구조체와 공용체

제주대학교 컴퓨터공학과
변영철 교수
(ycb@jejunu.ac.kr)

이 장을 공부하면

- 코드를 묶어 추상화하듯이 변수들을 묶어 추상화할 수 있다.
- 구조체와 공용체를 이해할 수 있다.
- 열거형을 이해할 수 있다.
- 자료형 이름을 간단히 만들 수 있다.

추상화

- 코드를 묶어서 간단히 표현
  - 코드 추상화
  - 그 결과 함수가 만들어짐
  - 예를 들어, 가장 간단한 C 프로그램을 작성한 후 printf 문장을 Say라는 함수로 추상화해 보자.

추상화

```
#include <stdio.h>

void main() {
	printf("Hello,World!\n");
	getchar();
}
```

추상화

```
#include <stdio.h>

void Say() {
	printf("Hello,World!\n");
}
void main() {
	Say();

	getchar();
}
```

지역변수 정의

학번과 학점을 저장하는 변수를 만들고 여기에 값을 저장한 후 출력해보자.

```
#include <stdio.h>

void main() {
        int num;
        double grade;

        num = 2;
        grade = 2.7;

        printf("학번 : %d\n", num);
        printf("학점 : %.1lf\n", grade);

        getchar();
}
```

전역변수로

지역변수를 전역변수로...
어떤 의미가 있을까?

```
#include <stdio.h>

int num;
double grade;

void main() {
        num = 2;
        grade = 2.7;

        printf("학번 : %d\n", num);
        printf("학점 : %.1lf\n", grade);

        getchar();
}
```

코드 추상화

값을 할당하는 코드 추상화

```
void Assign(int n, double g) {
        num = n;
        grade = g;
}
void main() {
        Assign(2, 2.7);

        printf("학번 : %d\n", num);
        printf("학점 : %.1lf\n", grade);

        getchar();
}
```

코드 추상화

표시하는 코드 추상화

```
void Display() {
        printf("학번 : %d\n", num);
        printf("학점 : %.1lf\n", grade);
}
void main() {
        Assign(2, 2.7);
        Display();

        getchar();
}
```

코드 추상화

변수를 묶어 추상화
묶어서 만든 **자료형**

'자료형은 뭐 하라고 있는 것?'
변수 만들라고 있는 것!

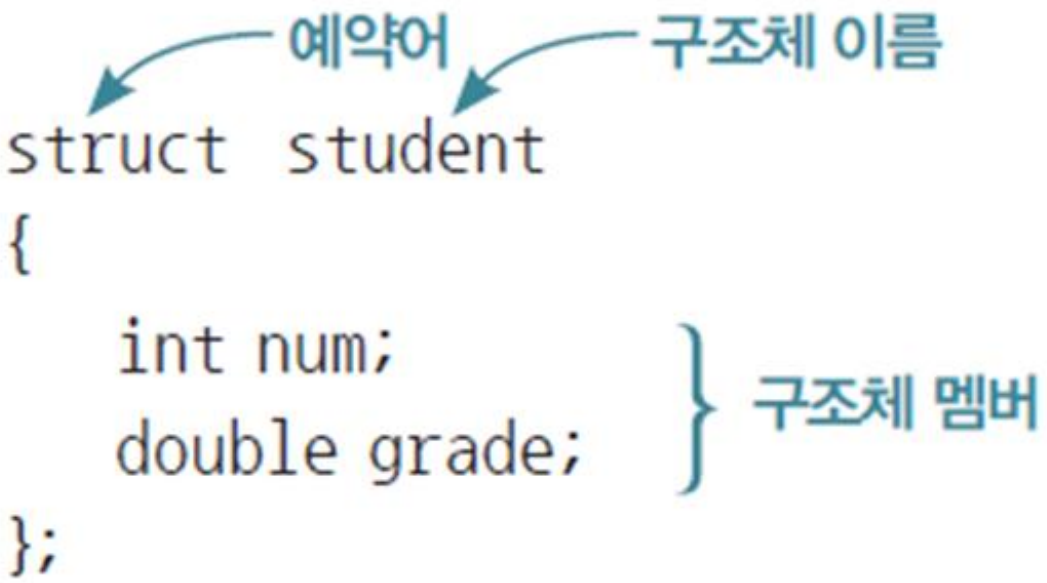
```
#include <stdio.h>

struct student {
        int num;
        double grade;
};

struct student s1;
```

구조체 자료형 선언

새로운 구조체 **자료형** '**struct student**'-> "자료형은 뭐 하라고 있는 것?"

