Tishk International University
Science Faculty
IT Department

Programming II - IT-118

Arrays

1st Grade - Fall Semester 2023

Lecture #1

Instructor: Hemin Ibrahim
Email: hemin.ibrahim@tiu.edu.iq

Overview

- ✓ Concept of Array
- ✓ Defining Array
- ✓ Array Initialization
- ✓ Inputting Array Contents
- ✓ Processing Array Contents
- ✓ Displaying Array Contents
- ✓ Search inside an array
- ✓ Two Dimensional Array
- ✓ Multi Dimensional Array

Class policy

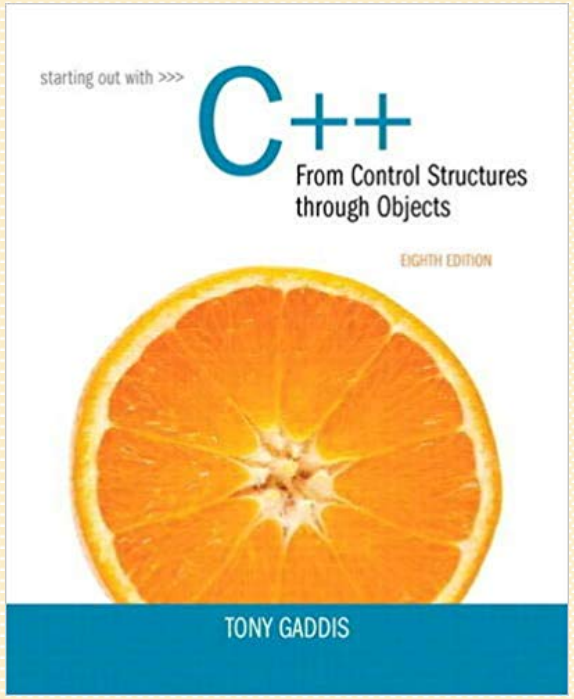
Mobile

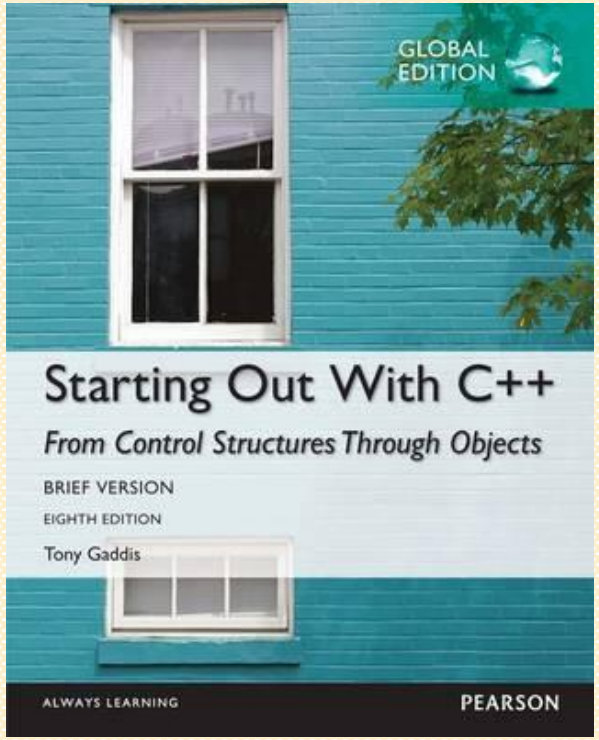
- ✓ Please keep your mobile phone on silent mode.
- ✓ Taking pictures inside the class without permission is prohibited.
- ✓ If you receive a call, kindly step out before answering it.
- ✓ If you need to send a message, please step out and do so.
- ✓ Don’t call me.
- ✓ Kindly avoid sending voice messages and stick to written messages only.

