Topic1

```cpp
#include <iostream>
using namespace std;
int main()
{
    string i;
    int ax[5] = {10, 20, 60, 40, 30};
    cout << "Enter Index :";
    getline(cin, i);
    try
    {
        if (stoi(i) < 0 || stoi(i) > 4)
        {
            throw "Index out of Range";
        }
        else if (stoi(i) > 0 && stoi(i) <= 4)
        {
            throw stoi(i);
        }
    }
    catch (char const *ex)
    {
        cout << ex << endl;
    }
    catch (int i)
    {
        cout << "a[" << i << "]=" << ax[i] << endl;
    }
    catch (...)
    {
        cout << "Your index is not an integer" << endl;
    }
}
```

Topic 2

```cpp
#include <iostream>
using namespace std;
template <typename G, typename R>
class A
{
    G x;
    R y;

public:
    void setData(G x, R y)
```

```
    {
        this->x = x;
        this->y = y;
    }
    template <typename O>
    O getSum()
    {
        return x + y;
    }
};
int main()
{
    A<int, int> a;
    a.setData(3, 4);
    cout << "The sum is " << a.getSum<int>() << endl;
    A<int, double> b;
    b.setData(3, 4.4);
    cout << "The sum is " << b.getSum<double>() << endl;
    A<double, double> c;
    c.setData(3.3, 4.4);
    cout << "The sum is " << c.getSum<double>() << endl;
    A<double, int> d;
    d.setData(3.3, 4.4);
    cout << "The sum is " << c.getSum<double>() << endl;
}
```

Topic 3

```
#include <iostream>
#include <array>
using namespace std;
int main()
{
    array<int, 6> ar = {10, 60, 30, 70, 20}, ar1 = {1, 6, 3, 7, 2};
    cout << "The third element  is : " << ar.at(2) << endl;
    cout << "The First element is : " << ar.front() << endl;
    cout << "The Last element is : " << ar.back() << endl;
    if (ar.empty())
    {
        cout << "Array is empty";
    }
    cout << "Before Swap" << endl;
    for (auto i = 0; i < ar.size(); i++)
        if (ar[i] != 0)
            cout << ar[i] << " ";
```

```
    cout << endl;
    ar.fill(100);
    cout << "After Filling 100" << endl;
    for (auto i = 0; i < ar.size(); i++)
        if (ar[i] != 0)
            cout << ar[i] << " ";
    cout << endl;
    ar.swap(ar1);
    cout << "After Swap" << endl;
    for (auto i = 0; i < ar.size(); i++)
        if (ar[i] != 0)
            cout << ar[i] << " ";
    cout << endl;
    cout << "The size of the array = " << ar.size() << endl;
    cout << "The maximum size of the array = " << ar.size() << endl;
    cout << "The address of the first element of the array is " << ar.begin() <<
endl;
    cout << "The address of the last element of the array is " << ar.end() <<
endl;
}
```

Topic 4

```
#include <iostream>
#include <utility>
using namespace std;
int main()
{
    pair<int, string> pa = make_pair(10, "Rajshahi");
    cout << "Before Swap" << endl;
    cout << "First Data : " << pa.first << endl;
    cout << "Second Data : " << pa.second << endl;
    get<int>(pa) = 20;
    pair<int, string> pb = make_pair(30, "Khulna");
    pa.swap(pb);
    cout << "After Swap" << endl;
    cout << "First Data : " << pa.first << endl;
    cout << "Second Data : " << pa.second << endl;
}
```

```
#include <iostream>
#include <utility>
#include <tuple>
```

Topic 5

```cpp
using namespace std;
int main()
{
    tuple<int, string, double> tx;
    tx = make_tuple(100, "Kamal", 3.5);
    cout << "Before Swap" << endl;
    cout << "First Data : " << get<int>(tx) << endl;
    cout << "Second Data : " << get<string>(tx) << endl;
    cout << "Third Data : " << get<double>(tx) << endl;
    get<double>(tx) = 3.7;
    tuple<int, string, double> bx = make_tuple(200, "Emon", 5.6);
    tx.swap(bx);
    cout << "After Swap" << endl;
    cout << "First Data : " << get<int>(tx) << endl;
    cout << "Second Data : " << get<string>(tx) << endl;
    cout << "Third Data : " << get<double>(tx) << endl;
}
```

