C++ BASICS RELEVANT TO CSI 228

FARIHA TABASSUM ISLAM

BASICS

- C++ language extends the C programming language with additional features such as type checking, object-oriented programming, exception handling etc.
  - C++ was developed by Bjarne Stroustrup in 1979.
  - File extension .cpp

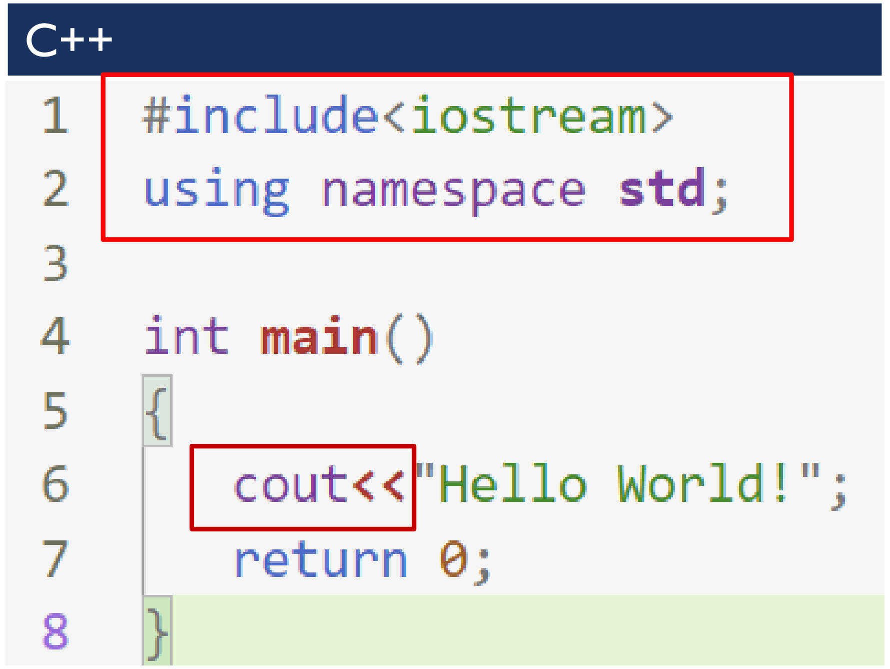
- Why C++ in this course?
  - The Standard Template Library (STL) of C++ provides useful codes
  - STL is a set of C++ template classes to provide common programming data structures and functions such as lists, stacks, arrays, etc. It is a library of container classes, algorithms, and iterators.

SIMILARITIES WITH C

- Variables, Operators
- struct
- Array
- Function
- Pointer
- Strings
- If, if…else-if statement, switch case, for loop, while loop, do-while loop, continue statement, break statement, goto statement
- Recursion

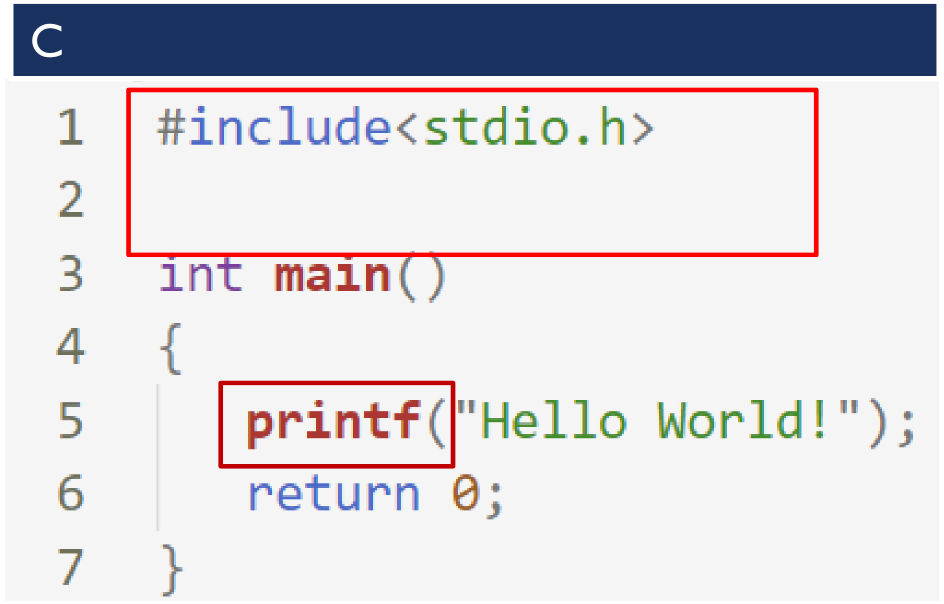
can use the same code as written in c

BASICS


C++

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

int main()
{
    cout<<"Hello World!";
    return 0;
}
```

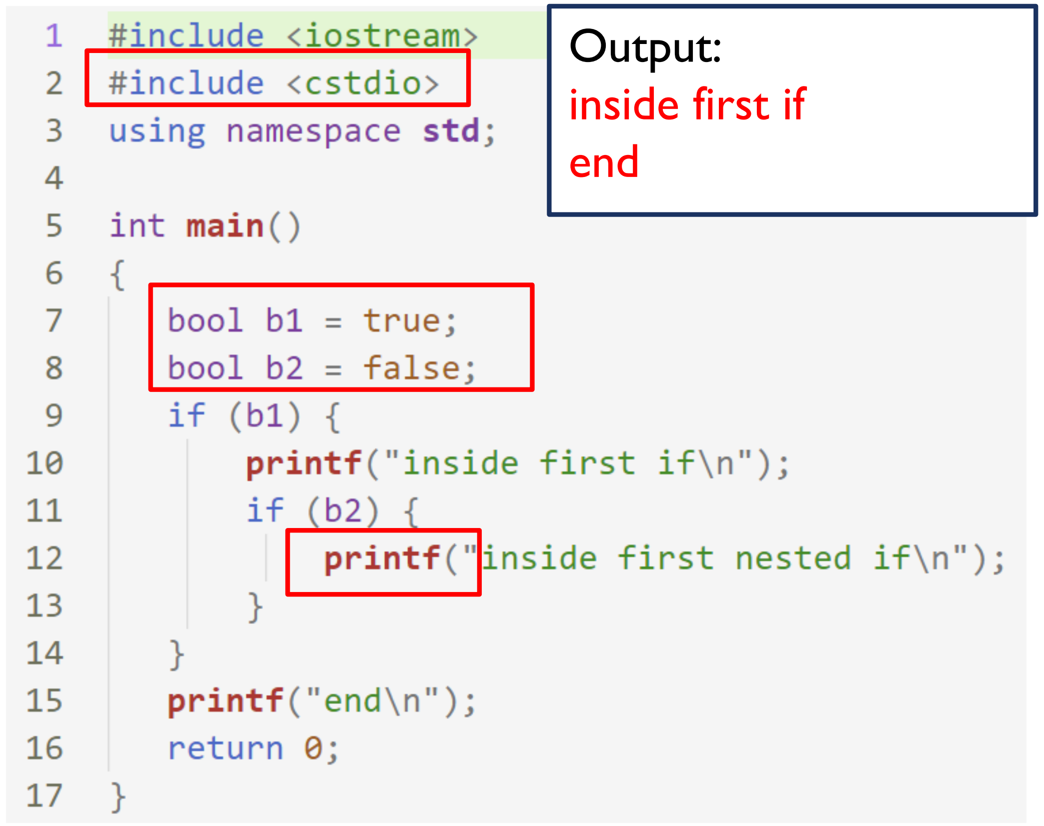
for details explanation: https://beginnersbook.com/2017/08/first-cpp-program/


C