Textbook Source

Tony Gaddis, Starting Out with C++: From Control Structures through Objects, 8th Edition

✓ Chapter 8 (from page 485 to 534)



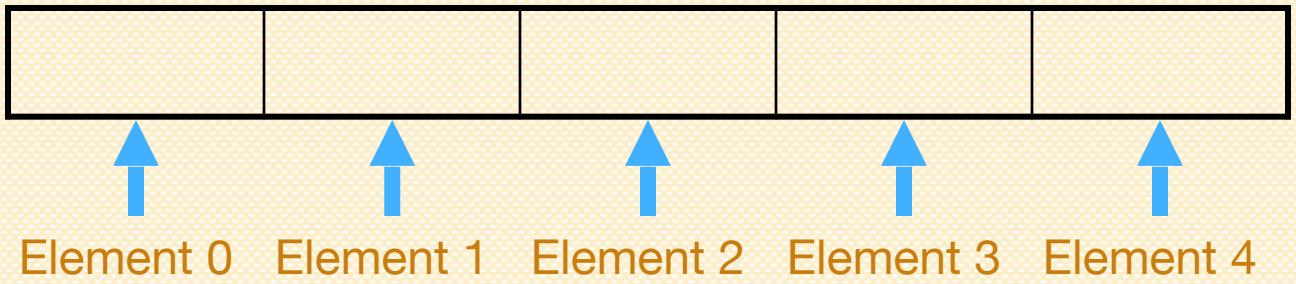


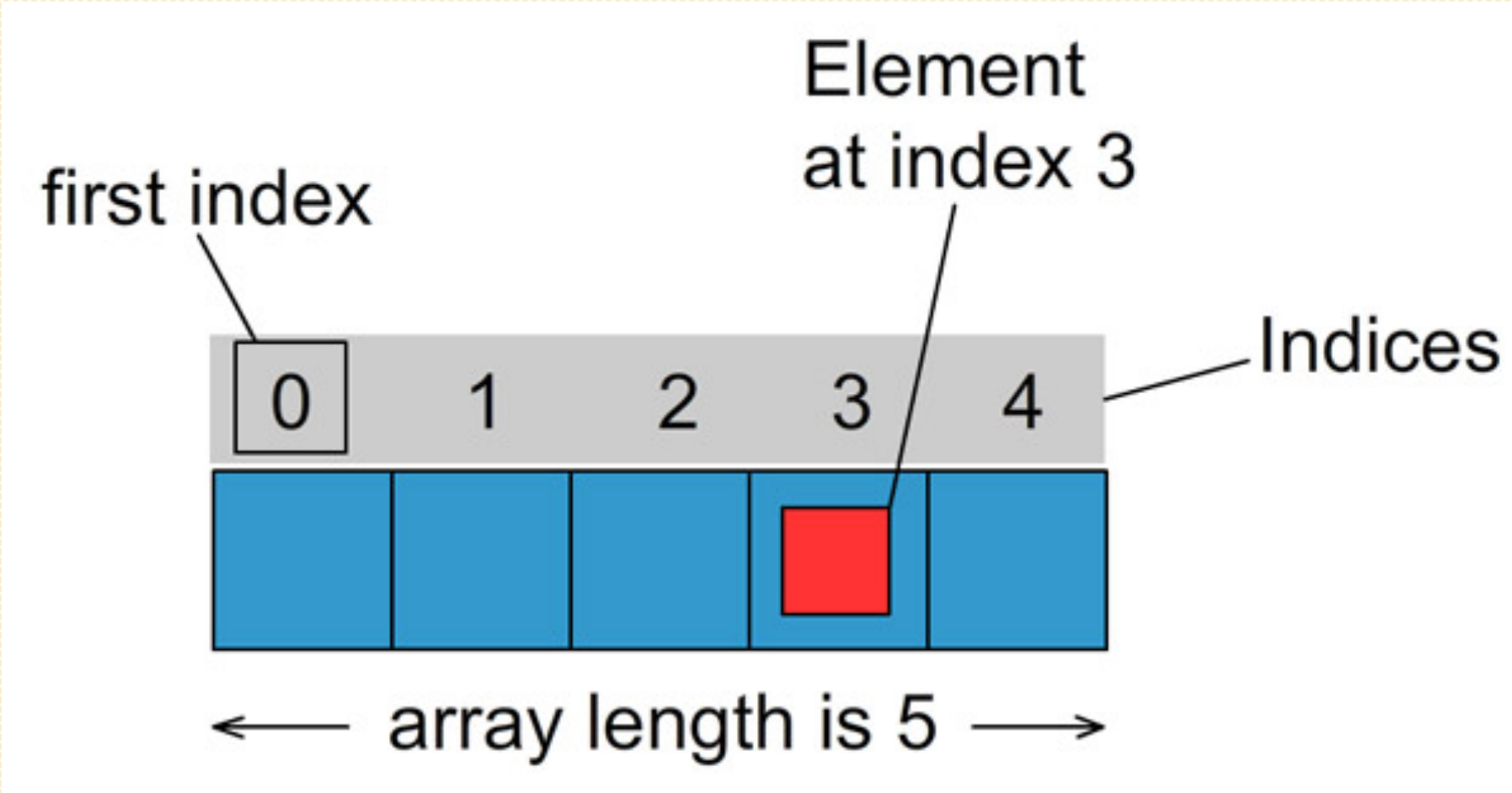
Array Concept

