Riphah International University
I-14 Main Campus
Faculty of Computing

Class:	Fall-2024	**Subject:**	Data Structures & Algorithms
Course Code:	CS 2124	**Lab Instructor:**	Zeeshan Ali

Learning Objective:

1. What is Static Array and why do we use Static Array?
2. Example of static array in C++.
3. What is Dynamic Array and why do we use Dynamic Array?
4. Example of dynamic array in C++.
5. Linear Search & and its working.
6. Linear Search in C++
7. Tasks

1. Static Array

Static array is a **fixed-size** collection of elements that are stored in **contiguous memory locations** and have a **predetermined** size that cannot change during runtime. Static arrays are one of the simplest and most common data structures used in C++ and other programming languages. Here's a more detailed explanation of static arrays in C++.

Declaration and Initialization:

To declare a static array, you **specify the data type** of its elements followed by the **array name** and the **size of the array** in square brackets (**[]**).

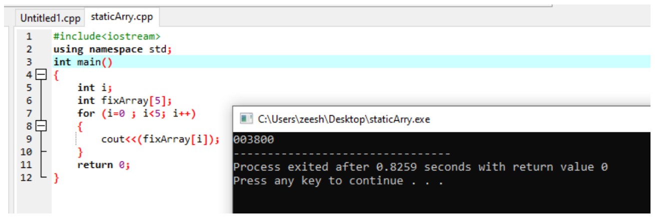
For example:

```
int fixArray[5]; // Declares an integer array with 5 elements
```

You can also initialize a static array at the **time of declaration**:

```
int statArray[5] = {1, 2, 3, 4, 5}; // Initializes the array with specific values
```

If you don't specify an initial value, the elements of the array will be uninitialized, and their **values** will be **unpredictable**.



```
#include<iostream>
using namespace std;
int main()
{
    int i;
    int fixArray[5];
    for (i=0 ; i<5; i++)
    {
        cout<<(fixArray[i]);
    }
    return 0;
}
```

```
003800
--------------------------------
Process exited after 0.8259 seconds with return value 0
Press any key to continue . . .
```

Size:
The **size** of a static array is determined at **compile-time** and cannot be changed during program execution. In the example above, statArray has a fixed size of 5.

Accessing Elements:
Elements in a static array are accessed using **zero-based** indexing, which means the first element is at index 0, the second at index 1, and so on.

You can access elements using the array name followed by the index in square brackets:

int element = statArray [3]; // Accesses the forth element of the array (index 3)

Common Operations:
Static arrays are suitable for operations that require **constant-time** access to elements. For example, you can easily retrieve or update the value of an element at a **specific index**.

Looping through the elements of a static array is a common operation for tasks like searching, sorting, or performing calculations on the array elements.

Limitations:
One of the main limitations of static arrays is their **fixed size**. You **must know** the **maximum** number of elements you need when defining the array.

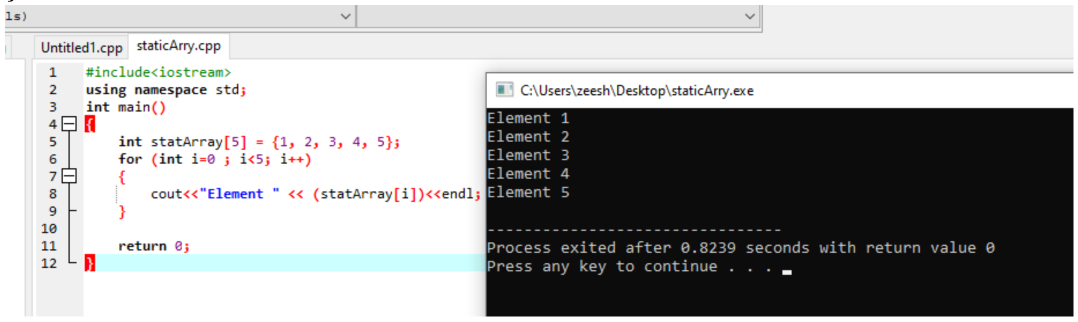
Memory Allocation:
Static arrays allocate **memory** for all their elements when the array is **created**. The **memory** is typically allocated on the **stack**, but you can also allocate it on the heap using pointers if necessary.