```c
#include<stdio.h>

int main()
{
    printf("Hello World!");
    return 0;
}
```

VARIABLES AND DATA TYPES

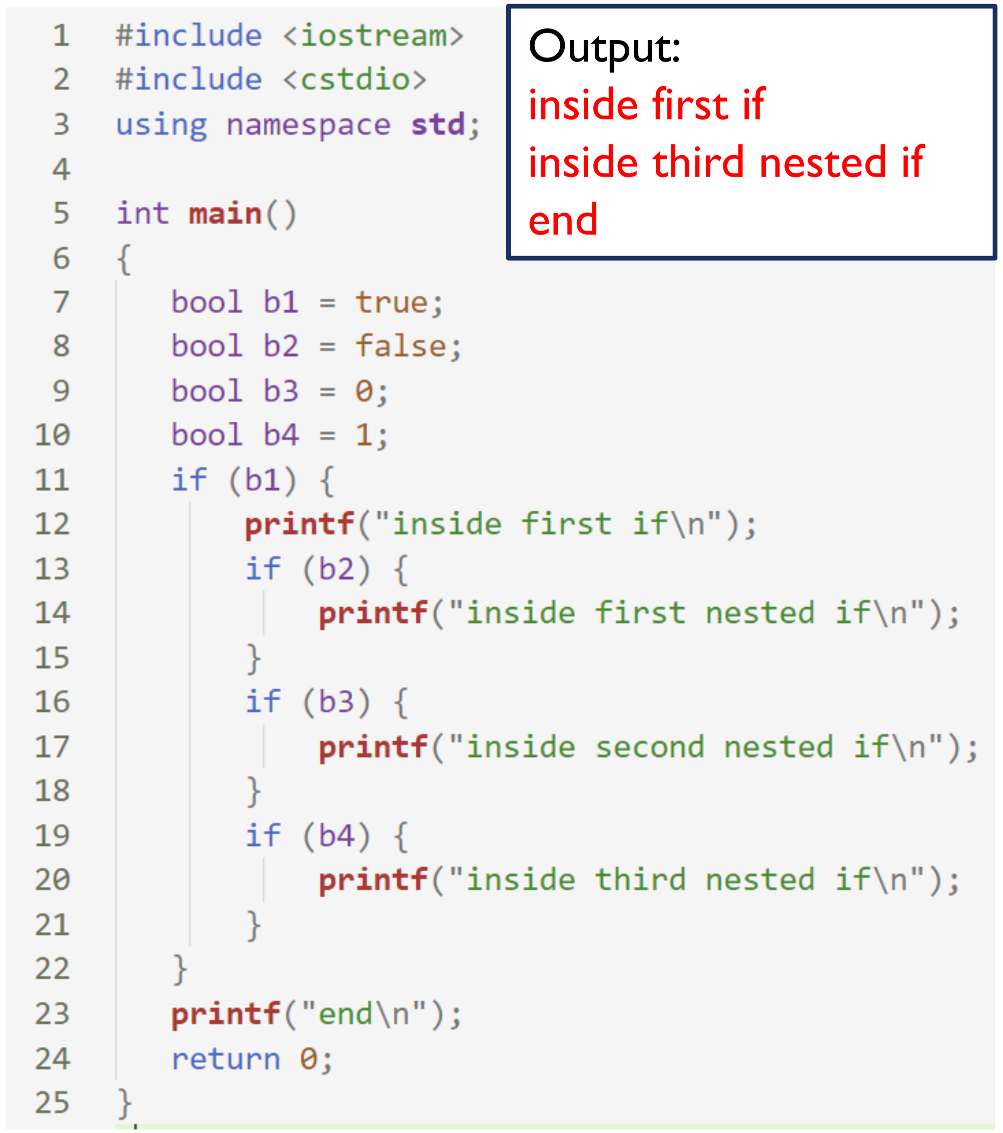
- int
- char
- bool
  - holds Boolean value true or false
- double
- float



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

int main()
{
    bool b1 = true;
    bool b2 = false;
    if (b1) {
        printf("inside first if\n");
        if (b2) {
            printf("inside first nested if\n");
        }
    }
    printf("end\n");
    return 0;
}
```