- Array: a variable that can store multiple values of the same type
- Values are stored in adjacent memory locations
- Declared using [] operator

  int A[5];

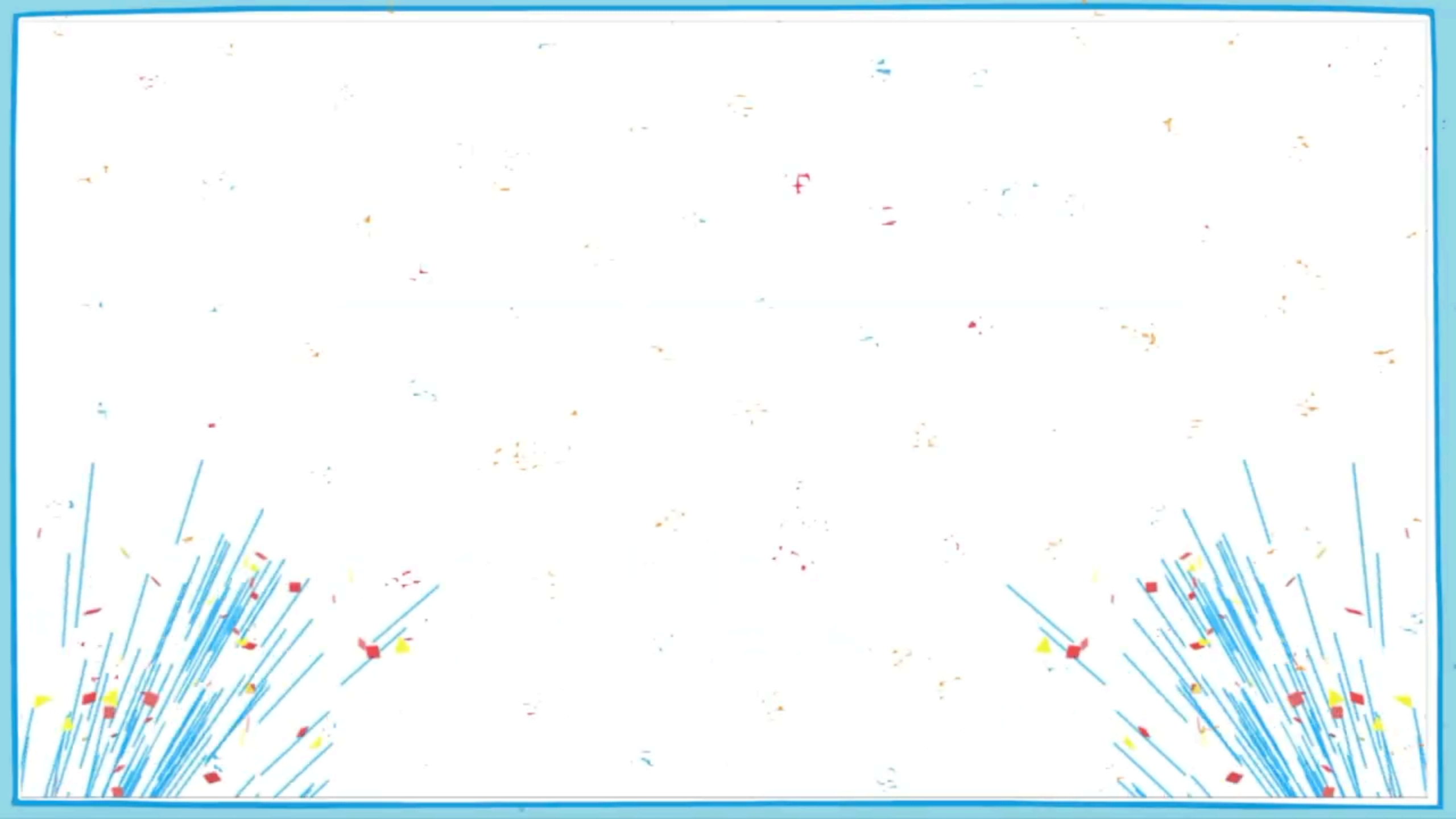
This allocates the following memory locations