2. Example of static array in C++
Simple example of using a static array in C++:

```
#include<iostream>
using namespace std;
int main()
{
        int statArray[5] = {1, 2, 3, 4, 5};
        for (int i=0 ; i<5; i++)
  {
        cout<<"Element " << (statArray[i])<<endl;
        }

  return 0;
```

}



```
#include<iostream>
using namespace std;
int main()
{
    int statArray[5] = {1, 2, 3, 4, 5};
    for (int i=0 ; i<5; i++)
    {
        cout<<"Element " << (statArray[i])<<endl;
    }

    return 0;
}
```

```
Element 1
Element 2
Element 3
Element 4
Element 5

--------------------------------
Process exited after 0.8239 seconds with return value 0
Press any key to continue . . .
```

This program declares and initializes a static integer array with five elements and then prints each element's value using a loop.

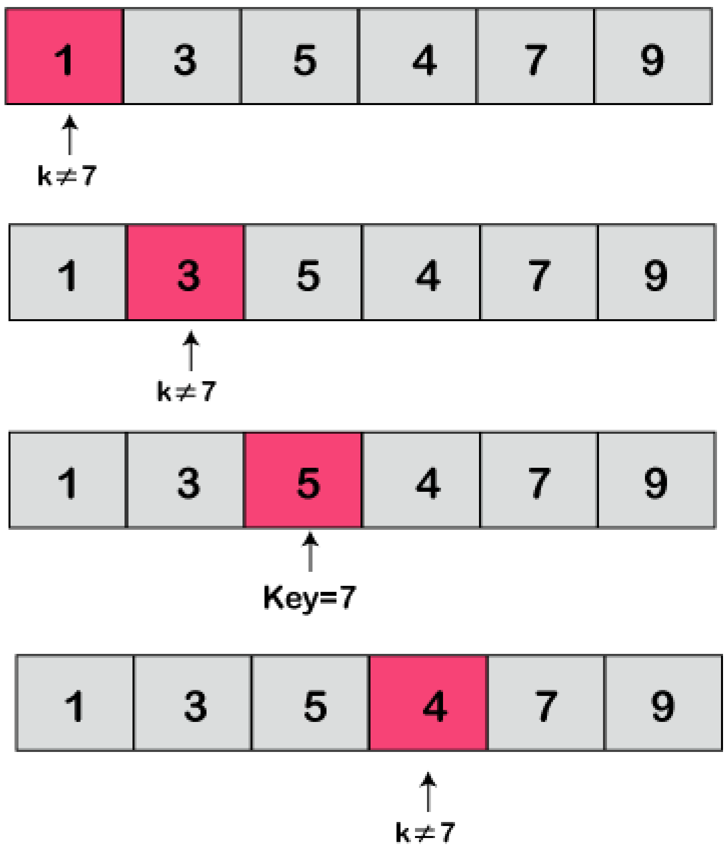
3. Dynamic Array

A dynamic array, also known as a **resizable** array or a dynamic array list, is a data structure that allows you to store a collection of elements, similar to a traditional static array, but with the key advantage of **automatic resizing**. In a dynamic array, the size of the array can **grow or shrink** as needed, making it more **flexible** and **convenient** than a static array. Dynamic arrays are commonly used in many programming languages, including C++, Python (e.g., list), and Java (e.g., ArrayList).