Output:

```
inside first if
end
```



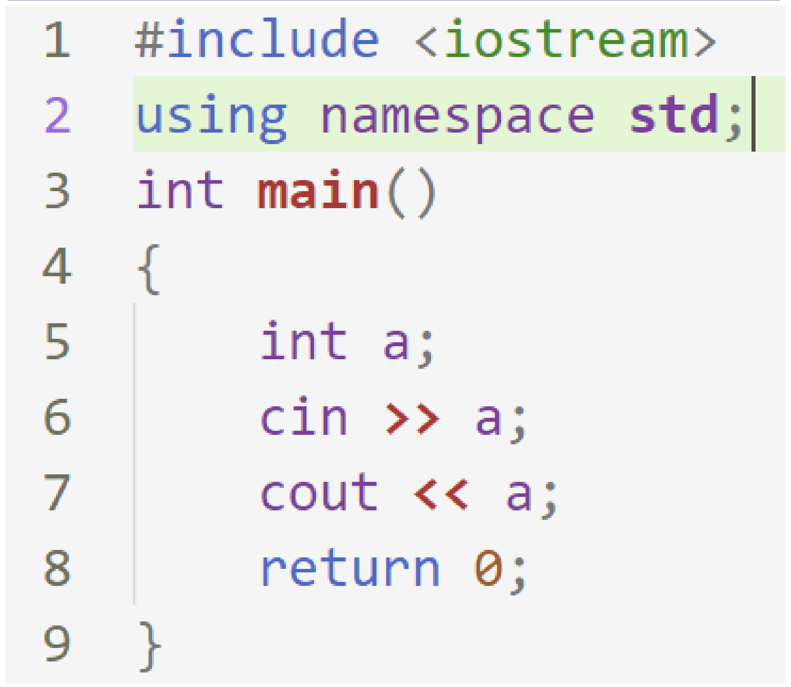
```cpp
#include <iostream>
#include <cstdio>
using namespace std;

int main()
{
    bool b1 = true;
    bool b2 = false;
    bool b3 = 0;
    bool b4 = 1;
    if (b1) {
        printf("inside first if\n");
        if (b2) {
            printf("inside first nested if\n");
        }
        if (b3) {
            printf("inside second nested if\n");
        }
        if (b4) {
            printf("inside third nested if\n");
        }
    }
    printf("end\n");
    return 0;
}
```

Output:

```
inside first if
inside third nested if
end
```

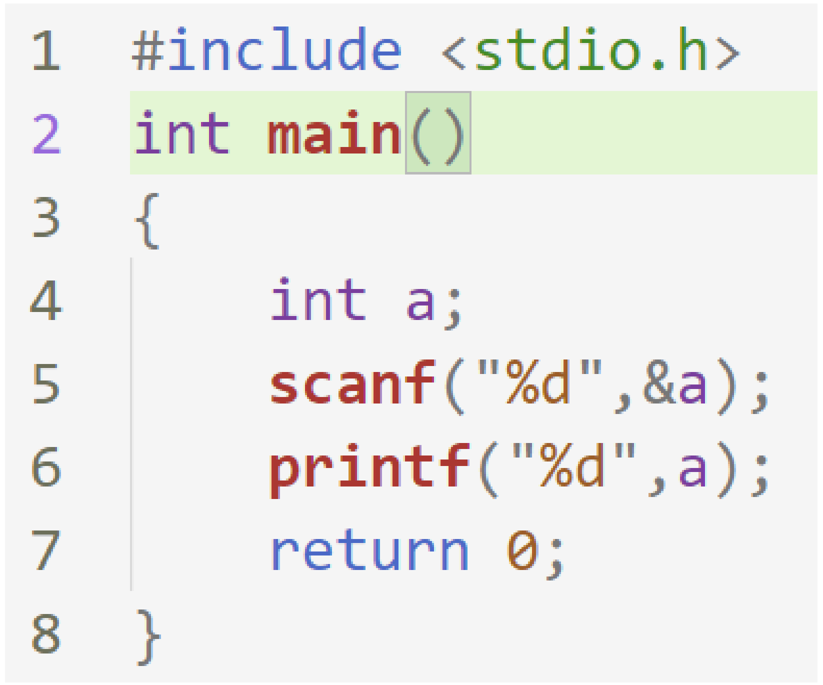
scanf(), printf() EQUIVALENT

C++



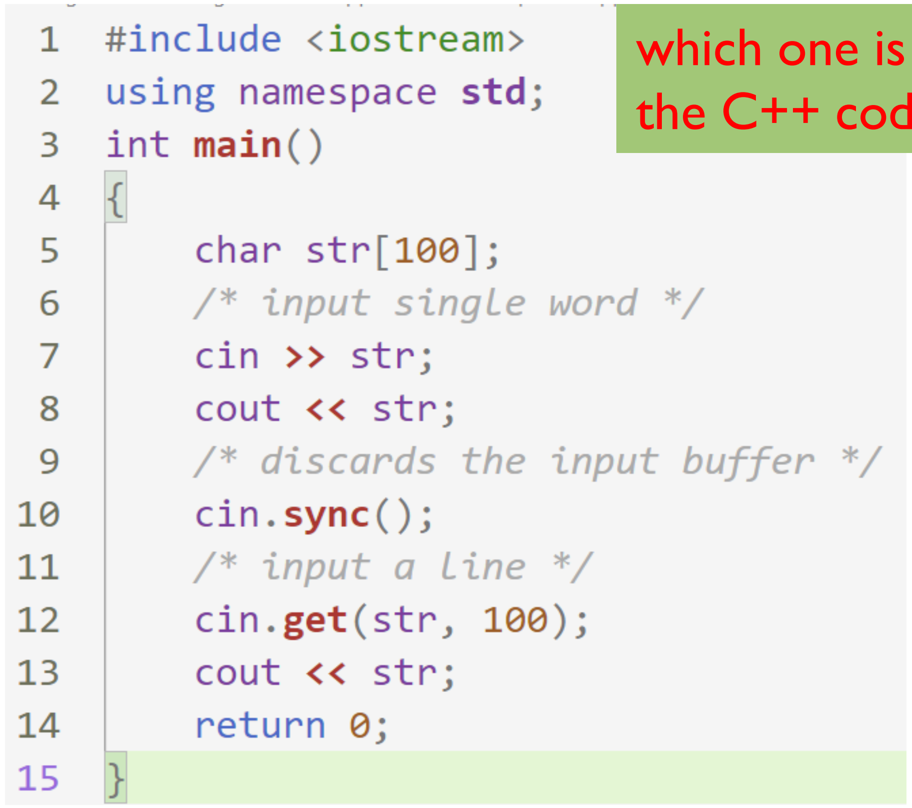
```cpp
#include <iostream>
using namespace std;
int main()
{
    int a;
    cin >> a;
    cout << a;
    return 0;
}
```

