

# **EECE.2160: ECE Application Programming**

Spring 2017

## **Lecture 26: Key Questions**

March 31, 2017

1. Explain how 2-D arrays are passed to functions, and how functions are used with 2-D arrays in Program 7.

2. Explain how structures can be nested inside one another.

For today's exercise, you will complete the following functions that work with the structures `Name` and `StudentInfo`. The structure definitions are listed below:

```
typedef struct {  
    char first[50];  
    char middle;  
    char last[50];  
} Name;
```

```
typedef struct {  
    Name sname;  
    unsigned int ID;  
    double GPA;  
} StudentInfo;
```

The function descriptions are as follows:

For the `Name` structure:

- **void printName(Name \*n):** Print the name pointed to by `n`, using format `<first> <middle>. <last>`
- **void readName(Name \*n):** Prompt for and read a first, middle, and last name, and store them in the structure pointed to by `n`

For the `StudentInfo` structure:

- **void printStudent(StudentInfo \*s):** Print information about the student pointed to by `s`
- **void readStudent(StudentInfo \*s):** Prompt for and read information into the student pointed to by `s`
- **void printList(StudentInfo list[], int n):** Print the contents of an array `list` that contains `n` `StudentInfo` structures
- **int findByLName(StudentInfo list[], int n, char lname[]):** Search for the student with last name `lname` in the array `list`. Return the index of the structure containing that last name, or -1 if not found
- **int findByID(StudentInfo list[], int n, unsigned int sID):** Search for the student with ID # `sID` in the array `list`. Return the index of the structure containing that last name, or -1 if not found

From Name.c:

```
// Print contents of Name struct
void printName(Name *n) {

}

// Read information into existing Name
void readName(Name *n) {

}

}
```

From StudentInfo.c:

```
// Print information about student
void printStudent(StudentInfo *s) {

}

// Reads student information into existing structure
void readStudent(StudentInfo *s) {

}

}
```

**From StudentInfo.c (continued):**

```
// Print list of students
void printList(StudentInfo list[], int n) {

}

// Find student in list, based on last name
// Returns index if student found, -1 otherwise
int findByName(StudentInfo list[], int n, char lname[]) {

}

}
```

**From StudentInfo.c (continued):**

```
// Find student in list, based on ID #  
// Returns index if student found, -1 otherwise  
int findByID(StudentInfo list[], int n, unsigned int sID) {
```

```
}
```