Lecture-11

Arrays and Pointers

Array – declaration & initialization

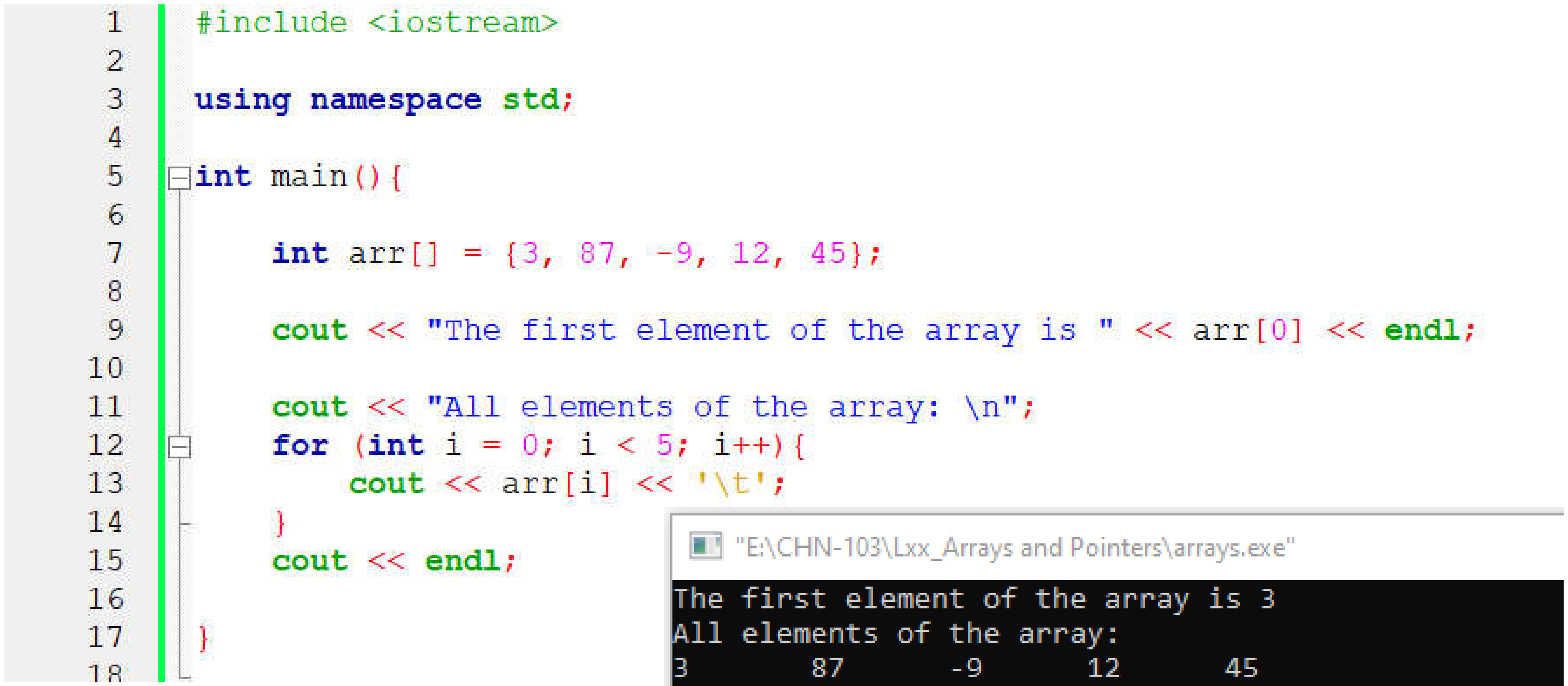
Arrays are sequence of variables of same datatype occupying contiguous location in memory, e.g. an array of integers

3	87	-9	12	45	78	98	23	0	-45

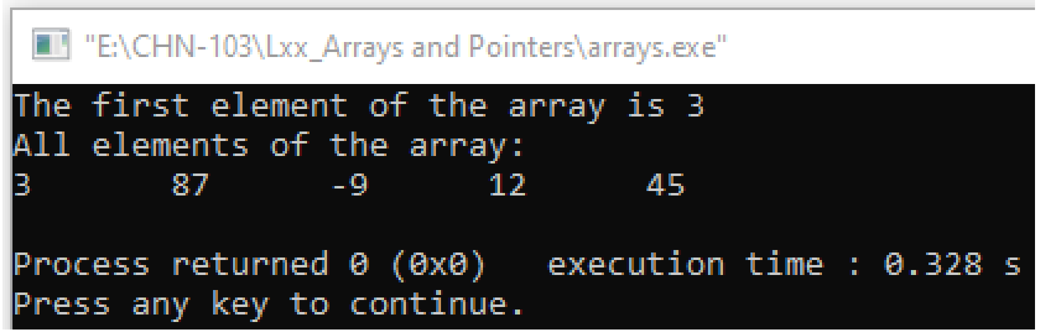
```
int arr[10];                      Declaration of an array of 10 integers

int arr[] = {3, 87, -9};          // universal initialization of an array
int arr[] {3,87,-9};

int arr[10] = {};                 // this is an array of 10 elements initialized to zeroes

int arr[];                        // not allowed
int arr[10] = {3, 87, -9};        // rest of the elements will be initialized to zeroes

int arr[3][3] = {};               // 3 x 3 array initialized with zeroes
int arr[][] = {{1,2,6},{3,4}};    // 2 x 3 array with elements [1 2 6] in first row and
                                  // [3,4,0] in second row
```

Array – Accessing arrays

- The indexing of arrays starts with 0.
- An index cannot be negative such as -3 or fractional number such as 2.34.