Element
at index 3
first index
Indices
0
1
2
3
4
array length is 5


We need to talk
OK
RUN CMD
You seem to have more time for your computer than me. I want to know how important I am to you
RUN
You're my number 1 in my life
Just that I start counting from 0


Array Index starts at 1



Example

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

int main() {
  int arr[] = {1, 2, 3, 4, 5};
  cout << "The second element of the array is: " << arr[1] << endl;

  return 0;
}
```

Output: The second element of the array is: 2

Exercise

- C++ program that initializes an array of integers with the values 6, 3, 8, 9, 1, 0, and 4. Then, print out the values located at index 1 and index 4 of the array. Also, change the values of index 1 and 2 to 45 and 88, and print them.

Defining Array

In the definition

```
int A[5]; //holds 5 integer cells
```

- ***int*** *is the data type of the array elements*
- ***A*** *is the name of the array*
- ***5*** *is the number of elements. It shows the number of elements in the array.*
- *Other Example*

```
double volumes[10];//holds 10 double cells
```

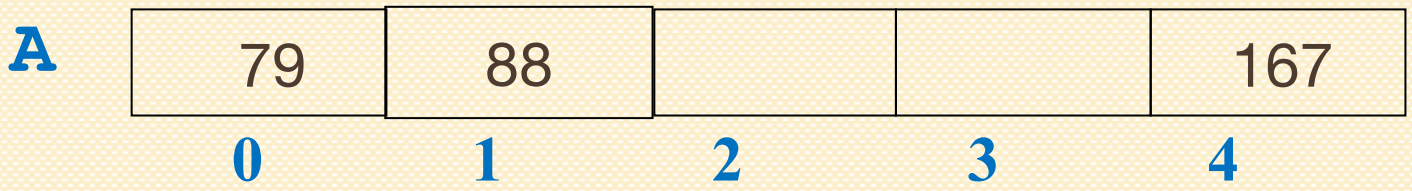
Array Size Declarators

Array declaration	Number of elements	Size of each element	Size of the array
`char letter[26];`	26	1 byte	26 bytes
`short ring[100];`	100	2 bytes	200 bytes
`int mile[84];`	84	4 bytes	336 bytes
`float temp[12];`	12	4 bytes	48 bytes
`double distance[1000];`	1000	8 bytes	8,000 bytes

Accessing Array Elements

Array elements (accessed by array name and subscript) can be used as regular variables


```
5       int A[5];
6       A[0] = 79;
7       cout << A[0];
8       cin  >> A[1]; //user puts 88
9       A[4] = A[0] + A[1];
```



Array Initialization

- Can be initialized during program execution with assignment statements

```
A[0] = 79;
A[1] = 82; // etc.
```

- Can be initialized at array definition with an initialization list

```
int A[5] = {79,82,91,77,84};
Or
int A[5] = {0};//all have zero value
int A[5] = {4};//only the first element has value 4,
  others are zero
```

Implicit Array Sizing

- Can determine array size by the size of the initialization list

```
short quizzes[]={12, 17, 15, 11};
```

- Must use either array size declarator or initialization list when array is defined

Inputting Array Contents

`cin` can be used to input values from keyboard and store these values into an array element

```
int A[5]; // Define 5-cells array
cout << "Enter first number ";
cin >> A[0];
```

Processing Array Contents

- Array elements can be
  - *treated as ordinary variables of the same type as the array*
  - *used in arithmetic operations, in relational expressions, etc.*
- Example:

