Fundamentals of Computer Programming

CS-110

Course Instructor:

Dr. Momina Moetesum

Dynamic Memory Allocation and Dynamic Arrays

Week 13-b

Learning Objectives

01

To understand dynamic memory allocation

02

To use new and delete operators

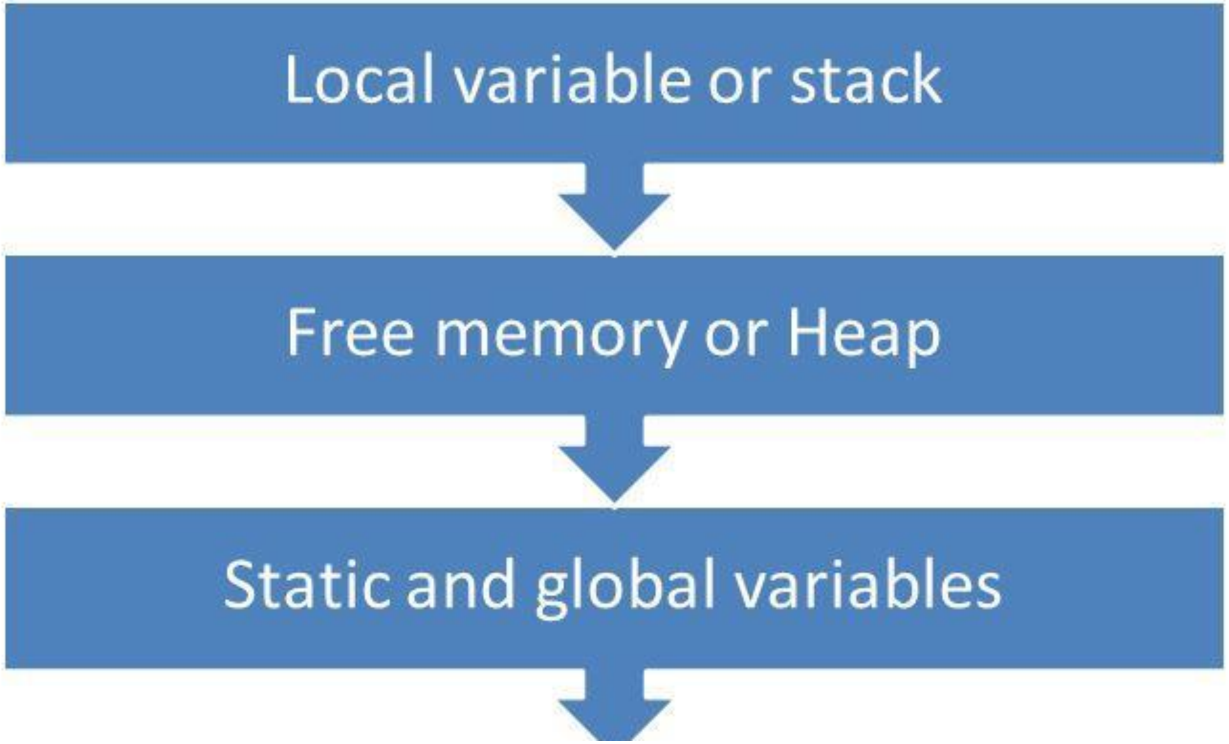
03

To understand how to declare and use dynamic arrays

Memory Allocation Process

Memory allocation process associated with a C program.

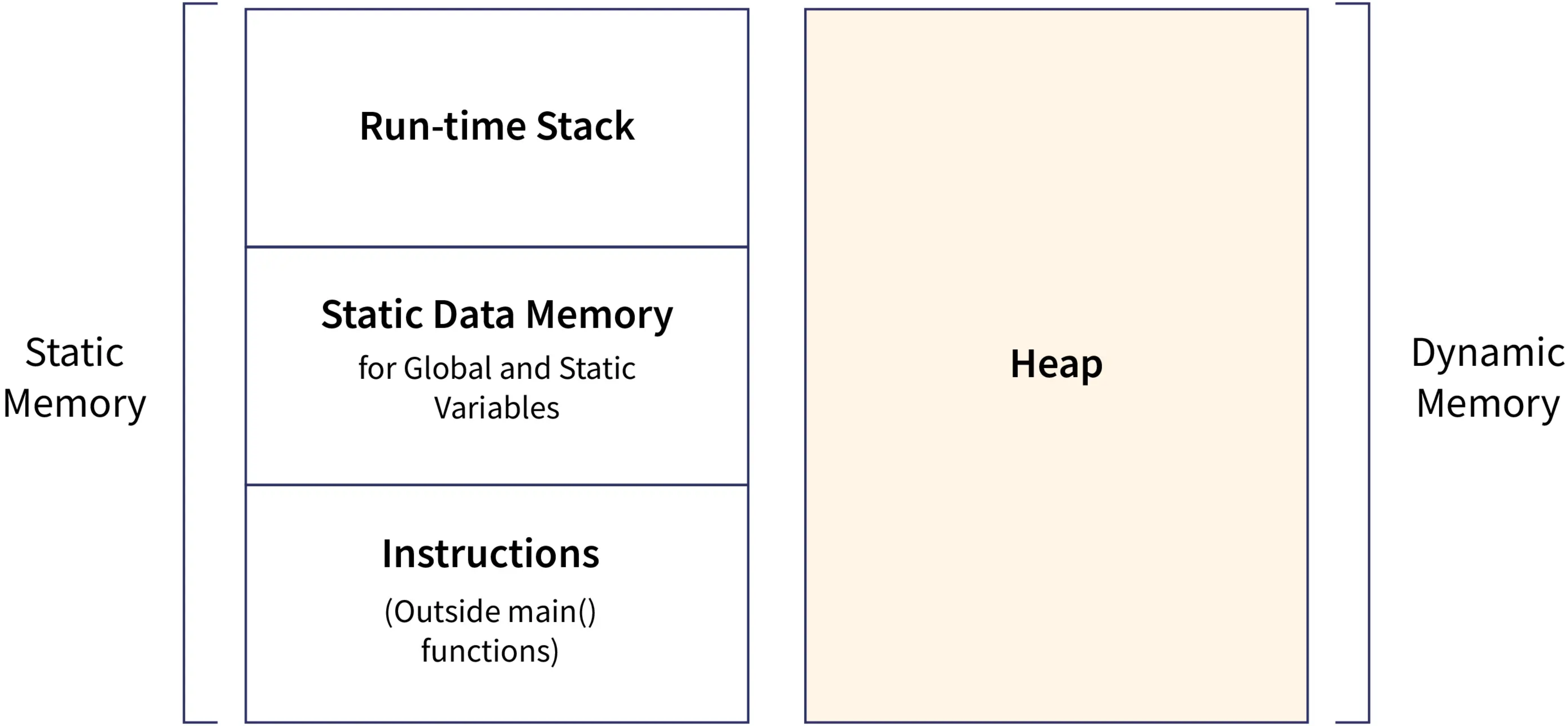
The RAM can divided into 4 areas.



