Lecture 20

User Defined Types

CSE115: Computing Concepts

Passing Structures to a Function

Call by Value:

- We can pass the student structure that we have created before to a function called display( ) as follows:

```
void display (struct student s1);
/*function prototype*/
display (student1);  /* function call
*/
```

  where `student1` is a variable of type `struct student`.

- In the above function, a copy of the student structure will be created locally for the use of the function. *Any changes to the structure inside the function will not affect the actual structure.*

Example Using Structure: Call by Value

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

struct student
{
        char name[20];
        int id;
};
void display(struct student s1)  /* make a local copy of the structure */
{
        printf("Name: %s\n", s1.name);
        printf("ID: %d\n", s1.id);
}
int main(void)
{
        struct student student1;
        strcpy(student1.name, "Ahmad");   /*initialising variable */
        student1.id = 12345;              /*initialising variable */
        display(student1);
        return 0;
}
```

Example Using Structure: A Function that Returns a Structure

```
#include <stdio.h>
#include <string.h>
struct student
{
        char name[20];
        int id;
};
```

```
struct student read(void)
{
        struct student s1;
        printf("Enter name:");
        gets(s1.name);
        printf("Enter ID:");
        scanf("%d",&s1.id);
        return s1;
}
void main(void)
{
        struct student student1;
        student1 = read();
        printf("Name: %s", student1.name);
        printf("\nID: %d\n", student1.id);
}
```

Call by reference

- It is also possible to use pointers and pass the reference of the structure to the function. This way, *any changes inside the function will change the actual structure as well.*
- To pass a structure variable as a reference, the Read( ) function can be written this way:

```
void Read(struct student *s1);  /* function prototype */
Read(&student1); /* function call */
```

where `student1` is a variable of type `struct student`.

Call by reference

- Take note that when a structure is declared as a pointer, the elements in the structure cannot be referred to using the '.' operator anymore. Instead, they need to be accessed using the '->' operator (**indirect component selection operator**).

- For example:

```
void Read(struct student *s1)
{
        s1->studentID = 10179;
        scanf("%s", s1->name);
}
```

Example Using Structure: Call by reference

```
#include <stdio.h>
#include <string.h>
struct student
{
        char name[20];
        int id;
};
```

```
void Read (struct student *s1)
{
        printf("Enter name:");
        gets(s1->name);
        printf("Enter ID:");
        scanf("%d",&s1->id);
}

void main(void)
{
        struct student student1;

        Read(&student1);
        printf("Name: %s", student1.name);
        printf("\nID: %d\n", student1.id);
}
```