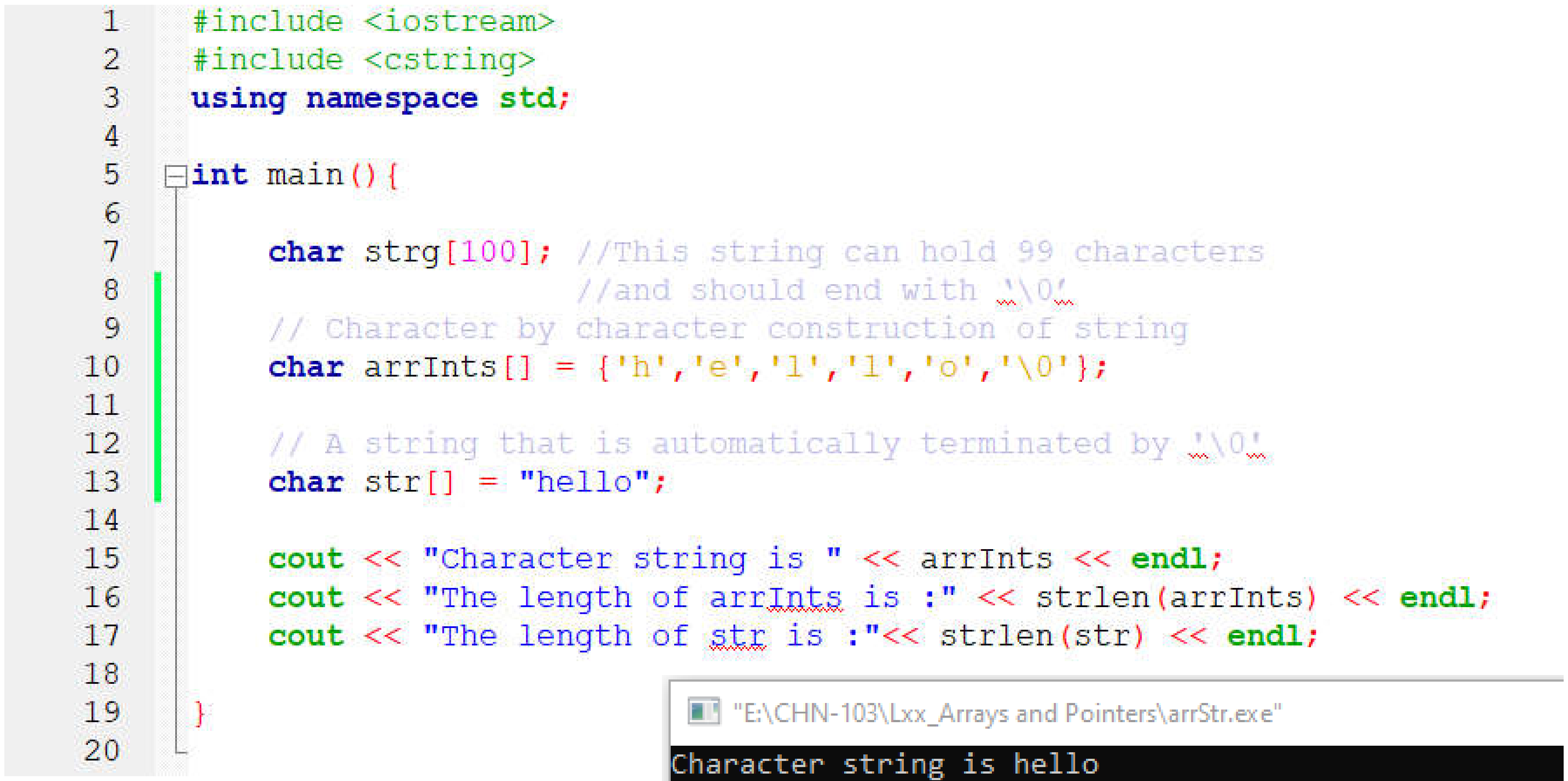
```cpp
#include <iostream>

using namespace std;

int main(){

    int arr[] = {3, 87, -9, 12, 45};

    cout << "The first element of the array is " << arr[0] << endl;

    cout << "All elements of the array: \n";
    for (int i = 0; i < 5; i++){
        cout << arr[i] << '\t';
    }
    cout << endl;

}
```



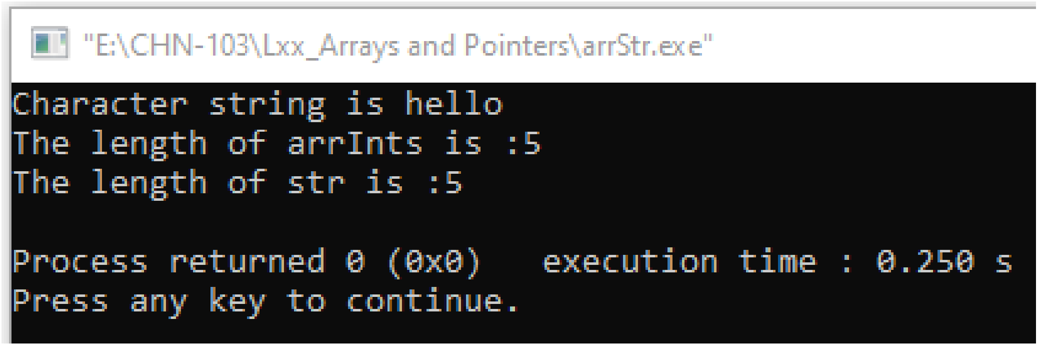
"E:\CHN-103\Lxx_Arrays and Pointers\arrays.exe"

```
The first element of the array is 3
All elements of the array:
3       87      -9      12      45

Process returned 0 (0x0)   execution time : 0.328 s
Press any key to continue.
```

Array of characters or strings

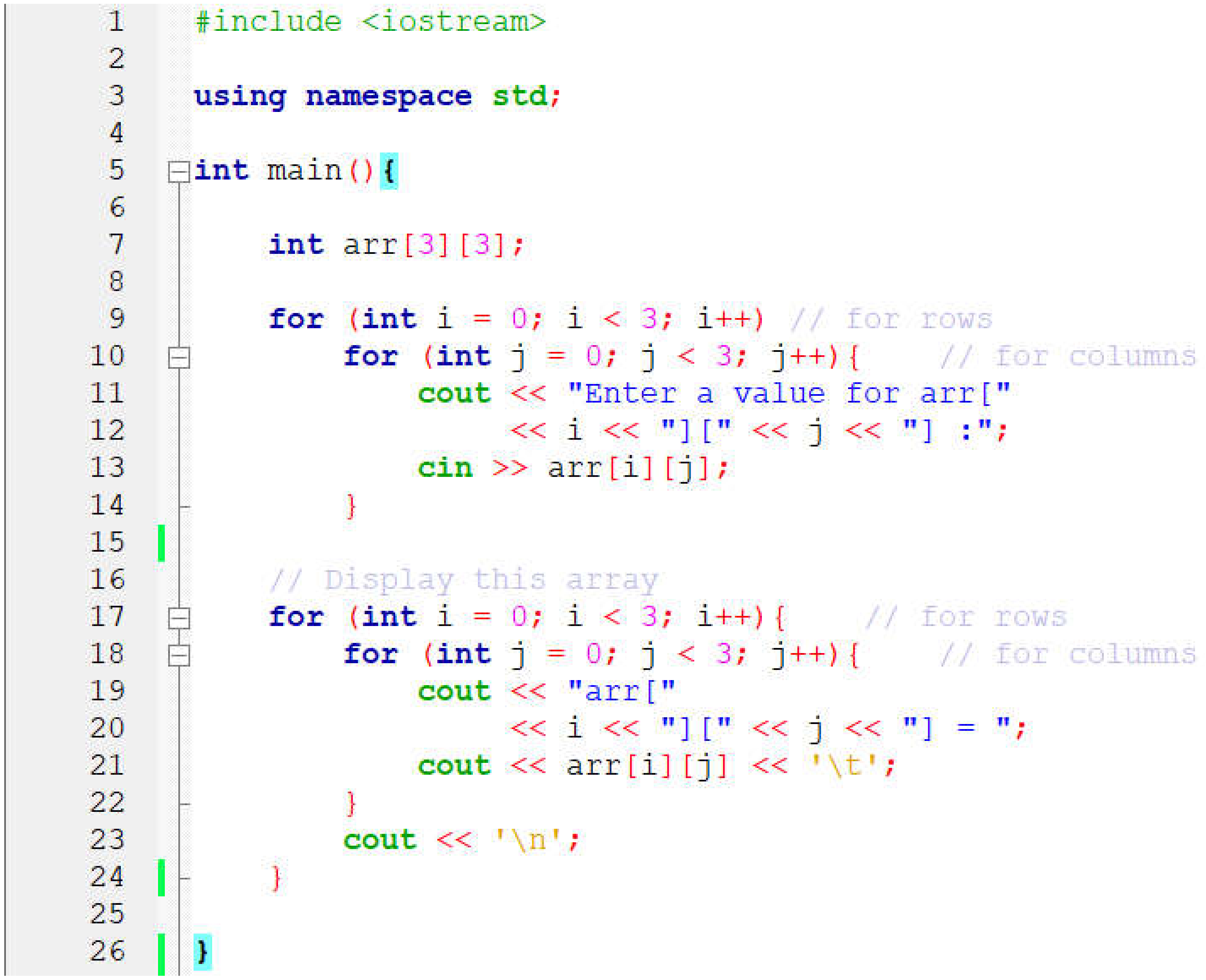


```cpp
#include <iostream>
#include <cstring>
using namespace std;

int main(){

    char strg[100]; //This string can hold 99 characters
                    //and should end with '\0'
    // Character by character construction of string
    char arrInts[] = {'h','e','l','l','o','\0'};

    // A string that is automatically terminated by '\0'
    char str[] = "hello";

    cout << "Character string is " << arrInts << endl;
    cout << "The length of arrInts is :" << strlen(arrInts) << endl;
    cout << "The length of str is :"<< strlen(str) << endl;

}
```



