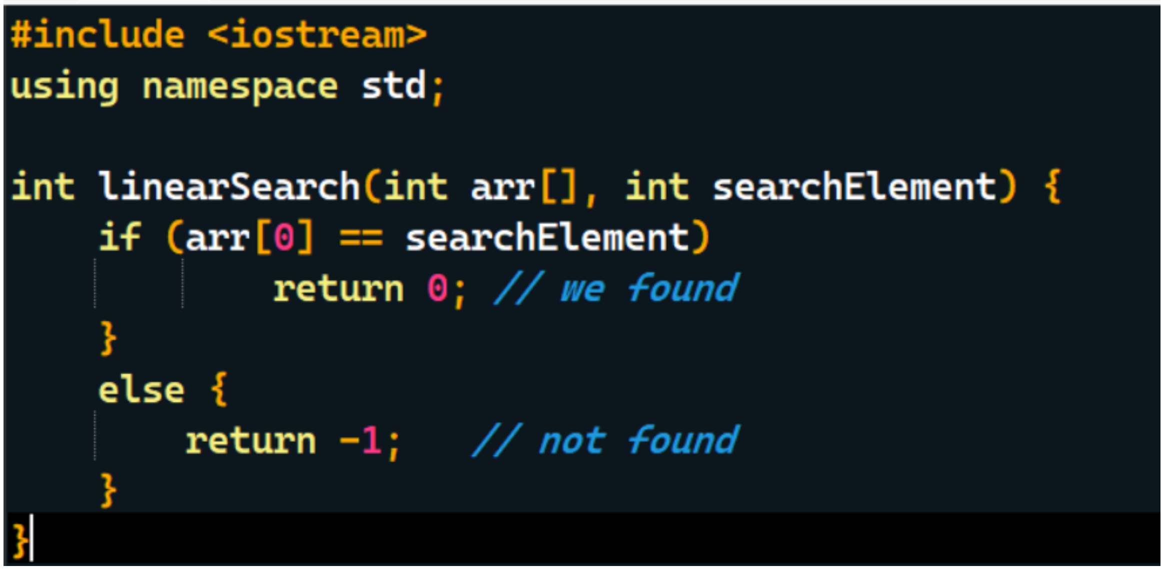
Summary DSA

Sessions No 05(13-12-2022)

- For solving searching problem we have multiple algorithm available & In DSA we have some standard algorithm such as,
    1) **Linear Search**
    2) **Binary Search**
    3) **Hashtable**

- **Linear search**:
    - ➔ Linear means a straight line we have to go one by one. CPU will pick one data from RAM and do something then pick next data then do something and so on.
    - ➔ We have data ie. **unordered** therefore we don't know about the next data. Therefore we have to search data one by one.

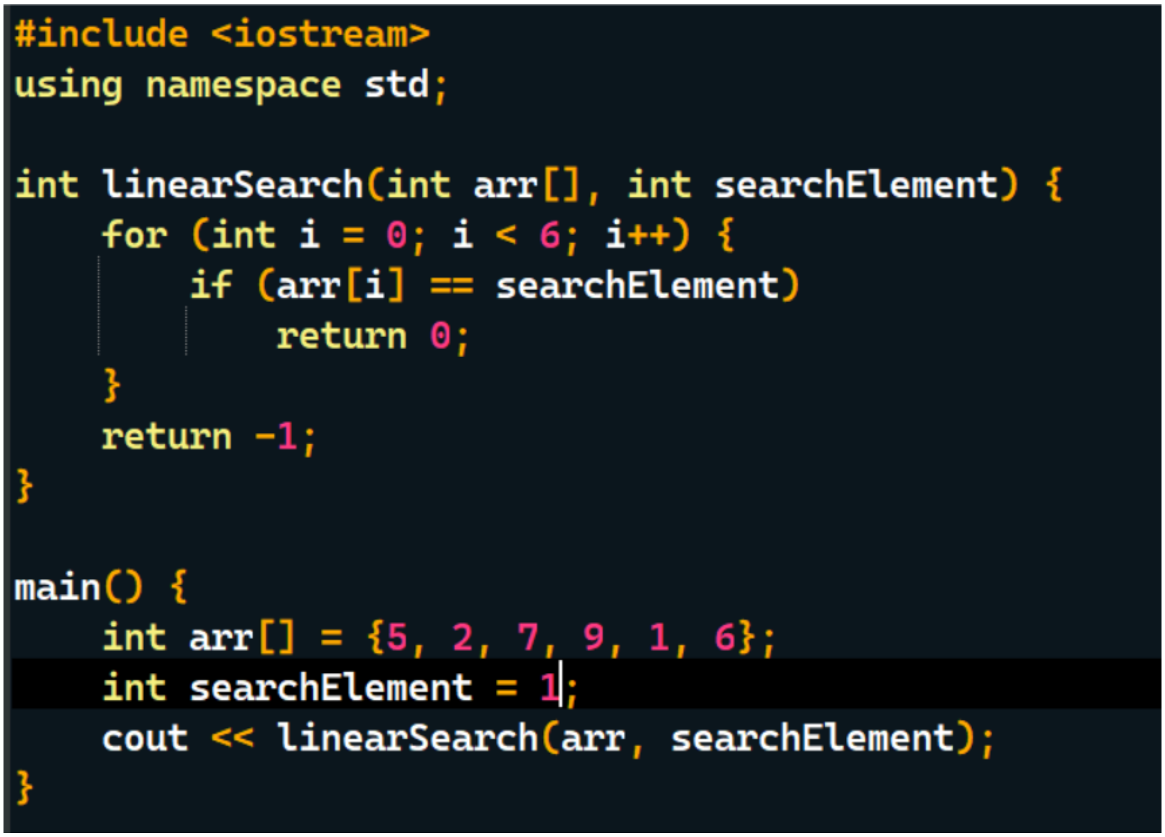


```
#include <iostream>
using namespace std;

int linearSearch(int arr[], int searchElement) {
    if (arr[0] == searchElement)
            return 0; // we found
    }
    else {
        return -1;   // not found
    }
}
```

- Above code is about only the first element search. It only need 2 CPU time therefore time complexity is O(1) ie. constant.

- But we have to search data in the whole array. And if we want to traverse into the array then we can used any loop eg. for loop.



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

int linearSearch(int arr[], int searchElement) {
    for (int i = 0; i < 6; i++) {
        if (arr[i] == searchElement)
            return 0;
    }
    return -1;
}

main() {
    int arr[] = {5, 2, 7, 9, 1, 6};
    int searchElement = 1;
    cout << linearSearch(arr, searchElement);
}
```

- Above code will go into the array and check one element with search element one by one. And if the search element is found then return 0 and if till last element is not found then return -1.

- Above code is hard coded. If our data size is changed then code will not work properly.