Local variable or stack

Free memory or Heap

Static and global variables



Static Memory

Run-time Stack

Static Data Memory

for Global and Static Variables

Instructions

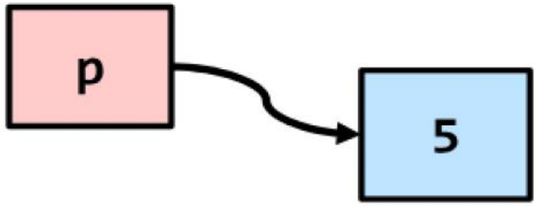
(Outside main() functions)

Heap

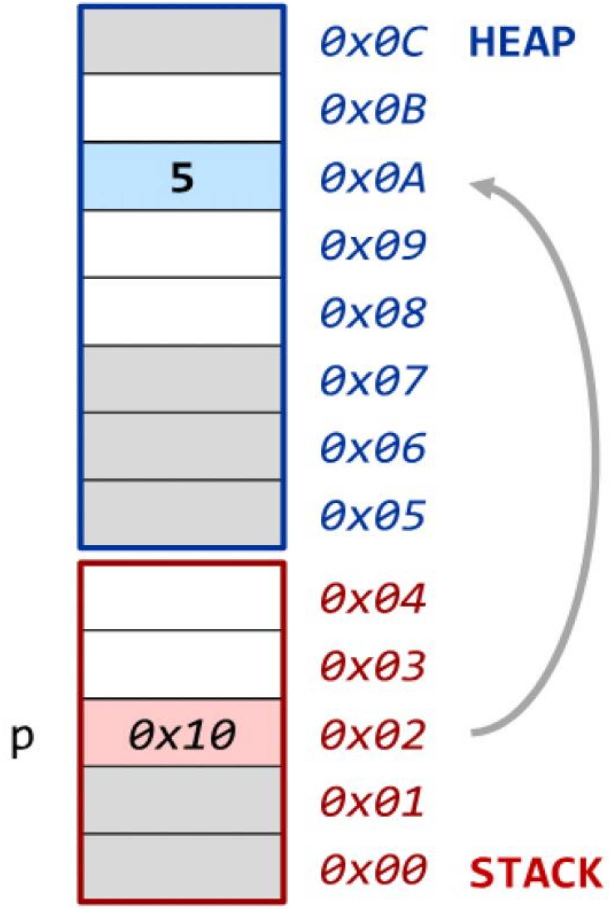
Dynamic Memory