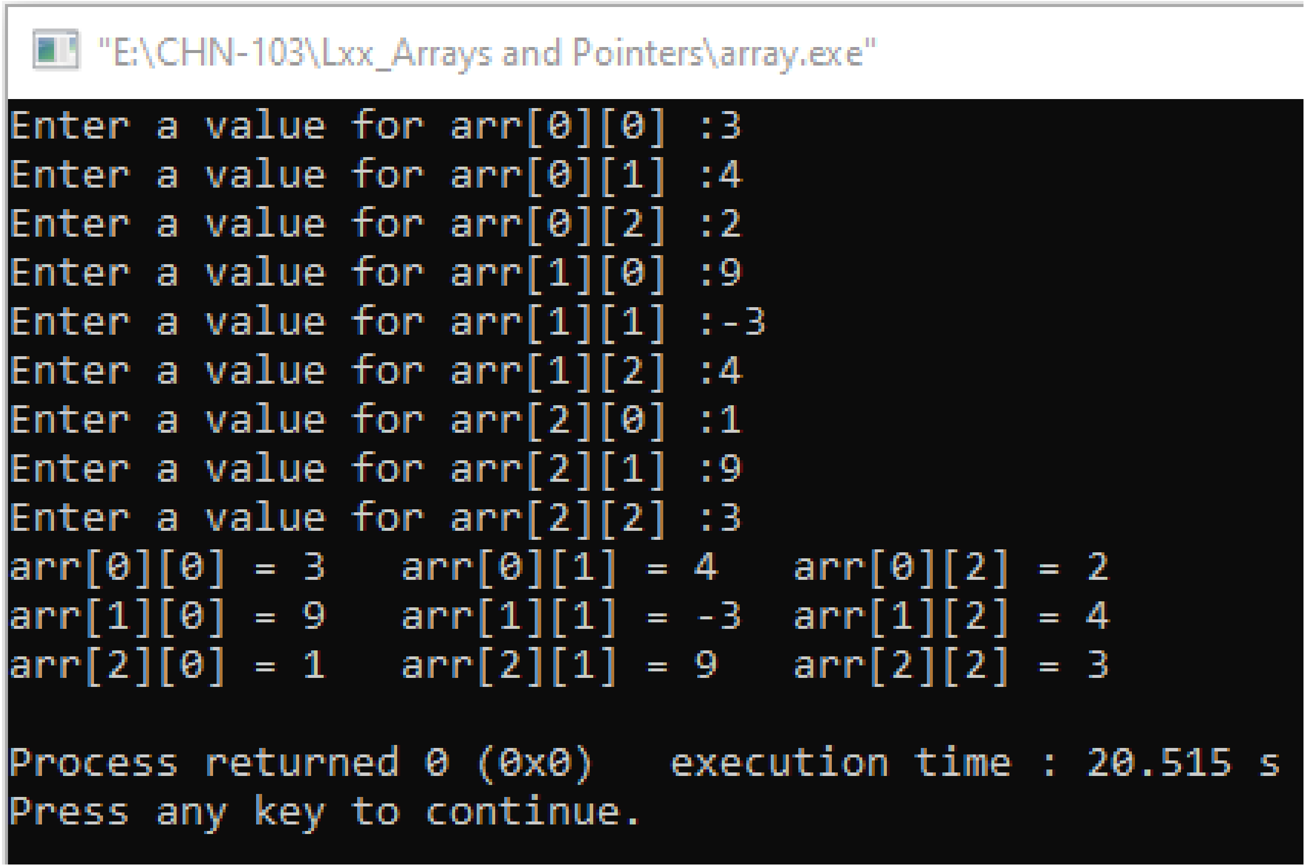
```
Character string is hello
The length of arrInts is :5
The length of str is :5

Process returned 0 (0x0)   execution time : 0.250 s
Press any key to continue.
```

Arrays - multidimensional



```
#include <iostream>

using namespace std;

int main(){

    int arr[3][3];

    for (int i = 0; i < 3; i++) // for rows
        for (int j = 0; j < 3; j++){     // for columns
            cout << "Enter a value for arr["
                 << i << "][" << j << "] :";
            cin >> arr[i][j];
        }

    // Display this array
    for (int i = 0; i < 3; i++){     // for rows
        for (int j = 0; j < 3; j++){     // for columns
            cout << "arr["
                 << i << "][" << j << "] = ";
            cout << arr[i][j] << '\t';
        }
        cout << '\n';
    }

}
```