C



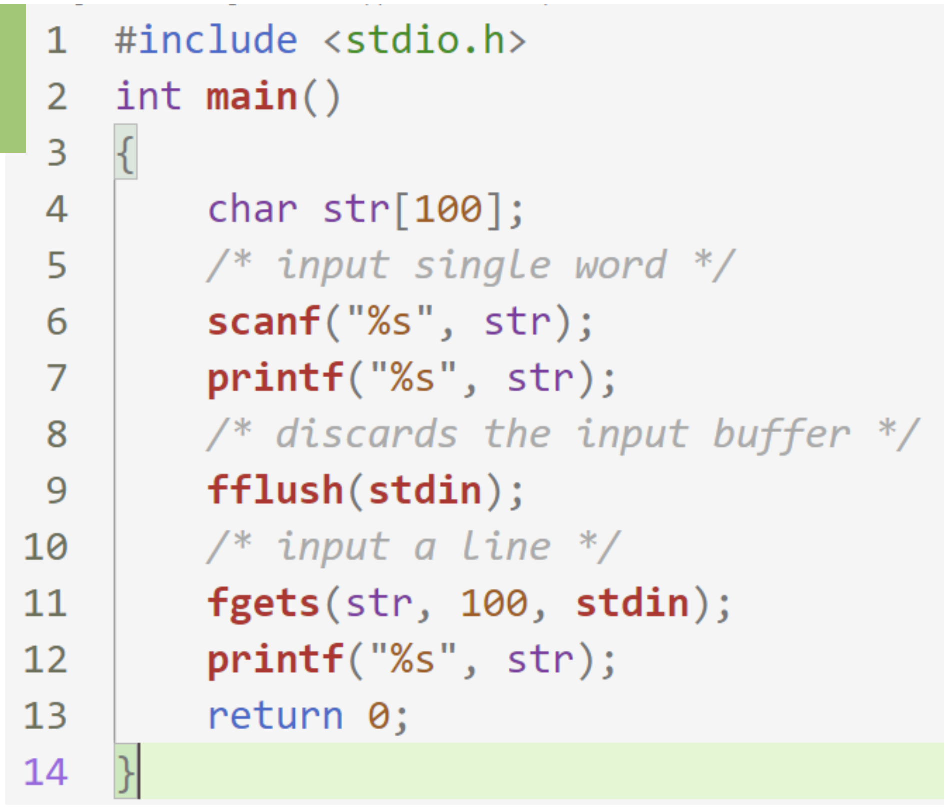
```c
#include <stdio.h>
int main()
{
    int a;
    scanf("%d",&a);
    printf("%d",a);
    return 0;
}
```

scanf(), printf() EQUIVALENT



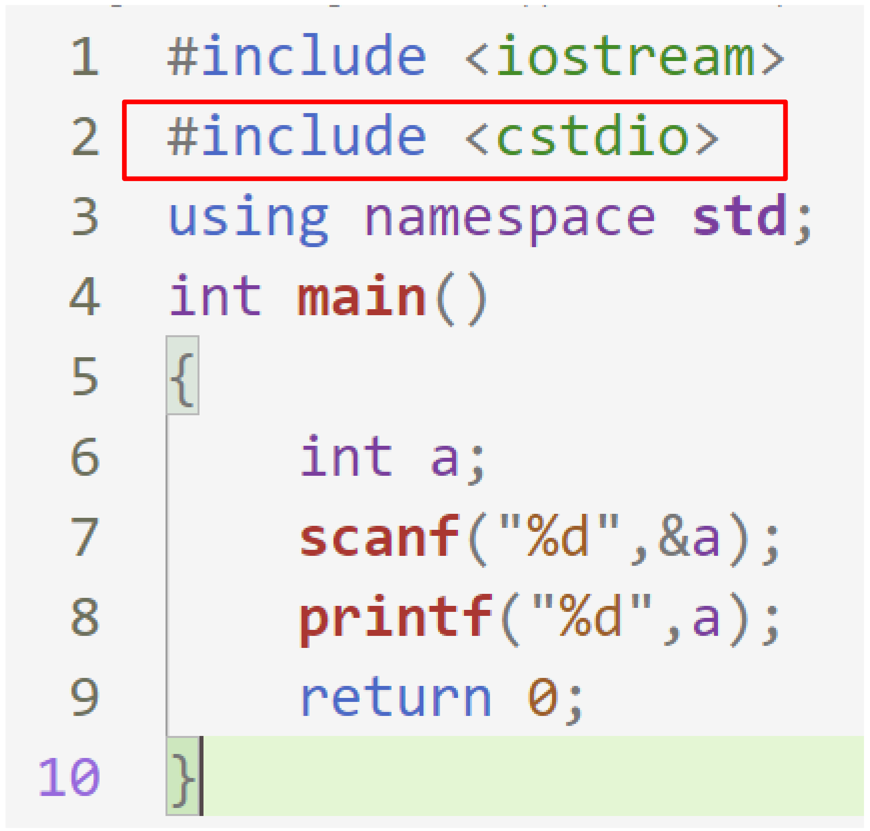
```cpp
#include <iostream>
using namespace std;
int main()
{
    char str[100];
    /* input single word */
    cin >> str;
    cout << str;
    /* discards the input buffer */
    cin.sync();
    /* input a line */
    cin.get(str, 100);
    cout << str;
    return 0;
}
```

which one is the C++ code?



```c
#include <stdio.h>
int main()
{
    char str[100];
    /* input single word */
    scanf("%s", str);
    printf("%s", str);
    /* discards the input buffer */
    fflush(stdin);
    /* input a line */
    fgets(str, 100, stdin);
    printf("%s", str);
    return 0;
}
```

IF YOU PREFER scanf(), printf() OVER cin, cout



```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    int a;
    scanf("%d",&a);
    printf("%d",a);
    return 0;
}
```

STANDARD TEMPLATE LIBRARY (STL)

