

19강 20강
자료구조
Code 추가 설명

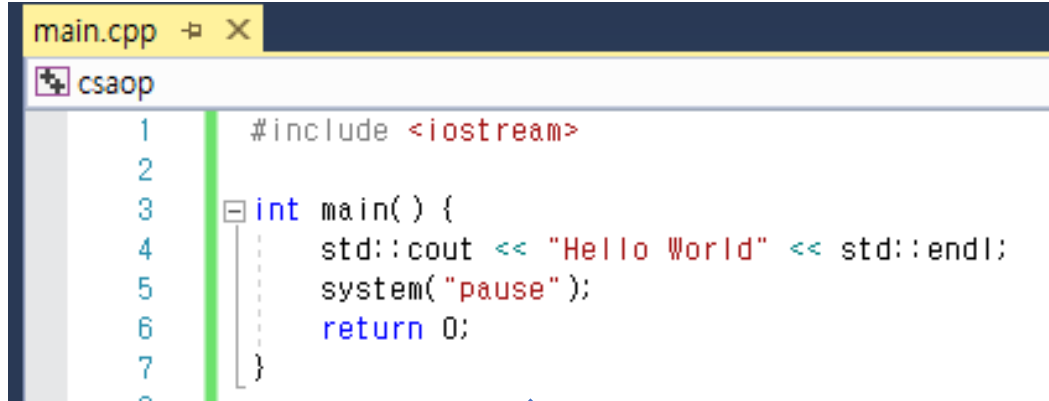
2018.11.23

김상근

내용

- 본 자료는 C/C++ 19, 20강에 나오는 내용에 대한 간단한 설명입니다.
- 아직 C++에 대한 강의가 정식으로 나와 있지 않은 관계로 간단하게 정리 하였습니다.
- 자세한 내용은 온라인 검색을 참고하세요.
- 지금 당장은 본 자료의 내용을 몰라도 강의를 이해하는데 크게 문제 없습니다.

C++ Hello World with namespace



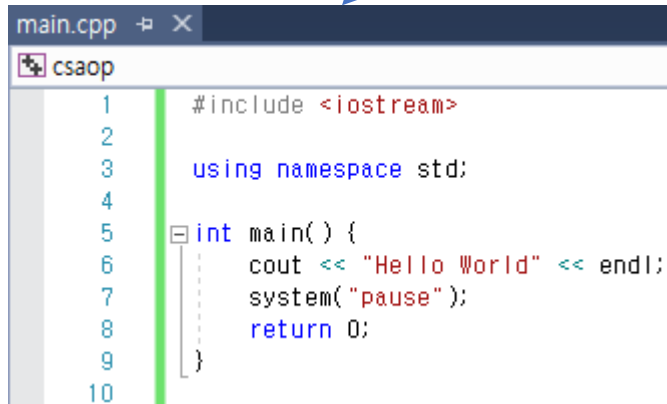
```
main.cpp X
csaop
1 #include <iostream>
2
3 int main() {
4     std::cout << "Hello World" << std::endl;
5     system("pause");
6     return 0;
7 }
```

iostream은 Input/Output Stream(입출력 스트림)의 줄임말 입니다.

printf("Hello world") => std::cout << "Hello world"

printf("\n") => std::endl;

printf("Hello world\n") => std::cout << "Hello world\n"



```
main.cpp X
csaop
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     cout << "Hello World" << endl;
7     system("pause");
8     return 0;
9 }
10
```

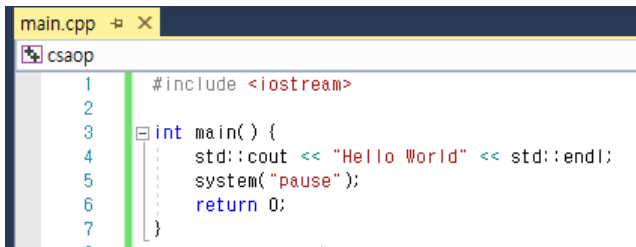
using namespace std; 를 통해서

std라는 namespace를 사용하겠다고 선언함

따라서 std::cout대신 cout만 사용해도 됩니다.