Arrays - multidimensional



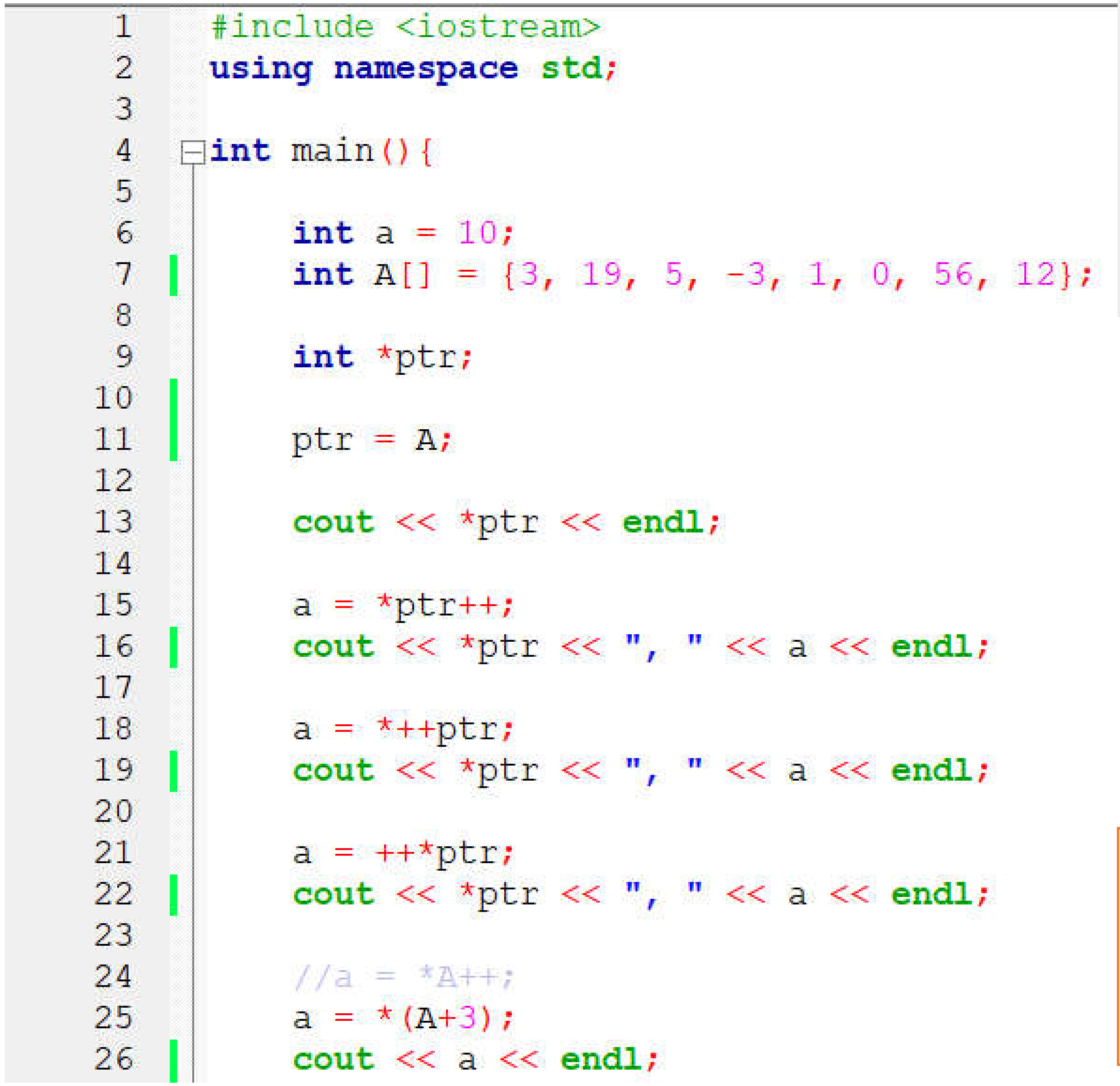
"E:\CHN-103\Lxx_Arrays and Pointers\array.exe"

```
Enter a value for arr[0][0] :3
Enter a value for arr[0][1] :4
Enter a value for arr[0][2] :2
Enter a value for arr[1][0] :9
Enter a value for arr[1][1] :-3
Enter a value for arr[1][2] :4
Enter a value for arr[2][0] :1
Enter a value for arr[2][1] :9
Enter a value for arr[2][2] :3
arr[0][0] = 3   arr[0][1] = 4   arr[0][2] = 2
arr[1][0] = 9   arr[1][1] = -3  arr[1][2] = 4
arr[2][0] = 1   arr[2][1] = 9   arr[2][2] = 3

Process returned 0 (0x0)   execution time : 20.515 s
Press any key to continue.
```

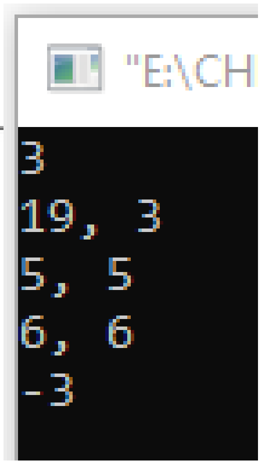
Multidimensional arrays do not reside in memory in this format, but in contiguous memory.

3	4	2	9	-3	4	1	9	3

Arrays and Pointers



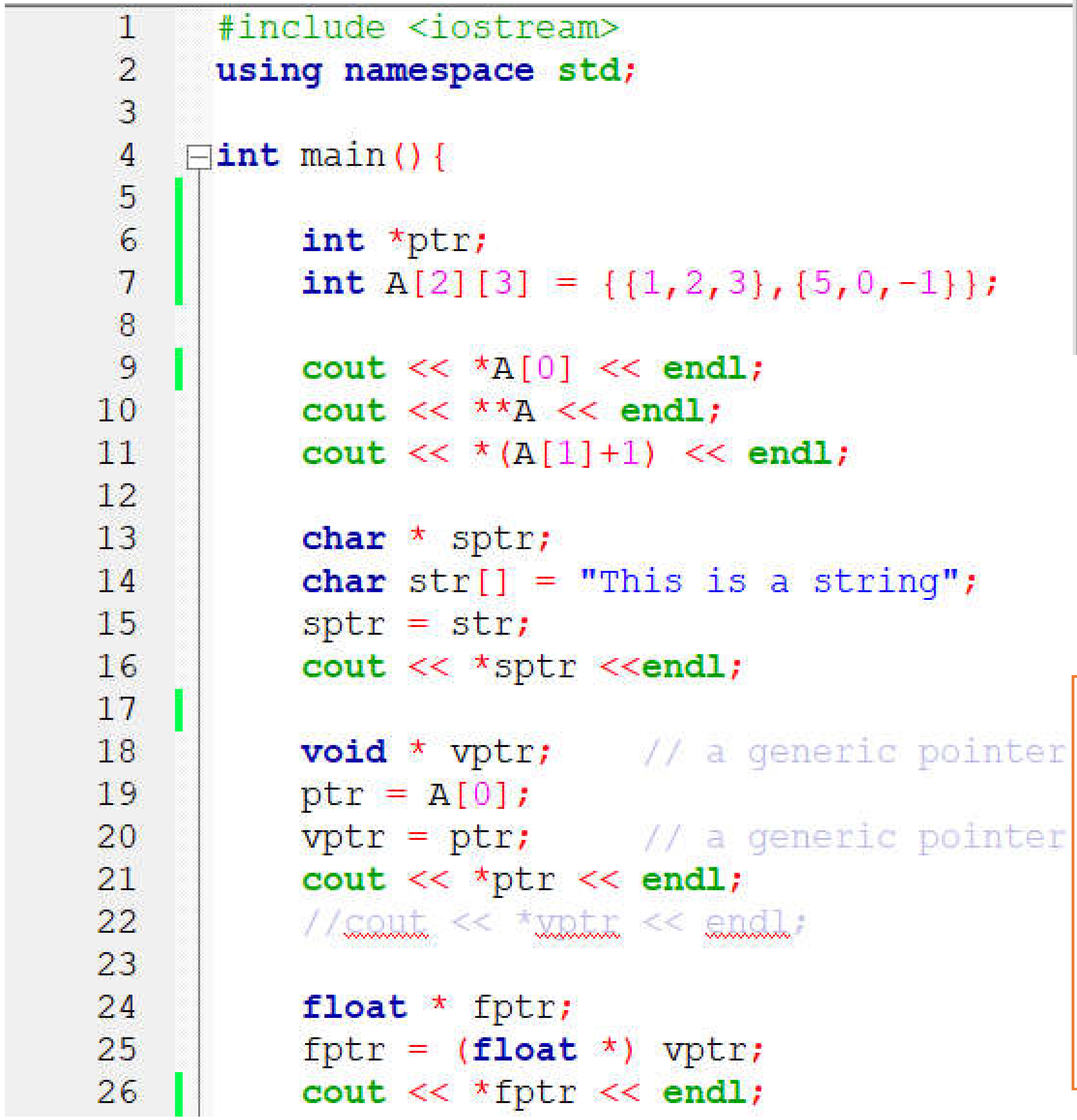
```
#include <iostream>
using namespace std;

int main(){

    int a = 10;
    int A[] = {3, 19, 5, -3, 1, 0, 56, 12};

    int *ptr;

    ptr = A;

    cout << *ptr << endl;

    a = *ptr++;
    cout << *ptr << ", " << a << endl;

    a = *++ptr;
    cout << *ptr << ", " << a << endl;

    a = ++*ptr;
    cout << *ptr << ", " << a << endl;

    //a = *A++;
    a = *(A+3);
    cout << a << endl;
```



