Introduction to C Programming

Section 10

Introduction

- Our variables until now
  - Single variable

```
int i, char c, float f
```

- Set of same type elements: Array

```
int a[10], char c[20]
```

- If data are not same type, but related? Example: Information about students
  - Student Name
  - Student Family Name
  - Student Number
  - Student Grade

Structure Basics

- A structure is a collection of data values, called *data members*, that form a single unit.
- Unlike arrays, the data members can be of different types.

struct: version 1

- Set of related variables
  - Each variable in **`struct`** has its own type

`struct` in C (version 1)

```
struct {
    <variable declaration>
} <identifier list>;
```

struct (version 1): Example

```
struct{
char st_name[20];
char st_fam_name[20];
  int id;  int grade;
} st1;
```

- We declare a variable `st1`
- Type of `st1` is struct
- `id` is a member of the struct
- `grade` is a member of the struct

struct: Version 2

`struct` in C (version 2)

```
struct <tag> {
    <variable declaration>
};

struct <tag> <identifiers>;
```

struct (version 2): Example

```
struct std_info{
  char st_name[20];
  char st_fam_name[20];
  int id;  int grade;
};

struct std_info st1, st2, st3;
```

- We define a struct with tag `std_info`
  - We don't allocate memory, it is just definition
- We declare variables `st1, st2, st3` from `std_info`

Pointer to struct: Definition

- A variable of struct type is a variable
- It has address, we can have pointer to it

```
struct std{
   int id;
   int grade;

};
struct std st1;

struct std *ps;

ps = &st1;
```

Pointer to struct: Usage (Version 1)

- We can use *pointer method
- **`*ps`** means the content of the address that
- **`ps`** refers to there > it is struct

- **`(*ps).id`** is the member of struct that `ps` refers to it
- **`(*ps).grade`** is the member of struct that `ps` refers to it

Pointer to struct: Usage (Version 2)

- We can use “->" method

```
struct std{
     int id;
     int grade;

};
struct std st1, *ps;
ps = &st1;
int y = ps->id; // (*ps).id
int z = ps->grade; // (*ps).grade
```