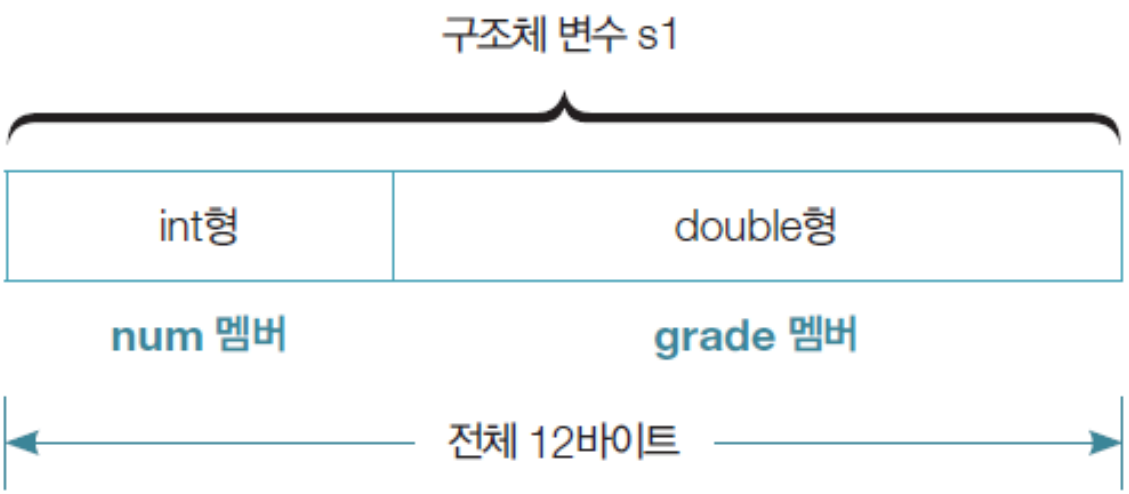


구조체 변수 정의

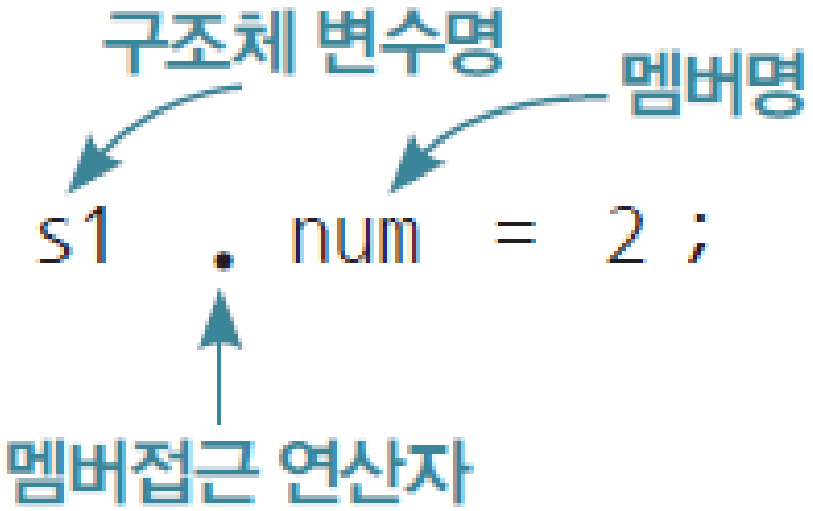
```
struct student s1;
```

구조체 → struct student

변수명 → s1



구조체 변수 접근



구조체 변수 접근

```
struct student s1;

void Assign(int n, double g) {
        s1.num = n;
        s1.grade = g;
}
void Display() {
        printf("학번 : %d\n", s1.num);
        printf("학점 : %.1lf\n", s1.grade);
}
void main() {
        Assign(2, 2.7);
        Display();

        getchar();
}
```

이름 추가

학번, 학점과 함께
이름도 저장하려면?

```
#include <stdio.h>
#include <string.h>

struct student {
        char name[20];
        int num;
        double grade;
};

struct student s1;
```

