

Ganpat University

Faculty of Engineering & Technology

Computer Science & Engineering

Name:- Dwij Vatsal Desai

Sem:- 2

Sub: - ESFP-II

Enrollment No.:- 23162121027

Prac:- 16

Practical 16

Definition:

Complete the code for the object assigned to you to satisfy following specifications.

1. You must perform this practical on given object.
2. Create four separate modules in your given object practical and this should be access by the use of switch case condition.
3. In first module you have to implement the concept of vector where you have to use all respective function that is push_back(), pop_back(), size(), insert(), begin(), end(), and swap(), etc.
4. In second module you have to implement the concept of deque where you have to use all respective function that is push_back(), push_front(), pop_back(), pop_front(), front(), back(), and size() function.
5. In third module you have to implement the concept of list where you have to use all respective function that is push_back(), push_front(), pop_back(), pop_front(), front(), back(), and size(), reverse(), unique(), sort(), and remove() function.
6. In fourth and fifth module you have to implement the concept of set and multiset, where you have to use all respective function that is insert(),size(), remove(), clear(),lower_bound(), upper_bound(), etc.
7. In 6th, 7th, 8th, and 9th module, you have to implement the concept of Map,

Multimap, Stack and Queue related minimum 6 pre-defined function.

8. Experiments with minimum 5 data/record from the user and display according **to the choice of user category wise. (Minimum eight different options should be there for displaying purpose of information, and if you want more, as per program requirement you can add more choices).**

.Code:-

```
#include <iostream>
#include <vector>
#include <deque>
#include <list>
#include <set>
#include <map>
#include <stack>
#include <queue>
#include <algorithm>

using namespace std;

void vectorModule()
{
    vector<int> vec;
    vec.push_back(10); vec.push_back(20); vec.push_back(30);

    cout << "Vector size: " << vec.size() << endl;
    cout << "Vector elements: "; for (auto i : vec) cout << i << " ";
    cout << endl;

    vec.pop_back();
    cout << "After pop_back(), Vector size: " << vec.size() << endl;
}

void dequeModule()
{
    deque<int> dq;
    dq.push_back(10); dq.push_front(5); dq.push_back(15);

    cout << "Deque size: " << dq.size() << endl;
    cout << "Deque elements: "; for (auto i : dq) cout << i << " ";
    cout << endl;

    dq.pop_back();
    cout << "After pop_back(), Deque size: " << dq.size() << endl;
}

void listModule()
{
    list<int> li;
    li.push_back(10); li.push_front(5); li.push_back(15);

    cout << "List size: " << li.size() << endl;
    cout << "List elements: "; for (auto i : li) cout << i << " ";
    cout << endl;

    li.pop_back();
    cout << "After pop_back(), List size: " << li.size() << endl;
}

void setModule()
{
    set<int> s;
    s.insert(10); s.insert(20); s.insert(10); // Duplicate, won't be added

    cout << "Set size: " << s.size() << endl;
    cout << "Set elements: "; for (auto i : s) cout << i << " ";
    cout << endl;

    s.erase(20);
    cout << "After erase(), Set size: " << s.size() << endl;
}
```

```

void mapModule()
{
    map<string, int> mp;
    mp["apple"] = 10; mp["banana"] = 20;

    cout << "Map size: " << mp.size() << endl;
    cout << "Map elements: "; for (auto i : mp) cout << i.first << ":" << i.second
<< " ";
    cout << endl;

    mp.erase("banana");
    cout << "After erase(), Map size: " << mp.size() << endl;
}

void stackModule()
{
    stack<int> s;
    s.push(10); s.push(20); s.push(30);

    cout << "Stack size: " << s.size() << endl;
    cout << "Stack top: " << s.top() << endl;
}

void queueModule()
{
    queue<int> q;
    q.push(10); q.push(20); q.push(30);

    cout << "Queue size: " << q.size() << endl;
    cout << "Queue front: " << q.front() << endl;
    cout << "Queue back: " << q.back() << endl;
}

int main()
{
    int choice;
    do {
        cout << "\n1. Vector Module\n2. Deque Module\n3. List Module\n4. Set
Module\n5. Map Module\n6. Stack Module\n7. Queue Module\n8. Exit\nEnter your
choice: ";
        cin >> choice;

        switch (choice)
        {
            case 1: vectorModule(); break;
            case 2: dequeModule(); break;
            case 3: listModule(); break;
            case 4: setModule(); break;
            case 5: mapModule(); break;
            case 6: stackModule(); break;
            case 7: queueModule(); break;
            case 8: cout << "Exiting..."; break;
            default: cout << "Invalid choice!" << endl;
        }
    } while (choice != 8);

    return 0;
}

```

Output:-

```
"C:\Users\dwijd\OneDrive\Documents\collage practicals\ESFP-II\Practical_16.exe"
```

1. Vector Module
2. Deque Module
3. List Module
4. Set Module
5. Map Module
6. Stack Module
7. Queue Module
8. Exit

Enter your choice:1

Vector size: 3

Vector elements: 10 20 30

After pop_back(), Vector size: 2

1. Vector Module
2. Deque Module
3. List Module
4. Set Module
5. Map Module
6. Stack Module
7. Queue Module
8. Exit

Enter your choice:2

Deque size: 3

Deque elements: 5 10 15

After pop_back(), Deque size: 2

1. Vector Module
2. Deque Module
3. List Module
4. Set Module
5. Map Module
6. Stack Module
7. Queue Module
8. Exit

Enter your choice:3

List size: 3

List elements: 5 10 15

After pop_back(), List size: 2

1. Vector Module
2. Deque Module
3. List Module
4. Set Module
5. Map Module
6. Stack Module
7. Queue Module
8. Exit

Enter your choice:4

```
Set size: 2
Set elements: 10 20
After erase(), Set size: 1
```

1. Vector Module
2. Deque Module
3. List Module
4. Set Module
5. Map Module
6. Stack Module
7. Queue Module
8. Exit

Enter your choice:5

```
Map size: 2
Map elements: apple:10 banana:20
After erase(), Map size: 1
```

1. Vector Module
2. Deque Module
3. List Module
4. Set Module
5. Map Module
6. Stack Module
7. Queue Module

8. Exit

Enter your choice:6

```
Stack size: 3
Stack top: 30
```

1. Vector Module
2. Deque Module
3. List Module
4. Set Module
5. Map Module
6. Stack Module
7. Queue Module
8. Exit

Enter your choice:7

```
Queue size: 3
Queue front: 10
Queue back: 30
```

1. Vector Module
 2. Deque Module
 3. List Module
 4. Set Module
 5. Map Module
 6. Stack Module
-

7. Queue Module

8. Exit

Enter your choice:8

Exiting...

Process finished with exit code 0