```
if (A[3] >= 10000){
  x = A[3] * 0.5;
} else {
  x = A[3] * 0.75;
}
```

```
#include <iostream>
using namespace std;
//Ali, aza, ara, akam
int main() {
  int A[4]={1,2,3,4};
    A[1]=A[0]+3;
    A[3]=A[1]+A[3];

    cout<<A[0]<<" , "<<A[1]<<" , "<<A[2]<<" , "<<A[3];
    return 0;
}
```

Displaying Array Contents and the size of array

cout can be used to display value of the value of an array element

```
int A[]={10,3,4,1,6}; // Define 5-cells array
cout << "A[0]= " << A[0] <<endl;

int size = sizeof(A)/sizeof(A[0]);
cout<<size<<endl; // 5
```

Array Subscripts

- Array subscript (index) can be an **integer constant**, **integer variable**, or **integer expression**
- Example:

```cpp
#include <iostream>
using namespace std;
int main() {
  // Declare an array
  int arr[] = {10, 12, 0, 4, 3};

  // Accessing array elements using integer constants
  cout << "Element at index 0: " << arr[0] << endl; // Output: 10
  cout << "Element at index 2: " << arr[2] << endl; // Output: 0

  // Accessing array elements using integer variables
  int index = 3;
  cout << "Element at index " << index << ": " << arr[index] << endl; // Output: 4

  // Accessing array elements using integer expressions
  int i = 1, j = 2;
  cout << "Element at index " << i+j << ": " << arr[i+j] << endl; // Output: 4

  return 0;
}
```


How do you execute a code block 100 times?
I just copy-paste it 100 times.
imgflip.com

Inputting All Array Elements

To access each element of an array

- *Use a loop*
- *Let the loop control variable be the array subscript*
- *A different array element will be referenced each time through each cycle of the loop*

```
 5  int A[5];
 6  for (int i = 0; i < 5; i++)
 7  {
 8      cout << “Enter Some numbers in to the array ";
 9      cin >> A[i];
10  }
```

Displaying All Array Elements

To display each element of an array

- *Use a loop*
- *Let the loop control variable be the array subscript*
- *A different array element will be referenced each time through each cycle of the loop*

```
5  int A[]={9,4,3,7,2};
6
7  for(int i = 0; i < 5; i++){
8        cout << A [i] << endl;
9  }
```

Sample input and output program of array

```cpp
#include <iostream>
using namespace std;
int main() {

int A[5];
for (int i = 0; i < 5; i++){
    cout << “Enter Some numbers in to the array ";
    cin >> A[i];
}

for (int i = 0; i < 5; i++){
    cout << A[i]<<endl;
}

  return 0;
}
```

Output

```
Enter Some numbers in to the array 3
Enter Some numbers in to the array 5
Enter Some numbers in to the array 12
Enter Some numbers in to the array 5
Enter Some numbers in to the array 0
3
5
12
5
0
```

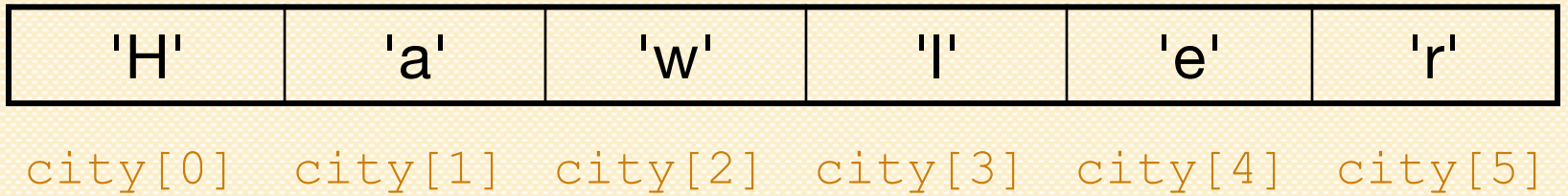
Strings and string Objects

String is a special type of array of characters.

It Can be processed using array name

- *Entire string at once, or*
- *One element at a time by using a subscript*