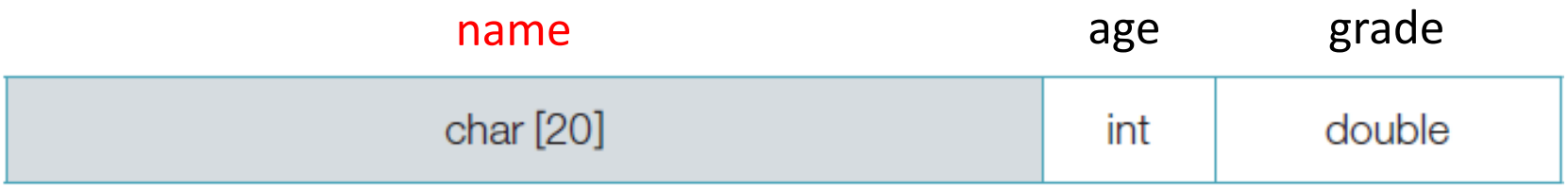
이름 추가

이름도 같이 할당하고 표시하려면...

```
void Assign(char irum[20], int n, double g) {
        strcpy(s1.name, irum);
        s1.num = n;
        s1.grade = g;
}
void Display() {
        printf("이름 : %s\n", s1.name);
        printf("학번 : %d\n", s1.num);
        printf("학점 : %.1lf\n", s1.grade);
}
void main() {
        Assign("Gildong Hong", 2, 2.7);
        Display();

        getchar();
}
```

이름 추가



구조체 변수 초기화

```
void main() {
        s1 = { "Gildong Hong", 2, 2.7};
        Assign("Gildong Hong", 2, 2.7);
        Display();

        getchar();
}
```

구조체 배열

여러 사람(최대 100명)을
저장하기 위하여
배열 정의

그리고 현재 몇 명이
있는지 저장하기 위한 변수
count 정의

```
#include <stdio.h>
#include <string.h>

struct student {
        char name[20];
        int num;
        double grade;
};

struct student s1[100];
int count = 0;
```

구조체 배열

Assign 함수
배열 끝에 추가하고
숫자를 하나 증가시킴.

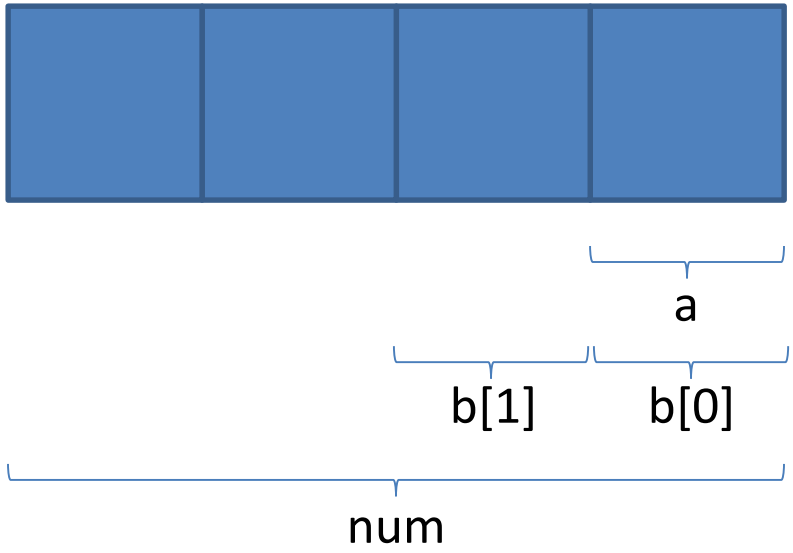
Display 함수
저장되어 있는 모든
정보를 출력함.

```
void Assign(char irum[20], int n, double g) {
        strcpy_s(s1[count].name, irum);
        s1[count].num = n;
        s1[count].grade = g;
        count = count + 1;
}
void Display() {
        for (int i = 0; i < count; i++) {
                printf("이름 : %s\n", s1[i].name);
                printf("학번 : %d\n", s1[i].num);
                printf("학점 : %.1lf\n", s1[i].grade);
        }
}
void main() {
        Assign("Gildong Hong", 100, 2.7);
        Assign("Cheolsu Kim", 101, 3.8);
        Display();

        getchar();
}
```