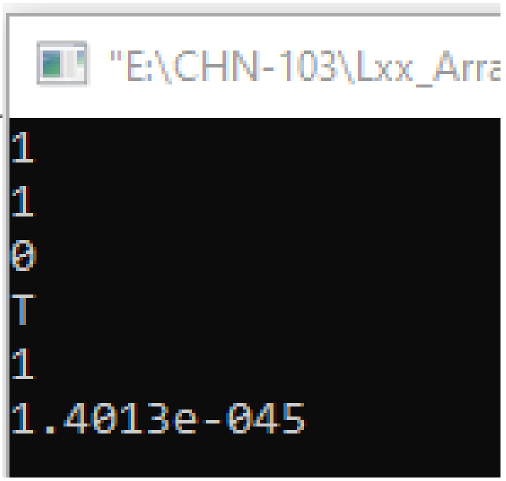
```
3
19, 3
5, 5
6, 6
-3
```

Array name is a constant pointer that points at the first element of the array

Arrays and Pointers



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

int main(){

    int *ptr;
    int A[2][3] = {{1,2,3},{5,0,-1}};

    cout << *A[0] << endl;
    cout << **A << endl;
    cout << *(A[1]+1) << endl;

    char * sptr;
    char str[] = "This is a string";
    sptr = str;
    cout << *sptr <<endl;

    void * vptr;     // a generic pointer
    ptr = A[0];
    vptr = ptr;      // a generic pointer
    cout << *ptr << endl;
    //cout << *vptr << endl;

    float * fptr;
    fptr = (float *) vptr;
    cout << *fptr << endl;
```

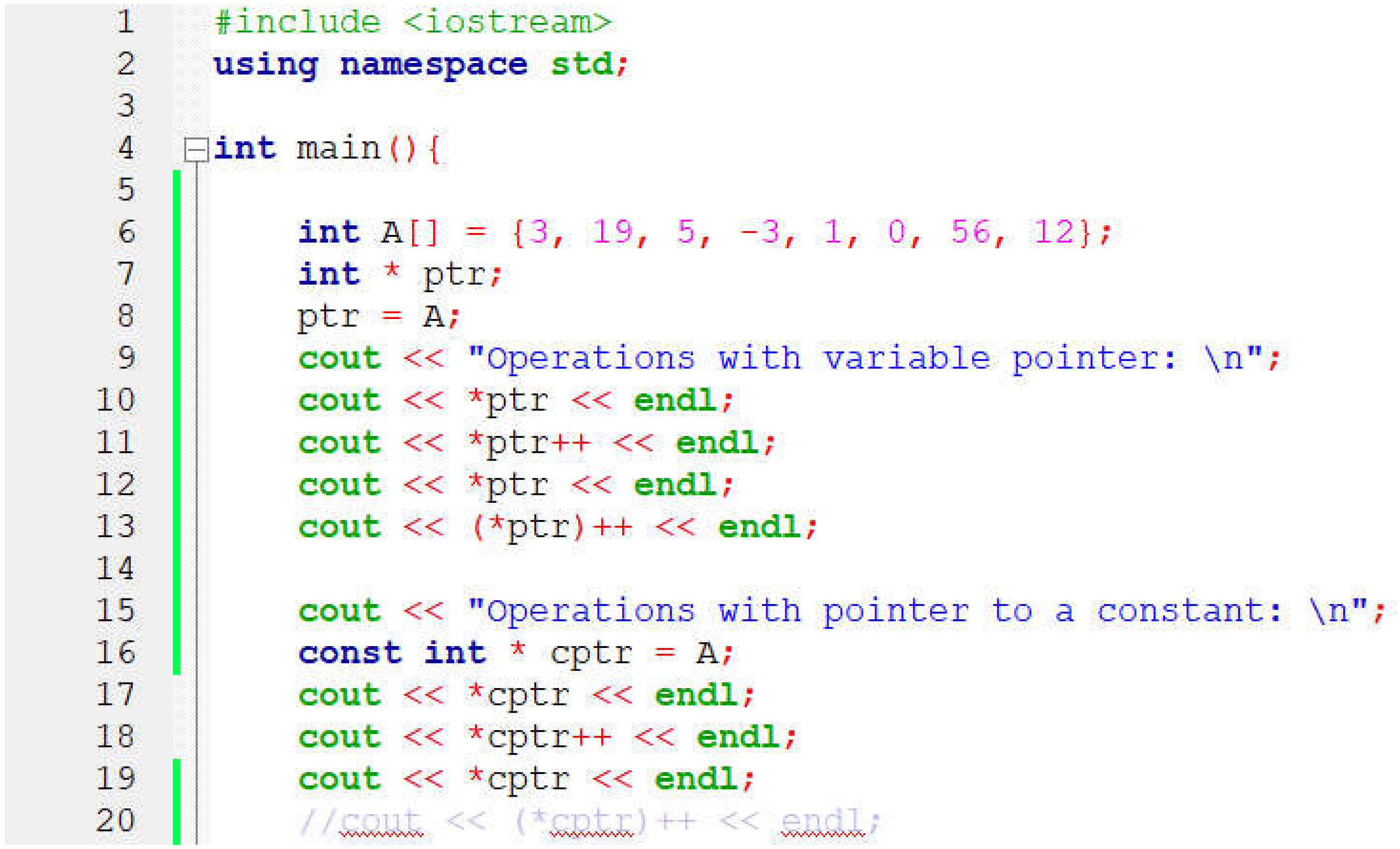


"E:\CHN-103\Lxx_Arra