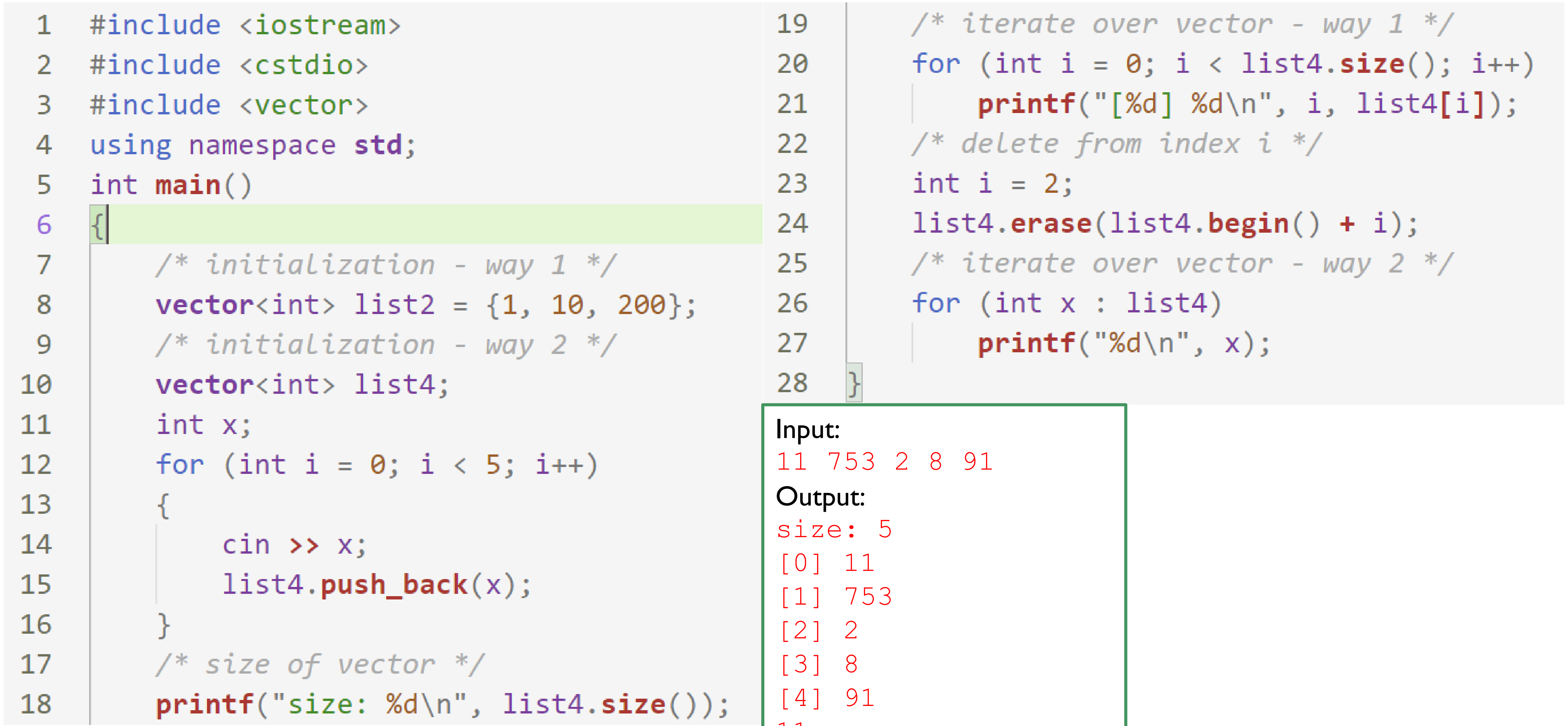
- STL is a set of C++ template classes to provide common programming data structures and functions such as lists, stacks, arrays, etc. It is a library of container classes, algorithms, and iterators.

- We are going to use
  - Sorting
  - vector
  - priority queue

VECTOR

- #include <vector>
- To know more about vectors: https://www.edureka.co/blog/vectors-in-cpp/

Vector



```cpp
#include <iostream>
#include <cstdio>
#include <vector>
using namespace std;
int main()
{
    /* initialization - way 1 */
    vector<int> list2 = {1, 10, 200};
    /* initialization - way 2 */
    vector<int> list4;
    int x;
    for (int i = 0; i < 5; i++)
    {
        cin >> x;
        list4.push_back(x);
    }
    /* size of vector */
    printf("size: %d\n", list4.size());
    /* iterate over vector - way 1 */
    for (int i = 0; i < list4.size(); i++)
        printf("[%d] %d\n", i, list4[i]);
    /* delete from index i */
    int i = 2;
    list4.erase(list4.begin() + i);
    /* iterate over vector - way 2 */
    for (int x : list4)
        printf("%d\n", x);
}
```

Input:

```
11 753 2 8 91
```

Output:

```
size: 5
[0] 11
[1] 753
[2] 2
[3] 8
[4] 91
11
753
8
91
```

Vector

```cpp
#include <bits/stdc++.h>


using namespace std;
int main()
{
    /* initialization - way 1 */
    vector<int> list2 = {1, 10, 200};
    /* initialization - way 2 */
    vector<int> list4;
    int x;
    for (int i = 0; i < 5; i++)
    {
        cin >> x;
        list4.push_back(x);
    }
    /* size of vector */
    printf("size: %d\n", list4.size());
    /* iterate over vector - way 1 */
    for (int i = 0; i < list4.size(); i++)
        printf("[%d] %d\n", i, list4[i]);
    /* delete from index i */
    int i = 2;
    list4.erase(list4.begin() + i);
    /* iterate over vector - way 2 */
    for (int x : list4)
        printf("%d\n", x);
}
```