```
string city;
cout << "Enter city name: ";
cin >> city;
```



Strings and string Objects

```
2  #include <iostream>
3  using namespace std;
4  int main() {
5
6  string city;
7  cout << "Enter city name: ";
8  cin >> city;
9
10 for(int i=0;i<6;i++){
11     cout<<city[i]<<endl;
12 }
13
14   return 0;
15 }
```

```
Enter city name: Hawler
H
a
w
l
e
r
```

Months and Days Example

```
// This program displays the number of days in each month. It uses an
 // array of string objects to hold the month names and an int array
 // to hold the number of days in each month. Both are initialized with
 // initialization lists at the time they are created.
#include <iostream>
using namespace std;
#include <iomanip>
int main(){
    const int NUM_MONTHS = 12;
    string name[NUM_MONTHS] = {"January", "February", "March","April", "May",
        "June","July", "August", "September","October", "November", "December"};

    int days[NUM_MONTHS] = { 31, 28, 31, 30,31, 30, 31, 31,30, 31, 30, 31 };

    for (int month = 0; month < NUM_MONTHS; month++){
        cout << setw(9) << left << name[month] << " has ";
        cout << days[month] << " days.\n";
    }

    return 0;
}
```

Program Output

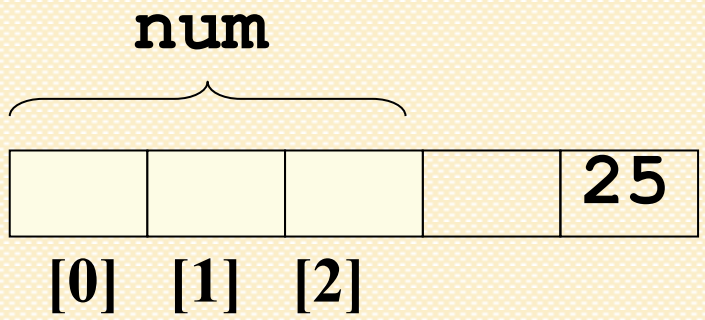
```
January   has 31 days.
February  has 28 days.
March     has 31 days.
April     has 30 days.
May       has 31 days.
June      has 30 days.
July      has 31 days.
August    has 31 days.
September has 30 days.
October   has 31 days.
November  has 30 days.
December  has 31 days.
```

Notes about Arrays

NOTE 1: No Bounds Checking

- There are no checks in C++ that an array subscript is in range
- An invalid array subscript can cause program to overwrite other memory locations
- Example:

```
int num[3]; //composed of =>num[0], num[1], num[2]
int i = 4;
num[i] = 25;//we don't have num[4]
```



NOTE 2: Using Increment and Decrement Operators with Array Elements

When using **++** and **--** operators, don't confuse the element with the subscript

```
A[i]++; // adds 1 to A[i]
A[i++]; // increments i, but has no effect on A
```

```
int score[5] = {7, 18, 9, 21, 11};
++score[2]; // Pre-increment operation on the value in score[2]
score[4]++; // Post-increment operation on the value in score[4]
```

NOTE 3: : Copying One Array to Another

```
int A1[]= {1,2,3,4,5};
int A2[5];
```

- Cannot copy with an assignment statement:

```
A2=A1 ;  //Not allowed
```

- But we can copy with an assignment statement inside a loop:

```
for (int i=0; i < 5; i++) {
A2[i] = A1[i];
}
```

Search inside an array

Write a program to search for an input inserted by user

```
5       const int size = 5;
6       int number;
7       int array[] = {1,2,3,4,5};
8       bool found = false;
9       cout << "search for a number: ";
10      cin >> number;
11      for (int i = 0; i < size; i++){
12          if (array[i] == number) {
13              found = true;
14              cout << "at index " << i ;
15              break;
16          }
17      }
18      if (found){
19          cout << " Element found Successfully" << endl;
20      } else {
21          cout << "Element not exist" << endl;
22      }
```

Vowel counter application