```
1
1
0
T
1
1.4013e-045
```

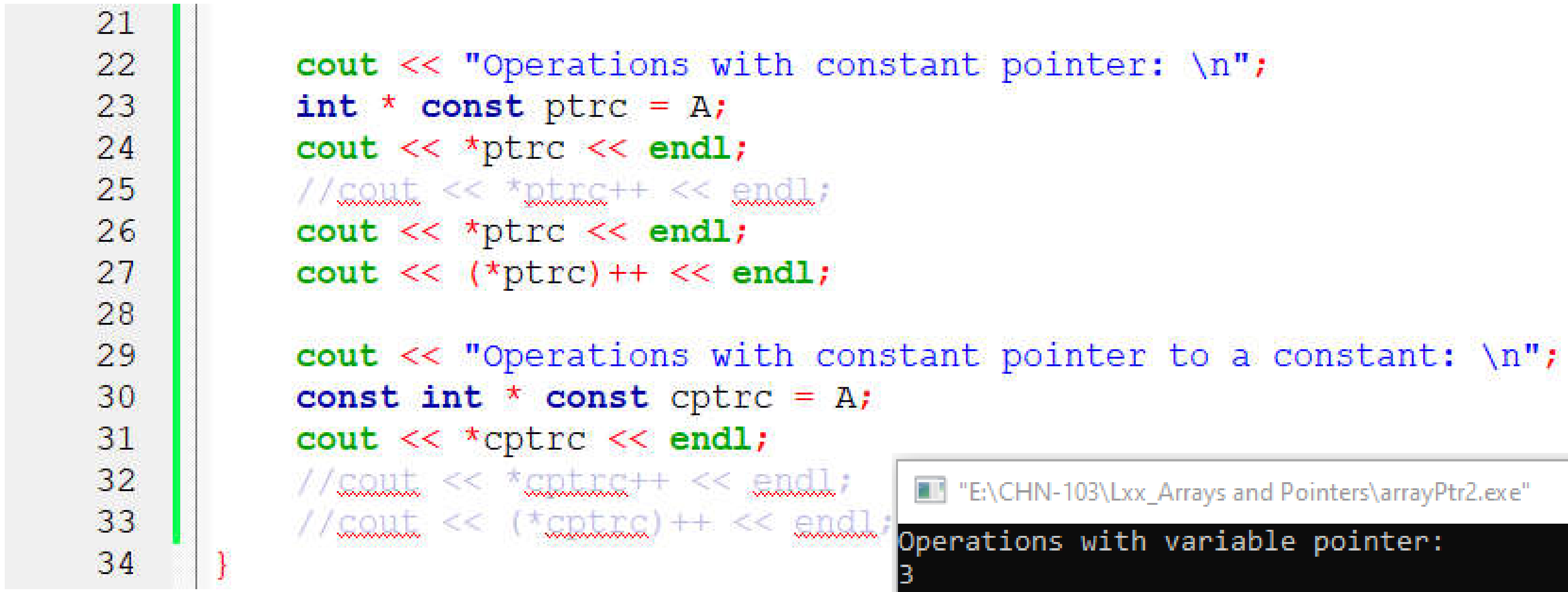
A generic pointer void* can be assigned any pointer, however it can not be dereferenced. A generic pointer can be cast into any other pointer type.

Arrays and Pointers

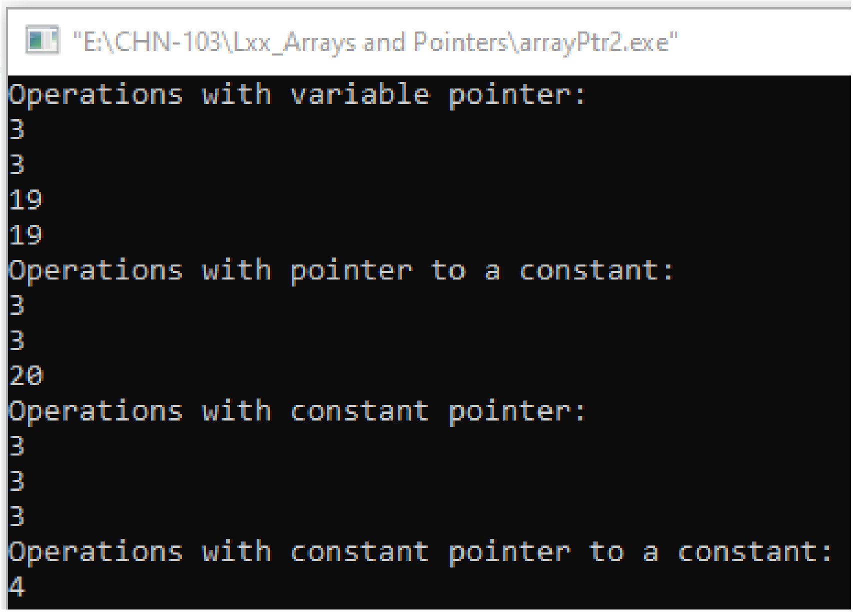


```cpp
#include <iostream>
using namespace std;

int main(){

    int A[] = {3, 19, 5, -3, 1, 0, 56, 12};
    int * ptr;
    ptr = A;
    cout << "Operations with variable pointer: \n";
    cout << *ptr << endl;
    cout << *ptr++ << endl;
    cout << *ptr << endl;
    cout << (*ptr)++ << endl;

    cout << "Operations with pointer to a constant: \n";
    const int * cptr = A;
    cout << *cptr << endl;
    cout << *cptr++ << endl;
    cout << *cptr << endl;
    //cout << (*cptr)++ << endl;
```

Arrays and Pointers



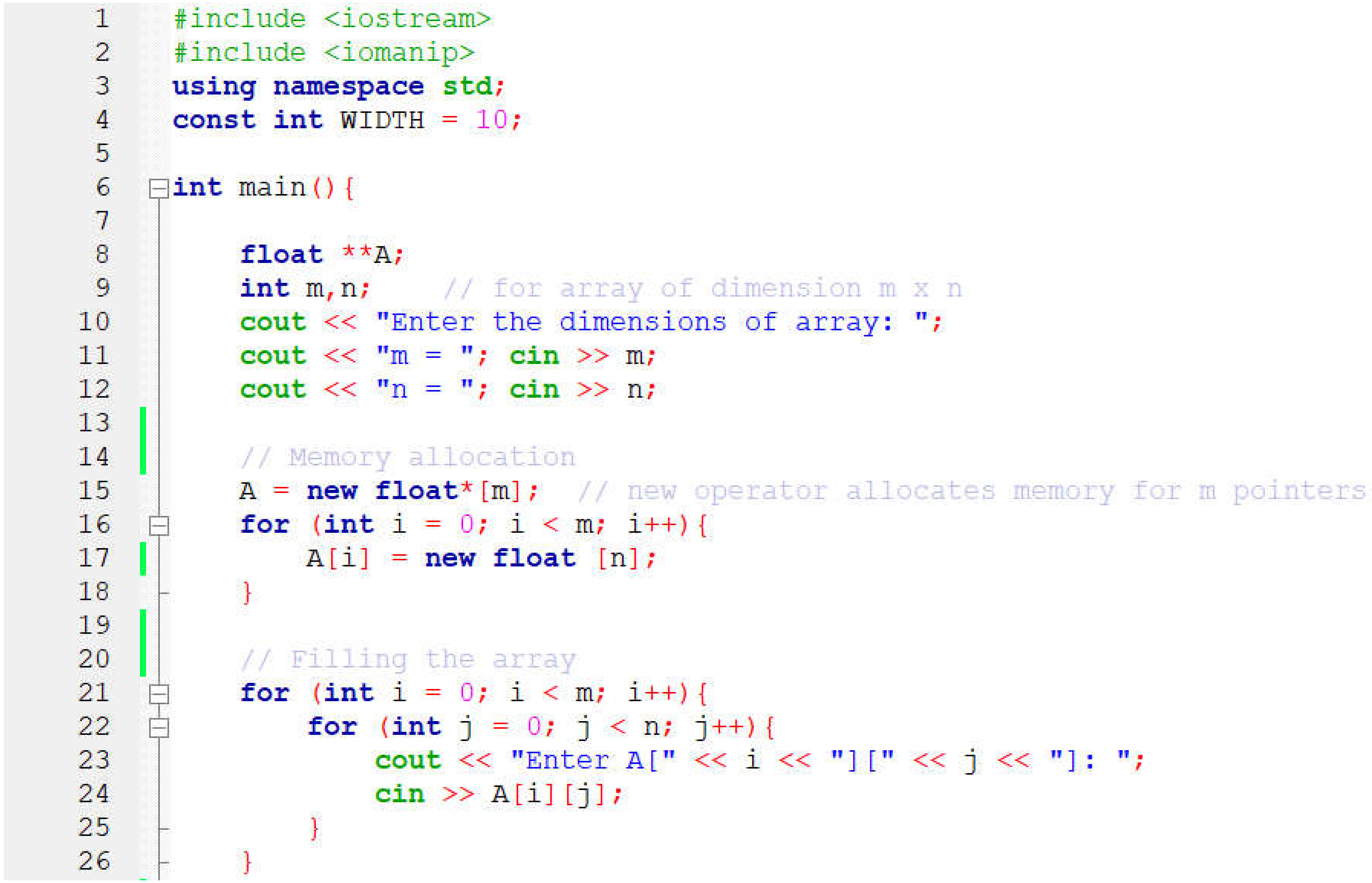
```cpp