Input:

```
11 753 2 8 91
```

Output:

```
size: 5
[0] 11
[1] 753
[2] 2
[3] 8
[4] 91
11
753
8
91
```

SORTING

- sort array
- sort vector
- sort structure

SORT Array

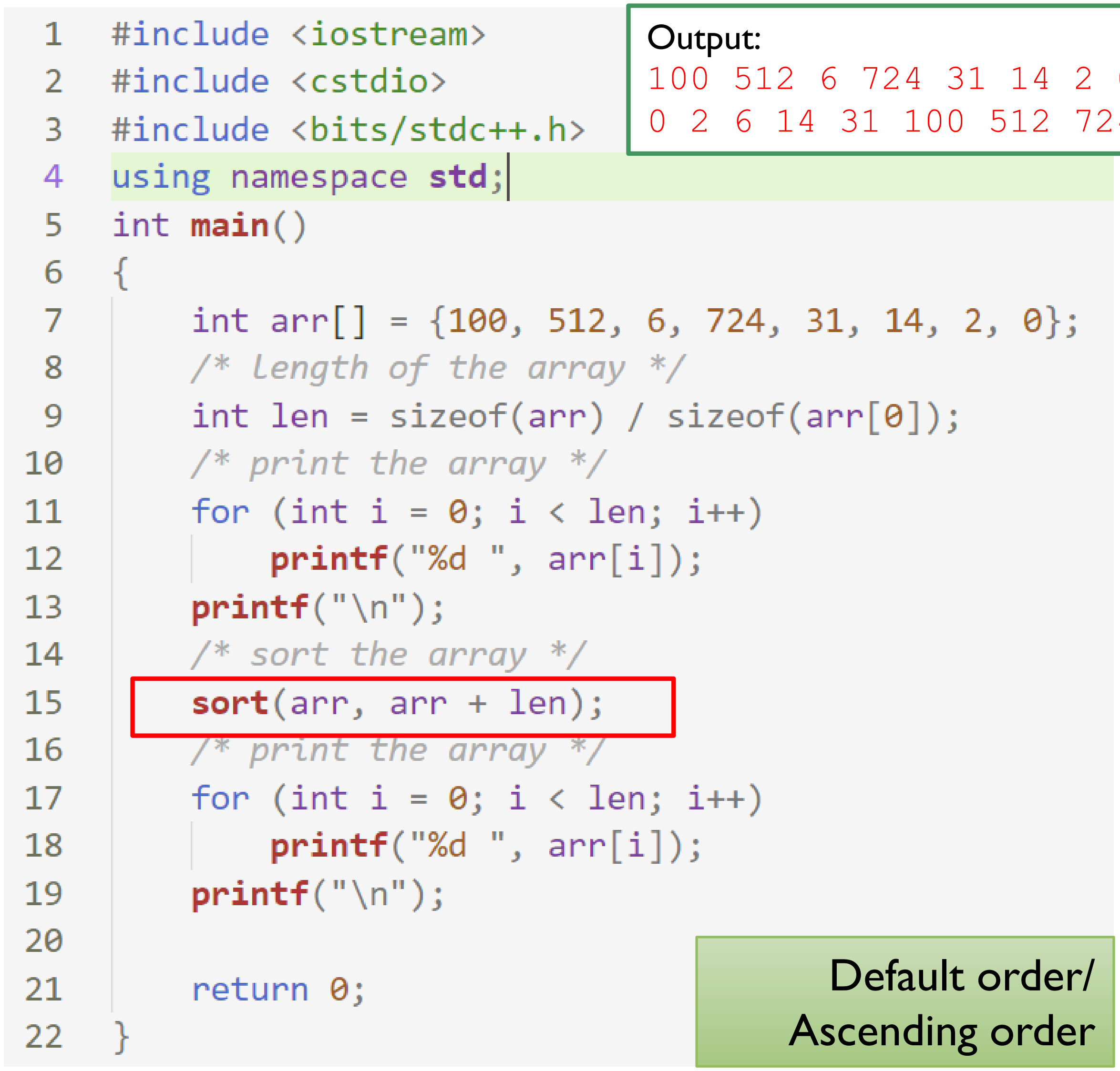
```
#include <iostream>
#include <cstdio>
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int arr[] = {100, 512, 6, 724, 31, 14, 2, 0};
    /* length of the array */
    int len = sizeof(arr) / sizeof(arr[0]);
    /* print the array */
    for (int i = 0; i < len; i++)
        printf("%d ", arr[i]);
    printf("\n");
    /* sort the array */
    sort(arr, arr + len);
    /* print the array */
    for (int i = 0; i < len; i++)
        printf("%d ", arr[i]);
    printf("\n");

    return 0;
}
```

Output:

```
100 512 6 724 31 14 2 0
0 2 6 14 31 100 512 724
```

Default order/
Ascending order

SORT Array

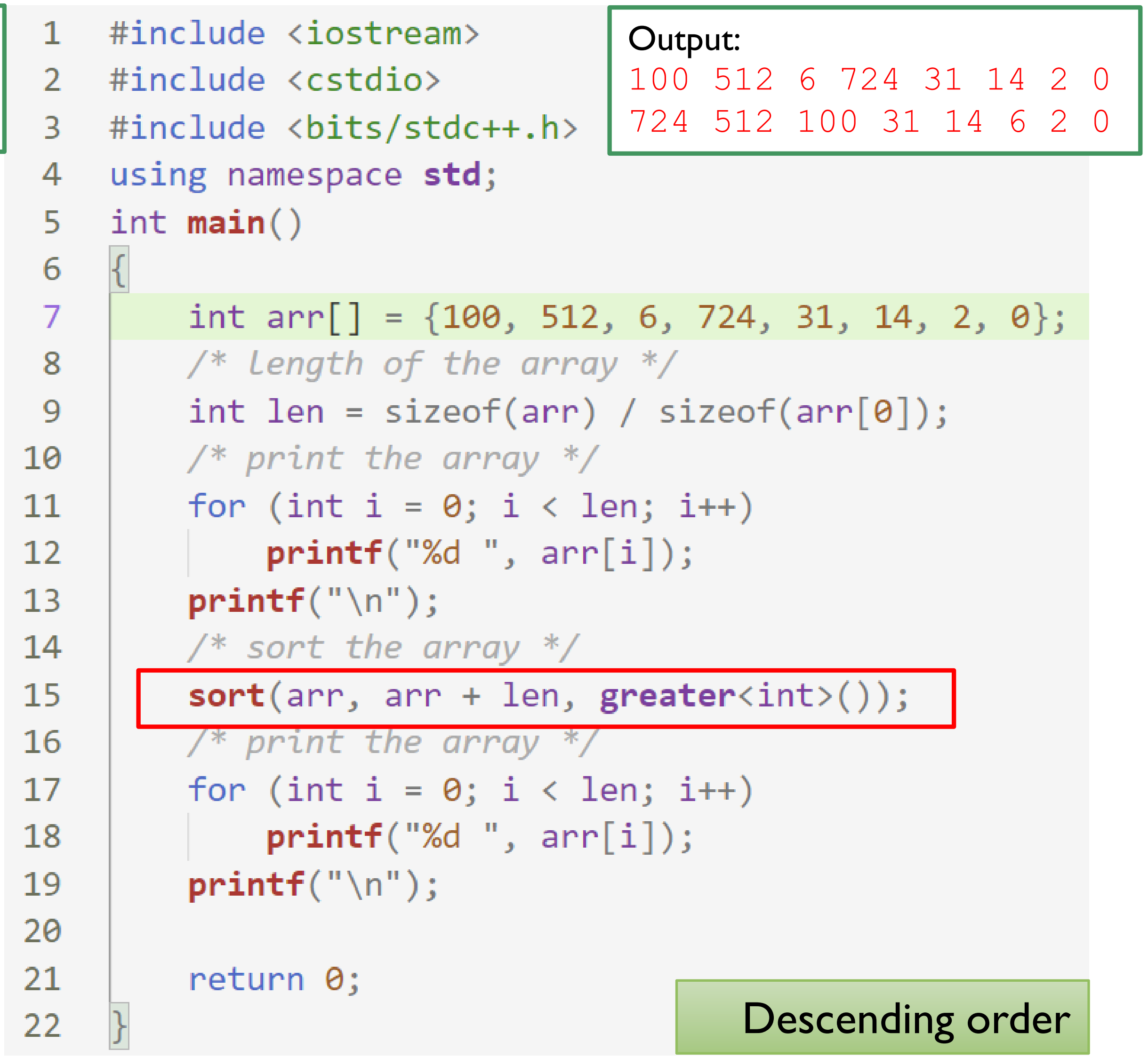


```cpp
#include <iostream>
#include <cstdio>
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int arr[] = {100, 512, 6, 724, 31, 14, 2, 0};
    /* length of the array */
    int len = sizeof(arr) / sizeof(arr[0]);
    /* print the array */
    for (int i = 0; i < len; i++)
        printf("%d ", arr[i]);
    printf("\n");
    /* sort the array */
    sort(arr, arr + len);
    /* print the array */
    for (int i = 0; i < len; i++)
        printf("%d ", arr[i]);
    printf("\n");

    return 0;
}
```

