<char> :: const\_iterator it;

    for(it=clist.begin(); it!= clist.end(); ++it)

    {

        cout<<\*it<<' ';

    }

   cout<<endl;

    return 0;

}