Manual Heap Allocation

```
int* p = new int{5};
…
delete p;
```

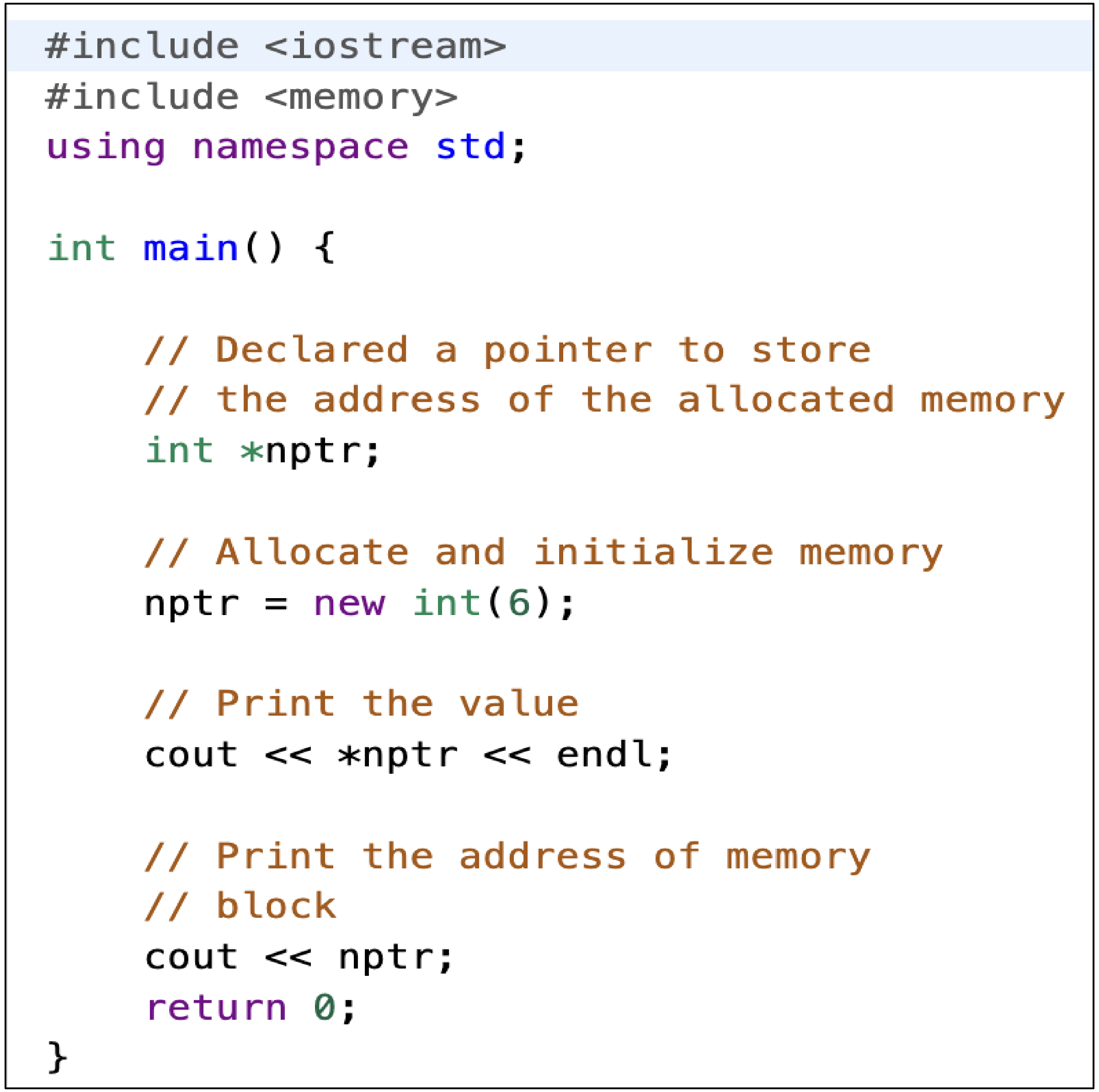


p is said to be
an ***owning*** *raw pointer*



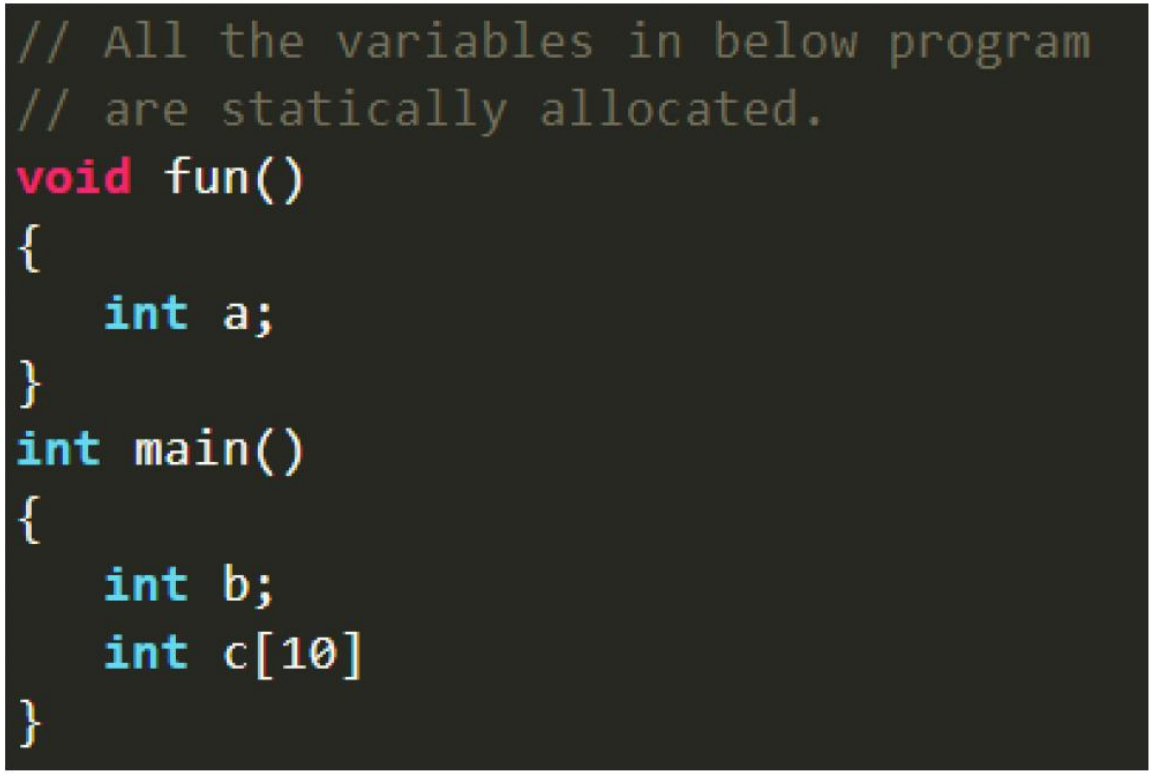
New and Delete Operators

https://www.geeksforgeeks.org/cpp/new-and-delete-operators-in-cpp-for-dynamic-memory/