Output:

```
100  512  6  724  31  14  2  0
0  2  6  14  31  100  512  724
```

Default order/
Ascending order



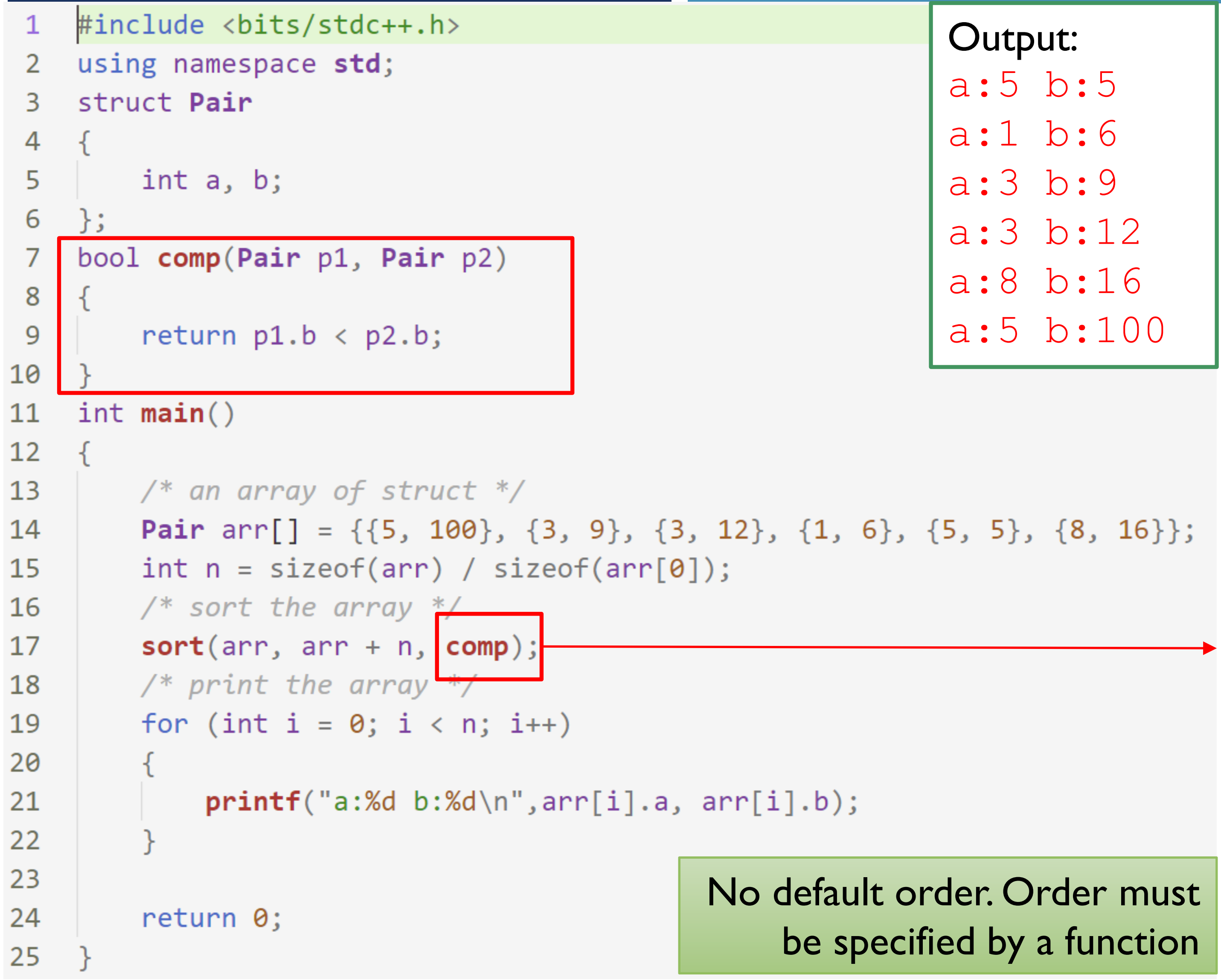
```cpp
#include <iostream>
#include <cstdio>
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int arr[] = {100, 512, 6, 724, 31, 14, 2, 0};
    /* length of the array */
    int len = sizeof(arr) / sizeof(arr[0]);
    /* print the array */
    for (int i = 0; i < len; i++)
        printf("%d ", arr[i]);
    printf("\n");
    /* sort the array */
    sort(arr, arr + len, greater<int>());
    /* print the array */
    for (int i = 0; i < len; i++)
        printf("%d ", arr[i]);
    printf("\n");

    return 0;
}
```