Topic 6

```cpp
#include <iostream>
#include <stack>
using namespace std;
int main()
{
    stack<int> st;
    for (int i = 0; i < 10; i++)
        st.push(i);
    cout << "Before Pop and After Push" << endl;
    for (int i = 0; i < 10; i++)
    {
        cout << st.top() << " ";
        st.pop();
    }
    cout << endl;
    if (st.empty())
    {
        cout << "Stack is Empty" << endl;
    }
}
```

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

Topic 7

```
int main()
{
    queue<int> hello;
    for (int i = 1; i <= 6; i++)
        hello.push(i);
    cout << hello.size() << " " << hello.back() << endl;
    for (int i = 1; i <= 6; i++)
    {
        cout << hello.front() << " ";
        hello.pop();
    }
    cout << endl;
    if (hello.empty())
    {
        cout << "Stack is Empty" << endl;
    }
}
```

Topic 8

```
#include <iostream>
#include <iterator>
#include <algorithm>
#include <list>
using namespace std;
bool isOdd(int x)
{
    return (x % 2 != 0);
}
bool isEven(int x)
{
    return (x % 2 == 0);
}
void display(list<int> j)
{
    list<int>::iterator k;
    for (k = j.begin(); k != j.end(); k++)
        cout << *k << " ";
    cout << endl;
}
void revdisplay(list<int> O)
{
    for (auto m = O.rbegin(); m != O.rend(); m++)
        cout << *m << " ";
    cout << endl;
```

```
}

int main()
{
    list<int> linkedli;
    list<int> linkedli2;
    int a[17];
    a[0] = 1;
    a[1] = 3;
    for (int i = 1; i <= 16; i += 2)
        linkedli.push_back(i);
    for (int i = 2; i <= 16; i += 2)
    {
        a[i] = i;
        a[i + 1] = i + 2 * i;
        linkedli2.push_back(i);
    }
    linkedli.push_back(7);
    linkedli.push_back(7);
    display(linkedli);
    revdisplay(linkedli);
    cout << "Front = " << linkedli.front() << endl;
    cout << "Back = " << linkedli.back() << endl;
    linkedli.pop_back();
    linkedli.pop_front();
    display(linkedli);
    list<int>::iterator k;
    int x = 9;
    k = find(linkedli.begin(), linkedli.end(), 9);
    int y = distance(linkedli.begin(), k);
    cout << "Found at index " << y << "" << endl;
    linkedli.insert(k, 100);
    display(linkedli);
    linkedli.insert(next(k), 400);
    display(linkedli);
    cout << "7 is counted " << count(linkedli.begin(), linkedli.end(), 7) << "
times" << endl;
    cout << "there are " << count_if(linkedli.begin(), linkedli.end(), isOdd) <<
" odd numbers" << endl;
    linkedli.erase(k);
    display(linkedli);
    list<int>::iterator range_end = linkedli.begin();
    advance(range_end, 4);
    linkedli.erase(linkedli.begin(), range_end);
    cout << "Deleted first 3 Elements " << endl;
```

```
    display(linkedli);
    linkedli.push_back(9);
    linkedli.push_back(12);
    linkedli.push_back(13);
    display(linkedli);
    linkedli.remove(x);
    linkedli.remove_if(isEven);
    display(linkedli);
    linkedli.assign(linkedli2.begin(), linkedli2.end());
    display(linkedli);
    linkedli.assign(a, a + 17);
    cout << "Before sorting : " << endl;
    display(linkedli);
    linkedli.sort();
    cout << "After sorting : " << endl;
    display(linkedli);
    cout << "Before Unique : " << endl;
    display(linkedli);
    linkedli.unique();
    cout << "After Unique : " << endl;
    display(linkedli);
}
```