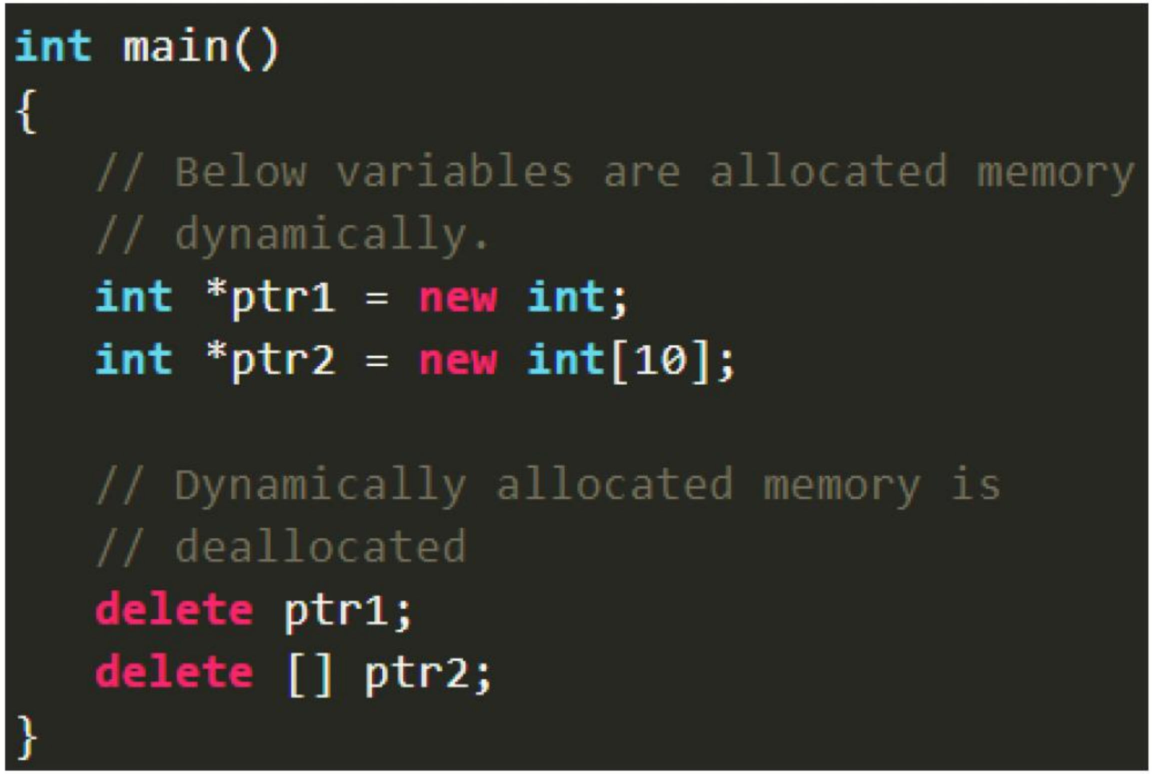


```cpp
#include <iostream>
#include <memory>
using namespace std;

int main() {

    // Declared a pointer to store
    // the address of the allocated memory
    int *nptr;

    // Allocate and initialize memory
    nptr = new int(6);

    // Print the value
    cout << *nptr << endl;

    // Print the address of memory
    // block
    cout << nptr;
    return 0;
}
```

Static vs Dynamic Memory



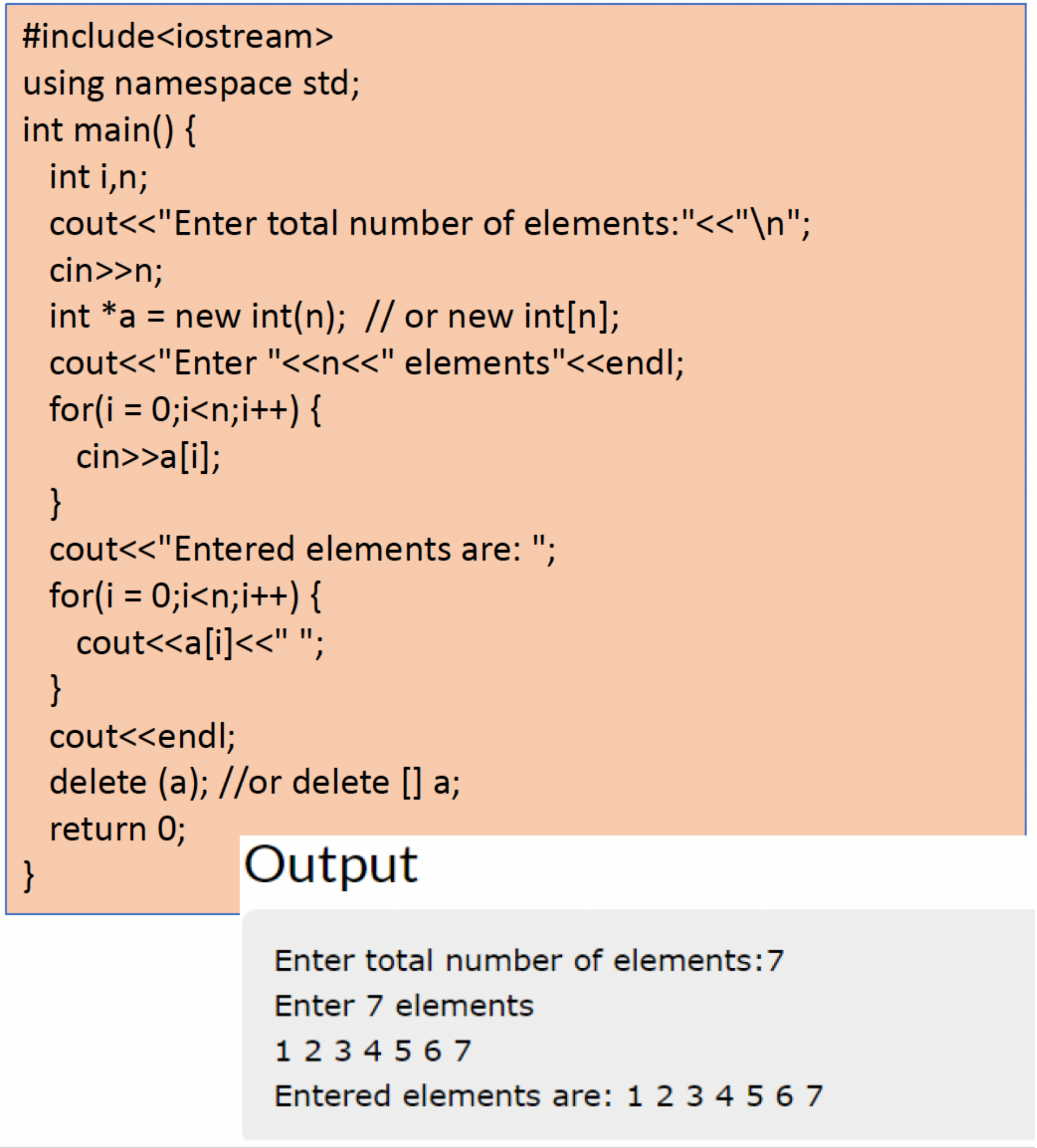
```
// All the variables in below program
// are statically allocated.
void fun()
{
    int a;
}
int main()
{
    int b;
    int c[10]
}
```



```
int main()
{
    // Below variables are allocated memory
    // dynamically.
    int *ptr1 = new int;
    int *ptr2 = new int[10];

    // Dynamically allocated memory is
    // deallocated
    delete ptr1;
    delete [] ptr2;
}
```