Output:

```
100  512  6  724  31  14  2  0
724  512  100  31  14  6  2  0
```

Descending order

SORT Array of struct



```cpp
#include <bits/stdc++.h>
using namespace std;
struct Pair
{
    int a, b;
};
bool comp(Pair p1, Pair p2)
{
    return p1.b < p2.b;
}
int main()
{
    /* an array of struct */
    Pair arr[] = {{5, 100}, {3, 9}, {3, 12}, {1, 6}, {5, 5}, {8, 16}};
    int n = sizeof(arr) / sizeof(arr[0]);
    /* sort the array */
    sort(arr, arr + n, comp);
    /* print the array */
    for (int i = 0; i < n; i++)
    {
        printf("a:%d b:%d\n",arr[i].a, arr[i].b);
    }

    return 0;
}
```

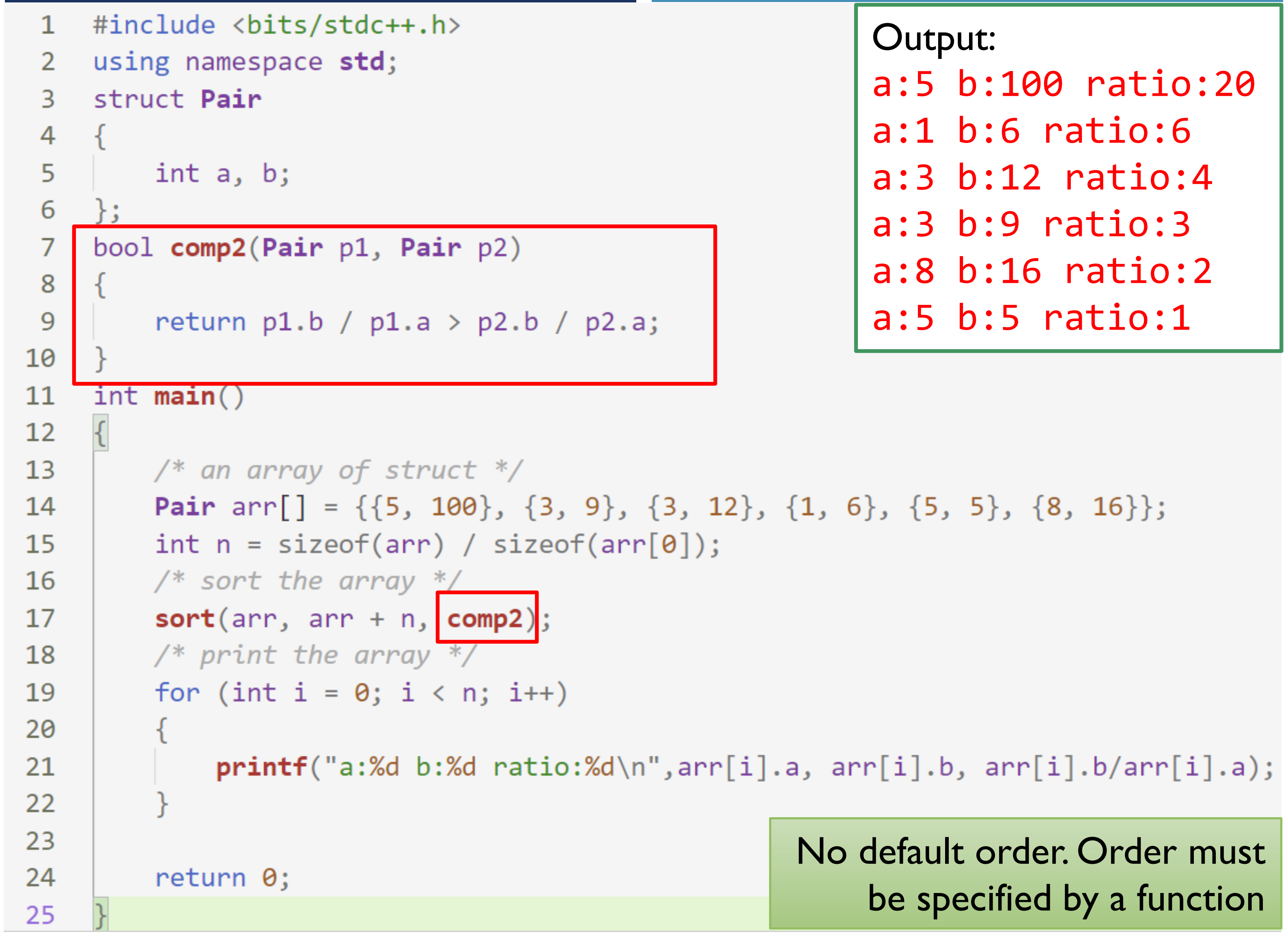
Output:

```
a:5 b:5
a:1 b:6
a:3 b:9
a:3 b:12
a:8 b:16
a:5 b:100
```

this function is a must for sorting an array of struct

No default order. Order must be specified by a function

SORT Array of struct



```cpp
#include <bits/stdc++.h>
using namespace std;
struct Pair
{
    int a, b;
};
bool comp2(Pair p1, Pair p2)
{
    return p1.b / p1.a > p2.b / p2.a;
}
int main()
{
    /* an array of struct */
    Pair arr[] = {{5, 100}, {3, 9}, {3, 12}, {1, 6}, {5, 5}, {8, 16}};
    int n = sizeof(arr) / sizeof(arr[0]);
    /* sort the array */
    sort(arr, arr + n, comp2);
    /* print the array */
    for (int i = 0; i < n; i++)
    {
        printf("a:%d b:%d ratio:%d\n",arr[i].a, arr[i].b, arr[i].b/arr[i].a);
    }

    return 0;
}
```

Output:

```
a:5 b:100 ratio:20
a:1 b:6 ratio:6
a:3 b:12 ratio:4
a:3 b:9 ratio:3
a:8 b:16 ratio:2
a:5 b:5 ratio:1
```

No default order. Order must be specified by a function

REFERENCES

- https://beginnersbook.com/2017/08/c-plus-plus-tutorial-for-beginners/
- https://www.edureka.co/blog/vectors-in-cpp/