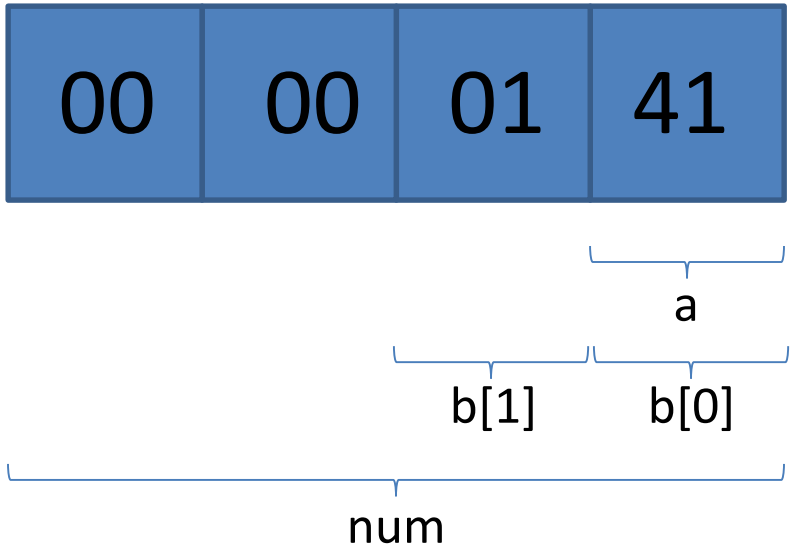
공용체

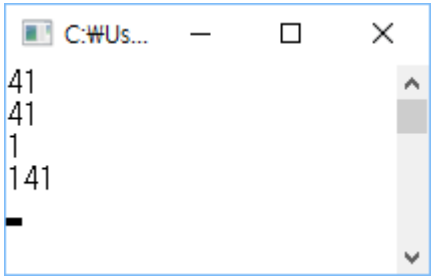
```
union student {
        char a;
        char b[2];
        int num;
};
union student info;
```



공용체

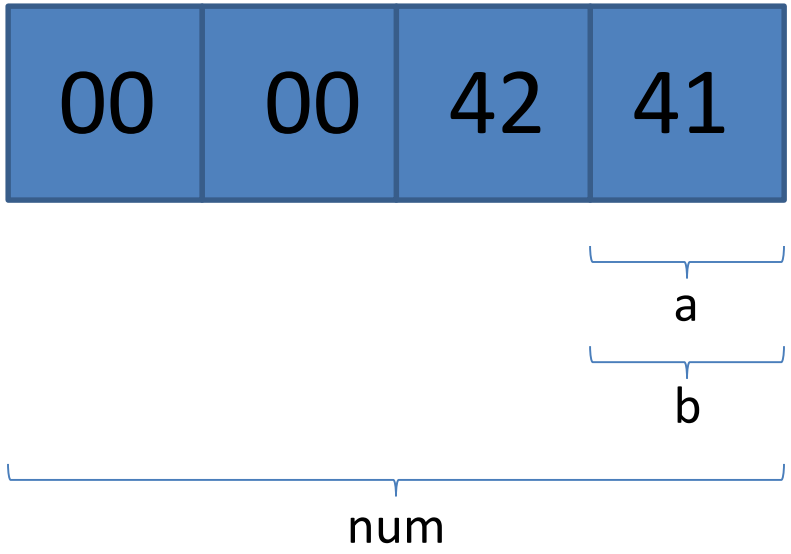
```
void main() {
        info.num = 0x00000141;
        printf("%X\n", info.a);
        printf("%X\n", info.b[0]);
        printf("%X\n", info.b[1]);
        printf("%X\n", info.num);
}
```

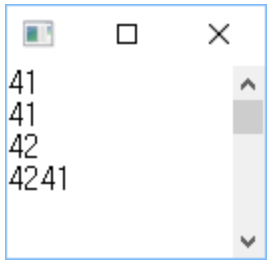




공용체

```
void main() {
        info.num = 0x00004241; // (16진수)
        printf("%c\n", info.a);
        printf("%c\n", info.b);
        printf("%d\n", info.num);
}
```





열거형

열거형을 쓰지 않을 경우 어떤 불편한 점이 있을까?

숫자를 쓸까, 기호를 쓸까?

```
enum season {
        SPRING, SUMMER, FALL, WINTER
};

void main() {
        int now = FALL;
        switch (now) {
        case SPRING:
            printf("아름다운 벚꽃!\n");
            break;
        case SUMMER:
            printf("시원한 바다!\n");
            break;
        case FALL:
            printf("아름다운 단풍!\n");
            break;
        case WINTER:
            printf("하얀  눈사람!\n");
            break;
        }
}
```

typedef 문

자료형 이름을 간단히 만들어 변수 만들 때 편함.

```
#include <stdio.h>
#include <string.h>
struct student {
        char name[20];
        int num;
        double grade;
};
typedef struct student Student;
Student s1;
```

typedef 문

```
#include <stdio.h>
#include <string.h>
typedef struct {
        char name[20];
        int num;
        double grade;
} Student;
Student s1;
```