Here are the main characteristics and operations associated with dynamic arrays:

Dynamic Sizing:

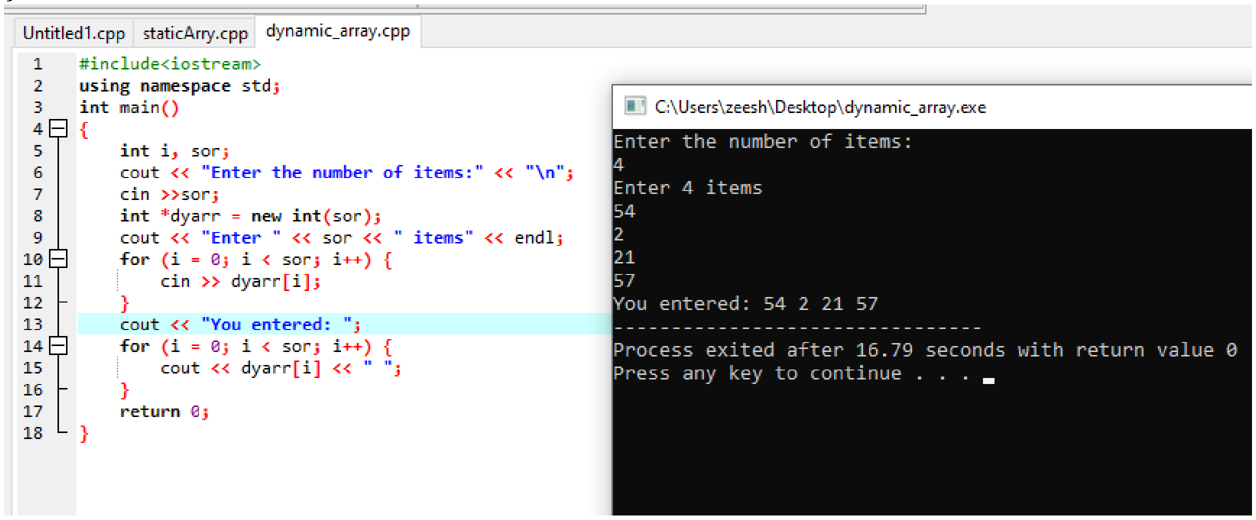
Dynamic arrays automatically adjust their size to accommodate the number of elements they contain. This means that you **don't** need to **specify the size** of the array at compile time.



4. Example of dynamic array in C++

#include<iostream>
using namespace std;
int main()
{
	int i, sor;
	cout << "Enter the number of items:" << "\n";
	cin >>sor;
	int *dyarr = new int(sor);

```
	cout << "Enter " << sor << " items" << endl;
	for (i = 0; i < sor; i++) {
		cin >> dyarr[i];
	}
	cout << "You entered: ";
	for (i = 0; i < sor; i++) {
		cout << dyarr[i] << " ";
	}
	return 0;
}
```



```
#include<iostream>
using namespace std;
int main()
{
	int i, sor;
	cout << "Enter the number of items:" << "\n";
	cin >>sor;
	int *dyarr = new int(sor);
	cout << "Enter " << sor << " items" << endl;
	for (i = 0; i < sor; i++) {
		cin >> dyarr[i];
	}
	cout << "You entered: ";
	for (i = 0; i < sor; i++) {
		cout << dyarr[i] << " ";
	}
	return 0;
}
```

```
Enter the number of items:
4
Enter 4 items
54
2
21
57
You entered: 54 2 21 57
--------------------------------
Process exited after 16.79 seconds with return value 0
Press any key to continue . . .
```