```
#include <iostream>
using namespace std;
int main(){
    char ch;
    int vowelCount = 0;
    string sentence;

    cout << "Enter any sentence you wish and I will \n tell you how many vowels are
        in it.\n";
    getline(cin, sentence);

    for (int i = 0; i < sentence.length(); i++){
        ch = tolower(sentence[i]);
        switch (ch){
            case 'a':
            case 'e':
            case 'i':
            case 'o':
            case 'u': vowelCount++;
       }
    }
    cout << "There are " << vowelCount << " vowels in the sentence.\n";

    return 0;
}
```

This program illustrates how a string can be processed as an array of individual characters. It reads in a string, then counts the number of vowels in the string.
using the toupper or tolower functions is important to either uppercase or lowercase each letter in the string and the string class member. function length() to determine how many characters are in the string.

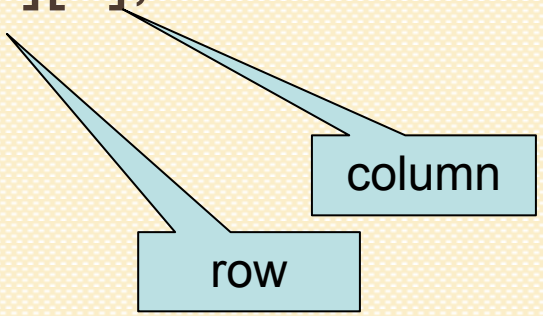
Vowel counter application (Using IF Statement)

```
#include <iostream>
using namespace std;
int main(){
    char ch;
    int vowelCount = 0;
    string sentence;

    cout << "Enter any sentence you wish and I will \n tell you how many vowels are
        in it.\n";
    getline(cin, sentence);

    for (int i = 0; i < sentence.length(); i++){
        ch = tolower(sentence[i]);
        if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u'){
            vowelCount++;
        }

    }
    cout << "There are " << vowelCount << " vowels in the sentence.\n";

    return 0;
}
```

Two Dimensional Array

A two-dimensional array can be think of as a table, which will have x number of rows and y number of columns.

```
int arr[2][3];
```



int arr[2][3] =

	0	1	2	
0	arr[0][0]	arr[0][1]	arr[0][2]	Row 0
1	arr[1][0]	arr[1][1]	arr[1][2]	Row 1
	Col 0	Col 1	Col 2	

Two Dimensional Array

```
6   int test[2][3]= {{2, -5, 6},{4, 0, 9}};
```

```
6   int test[2][3]= {
7       {2, -5, 6},
8       {4, 0, 9}
9       };
```

	1	2	3
1	2	-5	6
2	4	0	9

Two Dimensional Array

```
2   #include <iostream>
3   using namespace std;
4   int main(){
5
6       int test[2][3] = {{2, -5, 6}, {4, 0, 9}};
7
8       for (int i = 0; i < 2; i++){
9           for (int j = 0; j < 3; j++){
10              cout << test[i][j] << "\t";
11          }
12          cout << endl;
13      }
14      return 0;
15  }
```

Two Dimensional Array - Input and print values

```cpp
#include <iostream>
using namespace std;
int main(){

    int A[2][3];

    for (int i = 0; i < 2; i++){
        for (int j = 0; j < 3; j++){
            cout << "Enter value for row " << i+1 << " column " << j+1 << ": ";
            cin >> A[i][j];
        }
    }

    for (int i = 0; i < 2; i++){
        for (int j = 0; j < 3; j++){
            cout << A[i][j] <<"\t";
        }
        cout<<endl;
    }
    return 0;
}
```

Three Dimensional Array

```
int A[2][3][4] = {
    {
        {1, 2, 3, 4},
        {5, 6, 7, 8},
        {9, 10, 11, 12}
    },
    {
        {13, 14, 15, 16},
        {17, 18, 19, 20},
        {21, 22, 23, 24}
    }
};
```

Three Dimensional Array

Declare three dimensional array and ask user to input the values. After that print the array.