    cout << "Operations with constant pointer: \n";
    int * const ptrc = A;
    cout << *ptrc << endl;
    //cout << *ptrc++ << endl;
    cout << *ptrc << endl;
    cout << (*ptrc)++ << endl;

    cout << "Operations with constant pointer to a constant: \n";
    const int * const cptrc = A;
    cout << *cptrc << endl;
    //cout << *cptrc++ << endl;
    //cout << (*cptrc)++ << endl;
}
```



"E:\CHN-103\Lxx_Arrays and Pointers\arrayPtr2.exe"

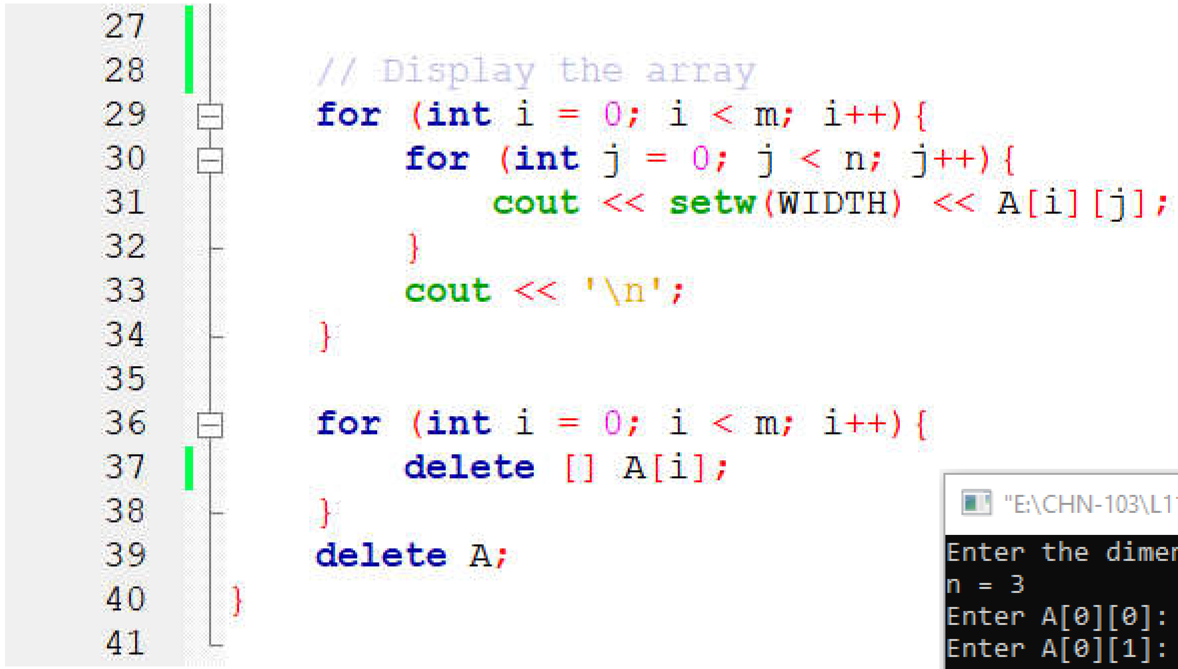
```
Operations with variable pointer:
3
3
19
19
Operations with pointer to a constant:
3
3
20
Operations with constant pointer:
3
3
3
Operations with constant pointer to a constant:
4
```

Arrays and Pointers

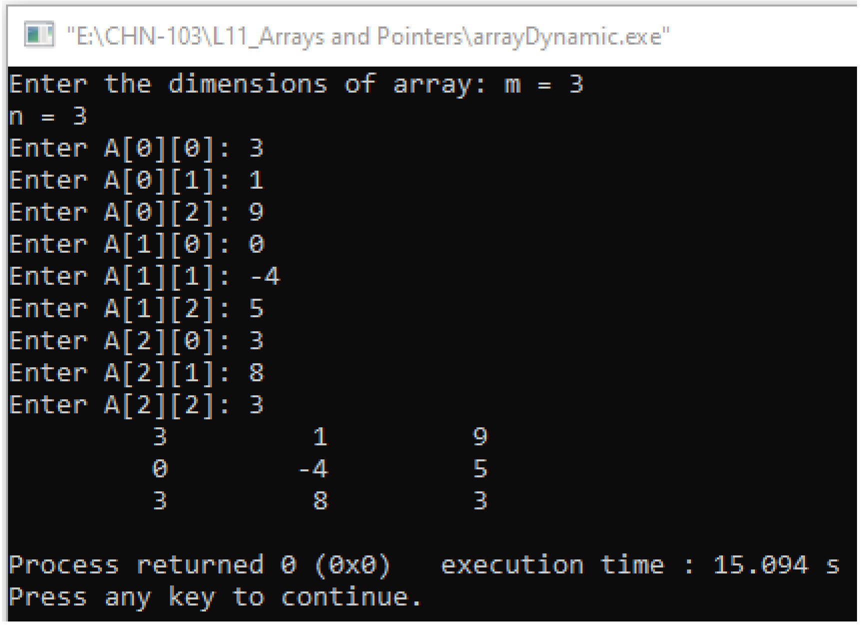


```cpp
#include <iostream>
#include <iomanip>
using namespace std;
const int WIDTH = 10;

int main(){

    float **A;
    int m,n;     // for array of dimension m x n
    cout << "Enter the dimensions of array: ";
    cout << "m = "; cin >> m;
    cout << "n = "; cin >> n;