Dynamic Arrays

- In C++, a dynamic array can be created using new keyword and can be deleted it by using delete keyword.
- In this program, memory is allocated by declaring, int *a=new int(n), using new keyword. The occupied memory can be retrieved by calling delete (a).



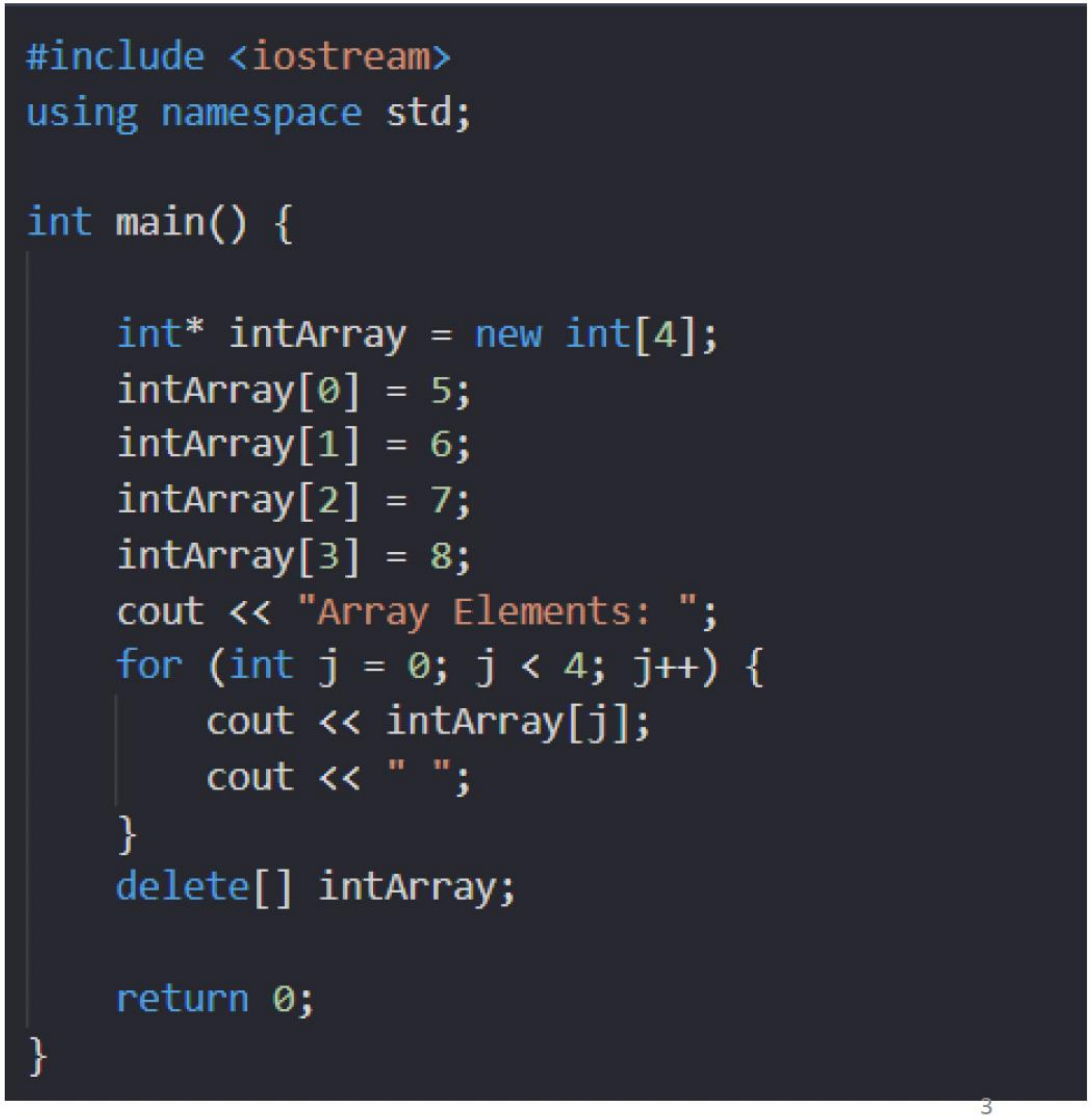
```
#include<iostream>
using namespace std;
int main() {
  int i,n;
  cout<<"Enter total number of elements:"<<"\n";
  cin>>n;
  int *a = new int(n);  // or new int[n];
  cout<<"Enter "<<n<<" elements"<<endl;
  for(i = 0;i<n;i++) {
    cin>>a[i];
  }
  cout<<"Entered elements are: ";
  for(i = 0;i<n;i++) {
    cout<<a[i]<<" ";
  }
  cout<<endl;
  delete (a); //or delete [] a;
  return 0;
}
```

