

# 게임공학과 20203324 최종현

## 4주차

```
✓ #include <iostream>
  #include <vector>
  #include <string>
  using namespace std;

✓ namespace my { // namespace 정의
  // namespace 이름공간은 변수, 함수, 클래스 등과 같은 식별자의 범위를
  // 정의하는 선언적 영역이다.
  // 즉, 코드의 특정 부분을 논리적으로 그룹화하고, 같은 이름을 가진
  // 식별자들이 서로 충돌하지 않도록 구분하는 역할을 한다.
  int value = 10;
}

namespace your {
  int value = 20;
}

void Hello(); // 함수 선언
void Hello1(int a); |
✓ // void fun() 반환 자료형 , 함수명(함수가 받는 입력 데이터들)
  //{
  // 함수의 기능이 들어가야 한다.
int main(int argc, char const* argv[])
{
  // 실습

  Hello1(10); // 함수 호출
  Hello();
  Hello1(20);
  Hello();
  Hello1(30);
  Hello();
```

```

void Hello() // 함수 정의
{
    // void 자료형을 사용하는 이유는?
    // 해당 함수는 어떠한 값도 반환하지 않는다는 것을 컴파일러와 사용자에게 명확하게 알린다.
    // void 함수는 return 문을 사용하거나 생략할 수 있지만, return문에 값을 지정하면
    // 컴파일 오류가 발생한다.
    cout << "Hello World" << endl;
    cout << endl;
}
void Hello1(int a) // 자료형을 넣고 매개변수를 넣어야 하는 이유는?
{
    // 함수의 매개변수에 자료형을 명시하는 것은 컴파일러의 자료형 검사, 함수 호출 시
    // 자료형 일치, 함수 내부 코드의 자료형 일치를 위해 필수적이다.

    // 매개변수를 명시하는 것은 함수 내부에서 외부로부터 전달받은 값을 사용하고,
    // 함수 호출 시 인수를 전달하기 위해 필수적이다.
    cout << "Hello World" << endl;
    cout << a << endl;
}

```

```

Hello World
10
Hello World

Hello World
20
Hello World

Hello World
30
Hello World

Hello World

```

```

// 1번 문제
// 다음 줄에 화면에 Hello World!! 를 출력하는 코드 작성

cout << "Hello World" << endl;
cout << endl;

// 2번 문제
int first = 0, second = 0;
// 다음 줄에 앞에서 선언된 변수 first, second에 각각 10, 20을 저장하는 코드 작성

cin >> first>>second;
cout << first << ", " << second << endl;
cout << endl;

// 3번 문제
// char buf[255] = "Hello World!";
char buf[255] = {};

// 다음 줄에 키보드로 부터 Hello World! 를 입력받아 buf 에 저장하는 코드 작성
// cin >> buf; 이렇게만 사용한다면 띄어쓰기, 줄바꿈을 하면 그 이전의 문자열만 저장한다.
// 그렇기 때문에 getline()함수를 사용해야 한다.
cin.getline(buf, 255);

cout << buf << endl;
cout << endl;

```

```
Hello World
```

```
10 20
10, 20
```

```
Hello World!
Hello World!
```

```

// 4번 문제
// 다음 줄에 10이 저장된 변수 value 값을 화면에 출력하는 코드 작성
cout << my::value << endl;
// namespace 사용
cout << your::value << endl;

cout << endl;
// 다음 줄에 20이 저장된 변수 value 값을 화면에 출력하는 코드 작성

// 5번 문제
cout << "Hello"<<endl;
cout << "World"<< endl;
cout << endl;

// 6번 문제

int a = 20;

cout << a << endl;
cout << endl;

// 7번 문제
int value = 65; // 'A' 의 ASCII 값은 65 임
cout << value << endl;
cout << endl;
// value에 저장된 값을 변수 ch를 선언하고 value의 값을 ch에 복사하는 코드를 작성
char ch = value;
cout << ch << endl;
cout << endl;

```

```

10
20

Hello
World

20

65

A

```

```
// 8번 문제
cout << "true" << endl;
cout << true << endl;
cout << "false" << endl;
cout << false << endl;

// "true"는 문자열
// true는 bool값으로 1
// "false"는 문자열
// faslse는 bool값으로 0
```

```
true
1
false
0
```