```
The output is:
0 1 2 3
1 2 3 4
2 3 4 5

1 2 3 4
2 3 4 5
3 4 5 6
```

```cpp
#include <iostream>
using namespace std;
int main()
{

int arr[2][3][4];
for(int i=0; i<2; i++) {
      for(int j=0; j<3; j++) {
         for(int k=0; k<4; k++) {
            cout<<"Input the number "<<endl;
            cin>> arr[i][j][k];
         }
      }
   }

   // Print the array
   for(int i=0; i<2; i++) {
      for(int j=0; j<3; j++) {
         for(int k=0; k<4; k++) {
            cout << arr[i][j][k] << " ";
         }
         cout << endl;
      }
      cout << endl;
   }
return 0;
}
```

Temperature Readings:

```
#include <iostream>
using namespace std;
int main() {
    int NUM_DAYS = 7;
    int NUM_HOURS = 24;
    double temperatureData[NUM_DAYS][NUM_HOURS];

    for (int day = 0; day < NUM_DAYS; day++) {
        for (int hour = 0; hour < NUM_HOURS; hour++) {
            temperatureData[day][hour] = getTemperature(day, hour);
        }
    }

    return 0;
}
```

Multi Dimensional Array - Example

Example: Ask user to decide the number of student. Then, create 3 arrays for name, surname and city, and let him insert the data. After that create a double dimensional array and send all the data in to the double dimensional array. Then, print the information.

Multi Dimensional Array

Ask user to decide the number of student. Then, create 3 arrays for name, surname and city, and let him insert the data. After that create a double dimensional array and send all the data in to the double dimensional array. Then, print the information.

```
int size;
cout << "Enter the number of students: ";
cin >> size;
```

Multi Dimensional Array

Ask user to decide the number of student. Then, create 3 arrays for name, surname and city, and let him insert the data. After that create a double dimensional array and send all the data in to the double dimensional array. Then, print the information.

```
string name[size];
string surname[size];
string city[size];
```

Multi Dimensional Array

Ask user to decide the number of student. Then, create 3 arrays for name, surname and city, and let him insert the data. After that create a double dimensional array and send all the data in to the double dimensional array. Then, print the information.

```
string name[size];
string surname[size];
string city[size];
```

```
for (int i = 0; i < size; i++) {
    cout << "Enter name, surname, and city for element " << i + 1 << ": ";
    cin >> name[i] >> surname[i] >> city[i];
}
```

Multi Dimensional Array

Ask user to decide the number of student. Then, create 3 arrays for name, surname and city, and let him insert the data. After that create a double dimensional array and send all the data in to the double dimensional array. Then, print the information.

```
string data[size][3];
```

Multi Dimensional Array

Ask user to decide the number of student. Then, create 3 arrays for name, surname and city, and let him insert the data. After that create a double dimensional array and send all the data in to the double dimensional array. Then, print the information.

```
for (int i = 0; i < size; i++) {
    data[i][0] = name[i];
    data[i][1] = surname[i];
    data[i][2] = city[i];
}
```

Multi Dimensional Array

Ask user to decide the number of student. Then, create 3 arrays for name, surname and city, and let him insert the data. After that create a double dimensional array and send all the data in to the double dimensional array.

Then, print the information.

```
cout << "----- Student Information ------" << endl;
for (int i = 0; i < size; i++) {
    cout << data[i][0] << " " << data[i][1] << " " << data[i][2] << endl;
}
```

Multi Dimensional Array

```cpp
#include <iostream>
using namespace std;
int main() {
    int size;
    cout << "Enter the number of students: ";
    cin >> size;

    string name[size];
    string surname[size];
    string city[size];

    for (int i = 0; i < size; i++) {
        cout << "Enter name, surname, and city for element " << i + 1 << ": ";
        cin >> name[i] >> surname[i] >> city[i];
    }

    string data[size][3];

    for (int i = 0; i < size; i++) {
        data[i][0] = name[i];
        data[i][1] = surname[i];
        data[i][2] = city[i];
    }

    cout << "----- Student Information ------" << endl;
    for (int i = 0; i < size; i++) {
        cout << data[i][0] << " " << data[i][1] << " " << data[i][2] << endl;
    }

    return 0;
}
```

The output

```
Enter the number of students: 3
Enter name, surname, and city for element 1: Karzan Ali Hawler
Enter name, surname, and city for element 2: Balen Qadir Slemani
Enter name, surname, and city for element 3: Shadan Hozan Kirkuk
----- Student Information ------
Karzan Ali Hawler
Balen Qadir Slemani
Shadan Hozan Kirkuk
```