Output

```
Enter total number of elements:7
Enter 7 elements
1 2 3 4 5 6 7
Entered elements are: 1 2 3 4 5 6 7
```

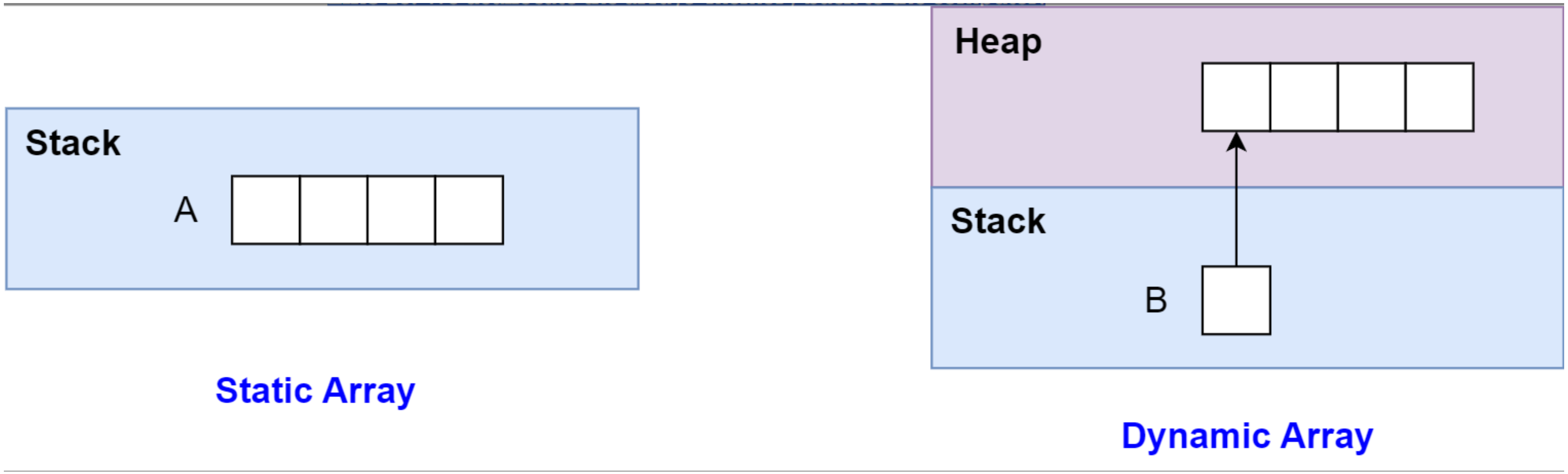
Example

- ***Line 6:*** *We define a dynamic integer array intArray with size 4 using the new operator.*
- ***Lines 7–10:*** *We define values for the array elements.*
- ***Lines 12–15:*** *We use a loop to print the array's elements.*
- ***Line 16:*** *We deallocate the arrays’ memory back to the computer.*



