

생성자

멤버 변수의 초기화

클래스를 이용하여 객체를 생성하면, 해당 객체는 메모리에 즉시 생성된다. 이 객체는 모든 멤버 변수를 초기화하기 전에는 사용할 수 없다. 객체의 멤버 중에는 직접 접근할 수 없는 private 멤버 변수가 있기 때문에 일반적인 초기화 방식으로 초기화 할 수 없다. 따라서, private 멤버에 접근할 수 있는 초기화를 위한 public 함수가 필요하다.

생성자 (Constructor)

C++에서는 객체의 생성과 동시에 멤버 변수를 초기화해주는 생성자라는 멤버 함수를 제공한다.
다음은 생성자의 특징이다.

1. 생성자는 초기화를 위한 데이터를 인자로 전달받을 수 있다.
2. 생성자는 리턴값이 없지만, void 형으로 선언하지 않는다.
3. 객체를 초기화하는 방법이 여러 개 존재할 경우에는 오버로딩 규칙에 따라 여러 개의 생성자를 가질 수 있다.

선언

```
class Person {  
    public:  
        string name;  
        int age ;  
        int weight;  
        int height ;  
  
    / Constructor  
    Person(string name , int age , int w, int h) { /  
        this->name = name;  
        this->age = age;  
        weight = w;  
        height = h;  
    }  
  
    void userInfo() {  
        cout << "NAME : " << name << endl;  
        cout << "AGE : " << age << endl;  
    }  
};
```

```

        cout << "WEIGHT : " << weight << endl;
        cout << "HEIGHT : " << height << endl;
    }

    ~Person();
};

```

클래스 생성자의 원형은 클래스 선언의 public 영역에 포함되어야 한다.

소멸자

생성자는 객체 멤버의 초기화뿐만 아니라, 객체를 사용하기 위한 외부 환경까지도 초기화하는 역할을 한다. 따라서 객체의 수명이 끝나면 생성자의 반대 역할을 수행할 멤버 함수도 필요하다. 이러한 역할을 하는 멤버 함수를 소멸자(destructor)라고 한다. 소멸자는 객체의 수명이 끝나면 컴파일러에 의해 자동으로 호출되며, 사용이 끝난 객체를 정리해 준다. 소멸자의 이름은 해당 클래스의 이름과 같으며, 이름 앞에 물결 표시(~)를 붙여 생성자와 구분한다. 즉, Person 클래스의 소멸자는 ~Person()이라는 이름을 가지게 된다.

다음은 소멸자의 특징이다.

1. 소멸자는 인수를 가지지 않는다.
2. 소멸자는 반환값이 없지만 void 형으로 선언하지 않는다.
3. 객체는 여러 개의 생성자를 가질 수 있지만, 소멸자는 단 하나만 가질 수 있다.
4. 소멸자는 const, volatile 또는 static 으로 선언될 수는 없지만, const, volatile 또는 static 으로 선언된 객체의 소멸을 위해서 호출될 수는 있다.

선언

```

~Person();

```