    // Memory allocation
    A = new float*[m];  // new operator allocates memory for m pointers
    for (int i = 0; i < m; i++){
        A[i] = new float [n];
    }

    // Filling the array
    for (int i = 0; i < m; i++){
        for (int j = 0; j < n; j++){
            cout << "Enter A[" << i << "][" << j << "]: ";
            cin >> A[i][j];
        }
    }
```

Array and Pointers



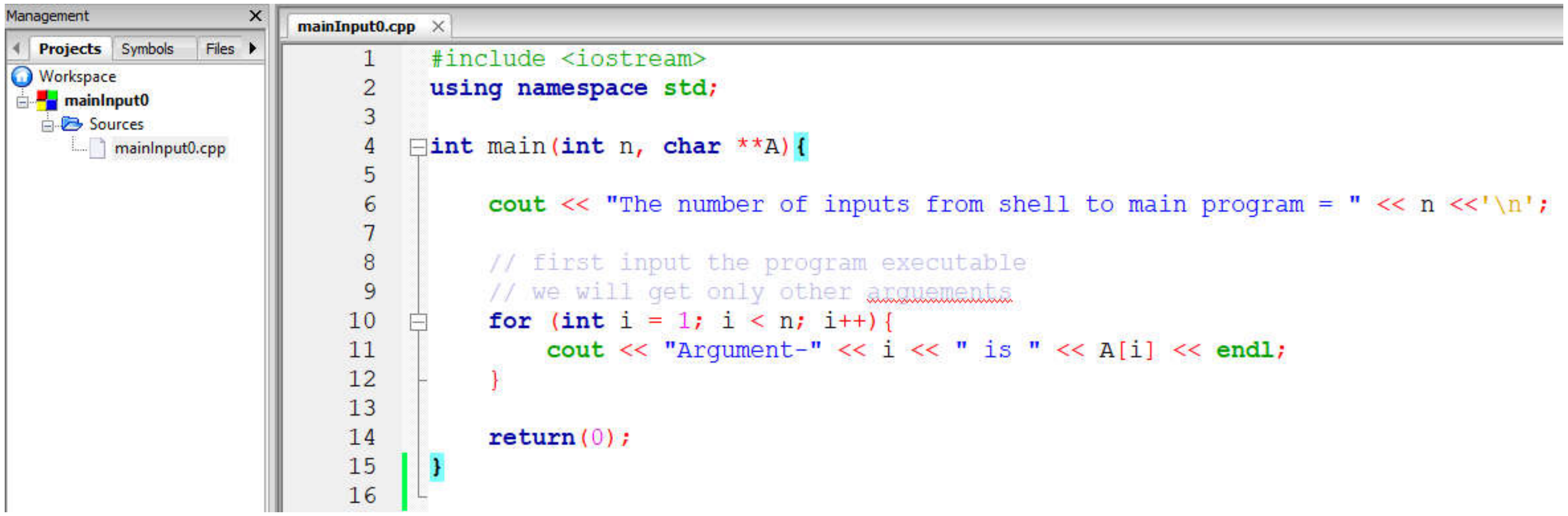
```
    // Display the array
    for (int i = 0; i < m; i++){
        for (int j = 0; j < n; j++){
            cout << setw(WIDTH) << A[i][j];
        }
        cout << '\n';
    }

    for (int i = 0; i < m; i++){
        delete [] A[i];
    }
    delete A;
}
```



"E:\CHN-103\L11_Arrays and Pointers\arrayDynamic.exe"

```
Enter the dimensions of array: m = 3
n = 3
Enter A[0][0]: 3
Enter A[0][1]: 1
Enter A[0][2]: 9
Enter A[1][0]: 0
Enter A[1][1]: -4
Enter A[1][2]: 5
Enter A[2][0]: 3
Enter A[2][1]: 8
Enter A[2][2]: 3
         3         1         9
         0        -4         5
         3         8         3

Process returned 0 (0x0)   execution time : 15.094 s
Press any key to continue.
```

main() function



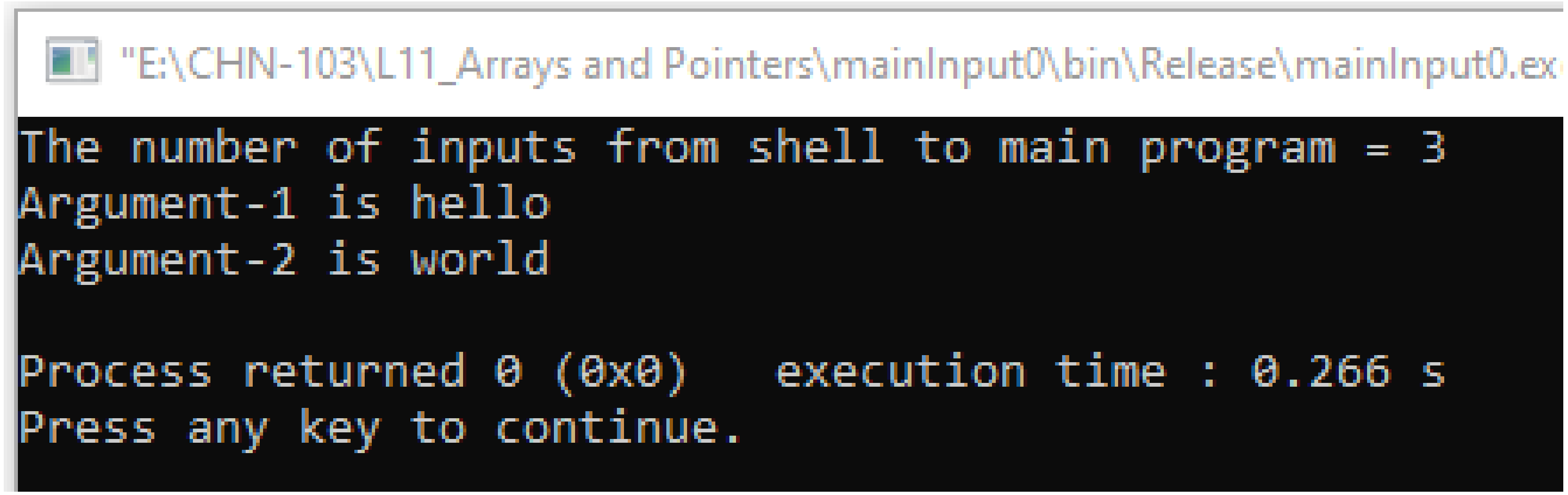
Management

Projects Symbols Files

Workspace
mainInput0
Sources
mainInput0.cpp

mainInput0.cpp

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

int main(int n, char **A){

    cout << "The number of inputs from shell to main program = " << n <<'\n';

    // first input the program executable
    // we will get only other arguements
    for (int i = 1; i < n; i++){
        cout << "Argument-" << i << " is " << A[i] << endl;
    }

    return(0);
}

```



"E:\CHN-103\L11_Arrays and Pointers\mainInput0\bin\Release\mainInput0.ex

```
The number of inputs from shell to main program = 3
Argument-1 is hello
Argument-2 is world

Process returned 0 (0x0)   execution time : 0.266 s
Press any key to continue.
```