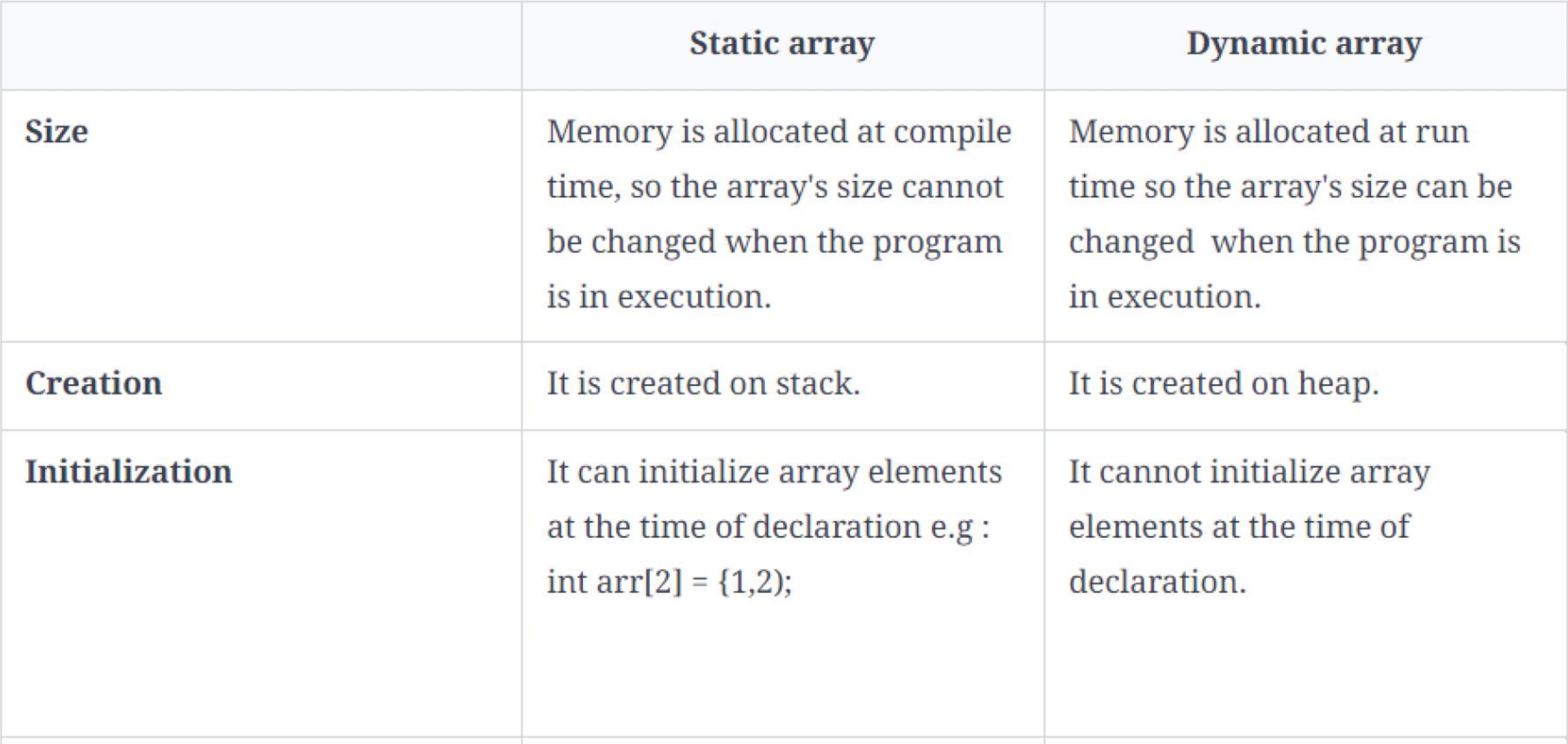
```
#include <iostream>
using namespace std;

int main() {

    int* intArray = new int[4];
    intArray[0] = 5;
    intArray[1] = 6;
    intArray[2] = 7;
    intArray[3] = 8;
    cout << "Array Elements: ";
    for (int j = 0; j < 4; j++) {
        cout << intArray[j];
        cout << " ";
    }
    delete[] intArray;

    return 0;
}
```