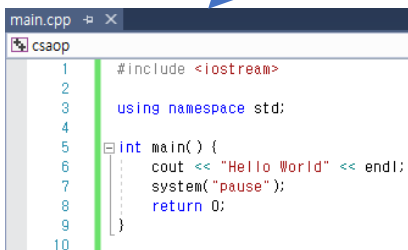
namespace

- namespace는 C++에서 제공되는 문법으로 말 그대로 이름 공간을 말합니다.



```
1 #include <iostream>
2
3 int main() {
4     std::cout << "Hello World" << std::endl;
5     system("pause");
6     return 0;
7 }
```

std라는 이름 공간을 사용하겠다고
using namespace std; 를 코드에 삽입하면



```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     cout << "Hello World" << endl;
7     system("pause");
8     return 0;
9 }
10
```

std라는 이름공간에 있는 함수인 cout과 endl, cin등을
사용하기 위해서 std::사용할 필요가 없다.

설명에 나오는 Code

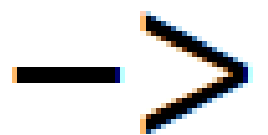
`std::cout` = C언어에서 `printf`와 유사

`std::cin` = C언어에서 `scanf`와 유사

`std::endl` = C언어에서 `printf("\n")`과 유사

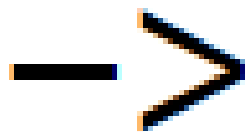
못보던 기호 -> ?? ->

```
main.c  X
csaop
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio.h>
4  #include <stdlib.h>
5
6  typedef struct {
7      int data;
8      struct Node *next;
9  } Node;
10
11  Node *head;
12
13  int main(void) {
14      head = (Node*)malloc(sizeof(Node));
15      Node *node1 = (Node*)malloc(sizeof(Node));
16      node1->data = 1;
17      Node *node2 = (Node*)malloc(sizeof(Node));
18      node2->data = 2;
19      head->next = node1;
20      node1->next = node2;
21      node2->next = NULL;
22      Node *cur = head->next;
23      while (cur != NULL) {
24          printf("%d ", cur->data);
25          cur = cur->next;
26      }
27      system("pause");
28      return 0;
29  }
30
31
```



이게 뭘까?

```
main.c  X
csaop
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio.h>
4  #include <stdlib.h>
5
6  typedef struct {
7      int data;
8      struct Node *next;
9  } Node;
10
11  Node *head;
12
13  int main(void) {
14      head = (Node*)malloc(sizeof(Node));
15      Node *node1 = (Node*)malloc(sizeof(Node));
16      node1->data = 1;
17      Node *node2 = (Node*)malloc(sizeof(Node));
18      node2->data = 2;
19      head->next = node1;
20      node1->next = node2;
21      node2->next = NULL;
22      Node *cur = head->next;
23      while (cur != NULL) {
24          printf("%d ", cur->data);
25          cur = cur->next;
26      }
27      system("pause");
28      return 0;
29  }
30
31
```



구조체가 가지고 있는 변수를 가르치기 위해서 사용

```
main.c  X
csaop
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio.h>
4  #include <stdlib.h>
5
6  typedef struct {
7      int data;
8      struct Node *next;
9  } Node;
10
11  Node *head;
12
13  int main(void) {
14      head = (Node*)malloc(sizeof(Node));
15      Node *node1 = (Node*)malloc(sizeof(Node));
16      node1->data = 1;
17      Node *node2 = (Node*)malloc(sizeof(Node));
18      node2->data = 2;
19      head->next = node1;
20      node1->next = node2;
21      node2->next = NULL;
22      Node *cur = head->next;
23      while (cur != NULL) {
24          printf("%d ", cur->data);
25          cur = cur->next;
26      }
27      system("pause");
28      return 0;
29  }
30
31
```

node1->data = 1

위 코드의 뜻은

node1의 data에 1를 넣는다. => **(*node1).data=1**

```

main.c  X
csaop
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio.h>
4  #include <stdlib.h>
5
6  typedef struct {
7      int data;
8      struct Node *next;
9  } Node;
10
11  Node *head;
12
13  int main(void) {
14      head = (Node*)malloc(sizeof(Node));
15      Node *node1 = (Node*)malloc(sizeof(Node));
16      (*node1).data = 1;
17      Node *node2 = (Node*)malloc(sizeof(Node));
18      (*node2).data = 2;
19      Node *node3 = (Node*)malloc(sizeof(Node));
20      (*node3).data = 3;
21
22      (*head).next = node1;
23      (*node1).next = node2;
24      (*node2).next = node3;
25      (*node3).next = NULL;
26      Node *cur = (*head).next;
27      while (cur != NULL) {
28          printf("%d ", (*cur).data);
29          cur = (*cur).next;
30      }
31      system("pause");
32      return 0;
33

```

-> 명령어를 (*link_list_name).value로 변경한 코드입니다.

node3까지 추가하였으니

직접 수행해보고 결과가 맞게 나오는지 확인해보세요