5. Linear Search

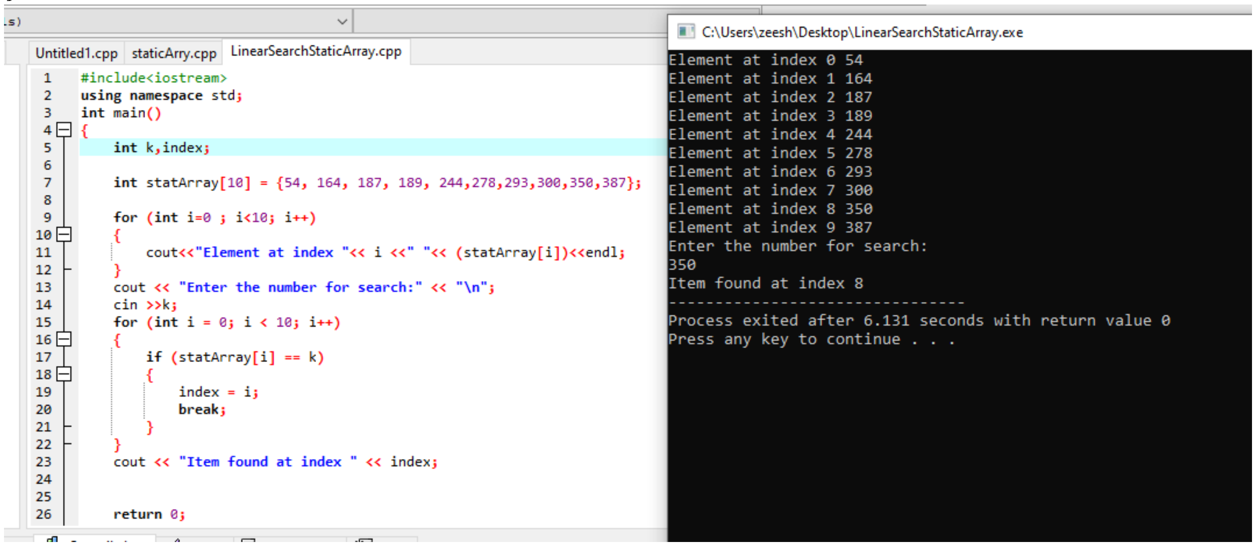
A linear search algorithm or sequential search is a method for finding an element within a list. It sequentially checks each element of the list until a match is found or the whole list has been searched.

Linear Search Working

6. Linear Search in C++

```
#include<iostream>
using namespace std;
int main()
{
	int k,index;
	int statArray[10] = {54, 164, 187, 189, 244,278,293,300,350,387};
	for (int i=0 ; i<10; i++)
   {
	cout<<"Element at index "<< i <<" "<< (statArray[i])<<endl;
	}
	cout << "Enter the number for search:" << "\n";
```

```
        cin >>k;
        for (int i = 0; i < 10; i++)
        {
                if (statArray[i] == k)
                {
                        index = i;
                        break;
                }
        }
        cout << "Item found at index " << index;
        return 0;
}
```


Untitled1.cpp
staticArry.cpp
LinearSearchStaticArray.cpp
C:\Users\zeesh\Desktop\LinearSearchStaticArray.exe

```
#include<iostream>
using namespace std;
int main()
{
    int k,index;

    int statArray[10] = {54, 164, 187, 189, 244,278,293,300,350,387};

    for (int i=0 ; i<10; i++)
    {
        cout<<"Element at index "<< i <<" "<< (statArray[i])<<endl;
    }
    cout << "Enter the number for search:" << "\n";
    cin >>k;
    for (int i = 0; i < 10; i++)
    {
        if (statArray[i] == k)
        {
            index = i;
            break;
        }
    }
    cout << "Item found at index " << index;


    return 0;
```

```
Element at index 0 54
Element at index 1 164
Element at index 2 187
Element at index 3 189
Element at index 4 244
Element at index 5 278
Element at index 6 293
Element at index 7 300
Element at index 8 350
Element at index 9 387
Enter the number for search:
350
Item found at index 8
--------------------------------
Process exited after 6.131 seconds with return value 0
Press any key to continue . . .
```

7. Tasks

- Write down a program in C++ that take an age of 10 students as an input from user and display the largest age of the student from an array.
- Write down a program in C++ that take an input data from user in three different arrays and then add the arrays and store them in another array. (Through Dynamic array concept).
- Write a program for linear search using the concept of dynamic array (Note: Program should handle the situation if item is not in the list).