Static vs Dynamic Arrays

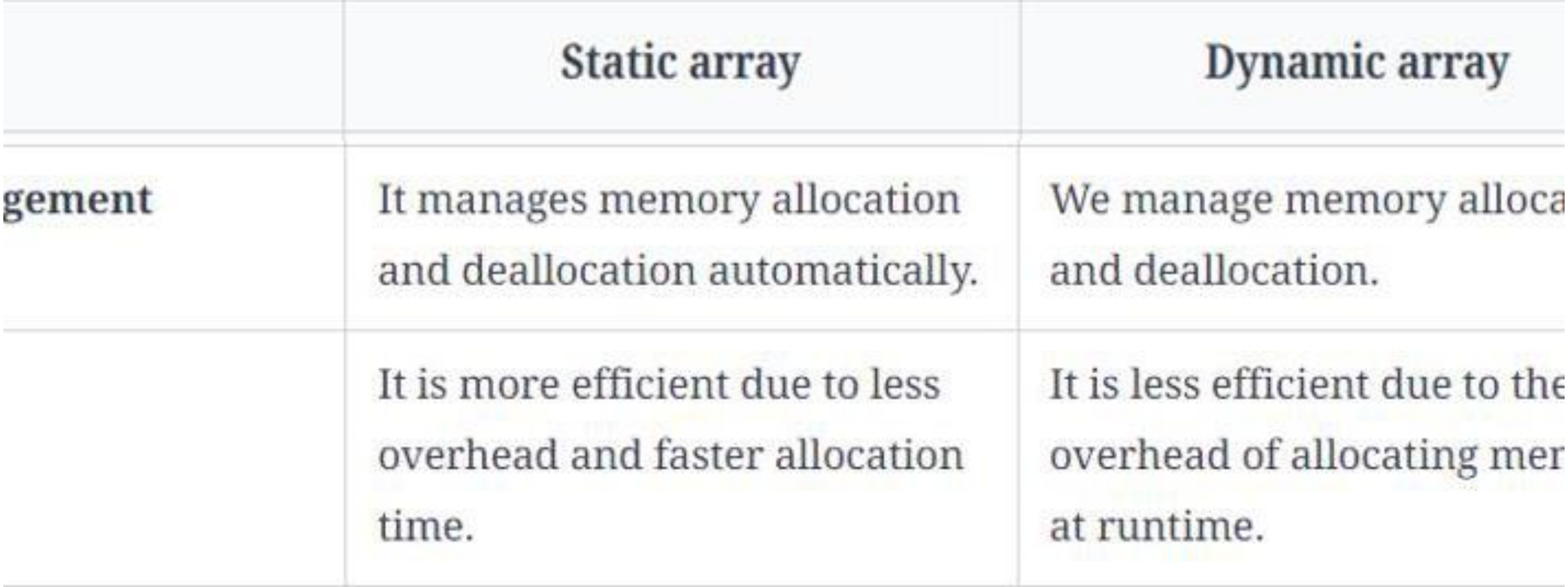


Static Array

Dynamic Array

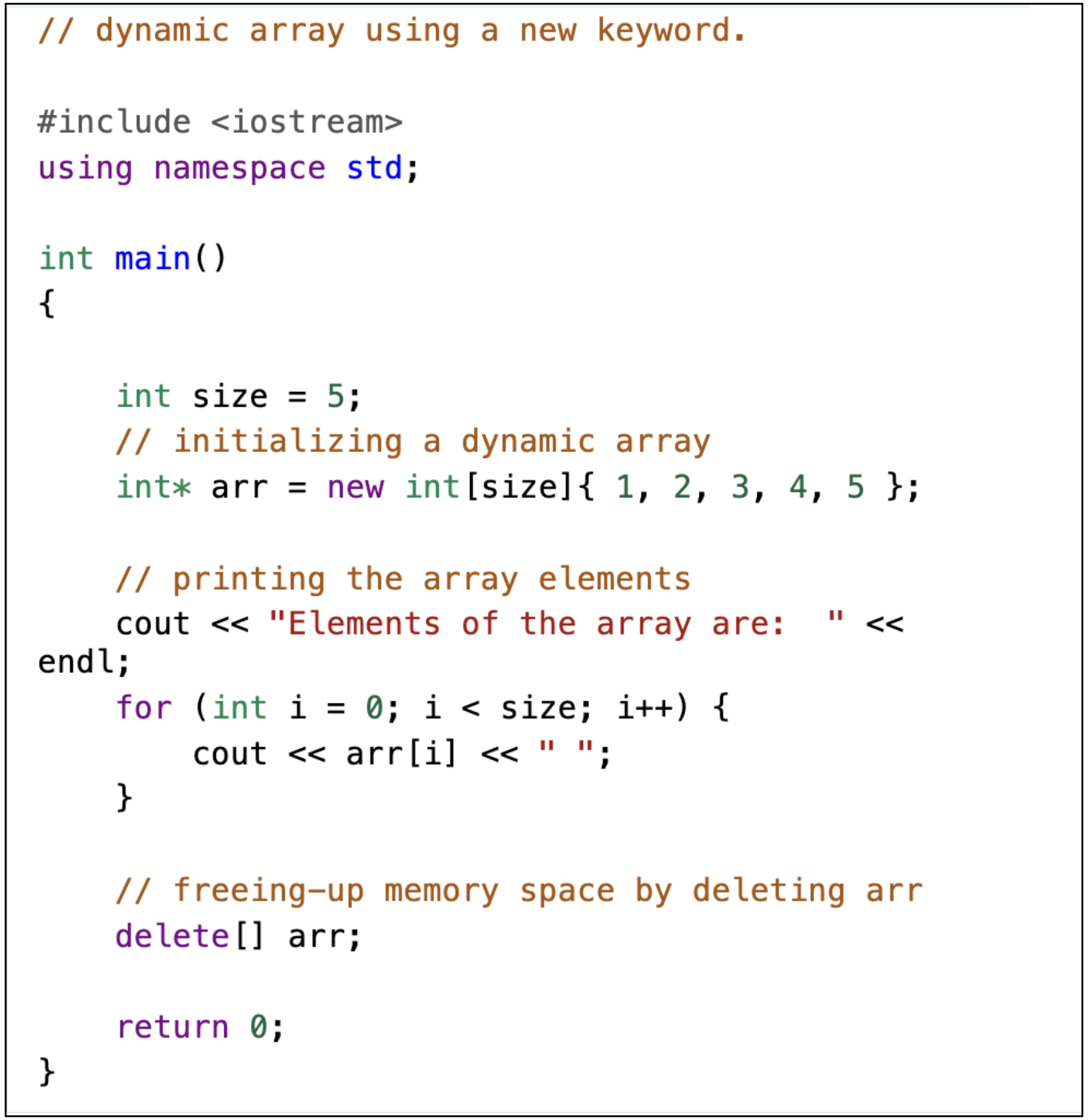


	Static array	Dynamic array
Size	Memory is allocated at compile time, so the array's size cannot be changed when the program is in execution.	Memory is allocated at run time so the array's size can be changed when the program is in execution.
Creation	It is created on stack.	It is created on heap.
Initialization	It can initialize array elements at the time of declaration e.g : int arr[2] = {1,2);	It cannot initialize array elements at the time of declaration.



	Static array	Dynamic array
gement	It manages memory allocation and deallocation automatically.	We manage memory alloca and deallocation.
	It is more efficient due to less overhead and faster allocation time.	It is less efficient due to the overhead of allocating mer at runtime.

Initializing a Dynamic Array



```
// dynamic array using a new keyword.

#include <iostream>
using namespace std;

int main()
{

    int size = 5;
    // initializing a dynamic array
    int* arr = new int[size]{ 1, 2, 3, 4, 5 };

    // printing the array elements
    cout << "Elements of the array are:  " <<
endl;
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }

    // freeing-up memory space by deleting arr
    delete[] arr;

    return 0;
}
```

https://www.geeksforgeeks.org/cpp/initialize-a-dynamic-array-in-cpp/

Acknowledgment

- Content of these slides are taken from:
  - https://www.geeksforgeeks.org/
  - https://www.tutorialspoint.com/
  - https://www.programiz.com/
  - https://www.w3schools.com/