## **EECE.2160: ECE Application Programming**Spring 2017

Lecture 24: Key Questions March 24, 2017

2.	Explain how we can essentially declare our own types using structures.

1. Describe what a structure is in C, and how structures can be useful.

3. Show how variables of a given structure type can be declared and initialized.

4. Show how elements within a structure can be accessed.

## 5. **Example:** What does the following program print?

```
#include <stdio.h>
typedef struct {
     double real;
     double imag;
} Complex;
int main() {
     Complex a = \{1, 2\};
     Complex b = \{3.4, 5.6\};
     Complex c, d, e;
     printf("A = %.21f + %.21fi\n", a.real, a.imag);
    printf("B = %.21f + %.21fi\n", b.real, b.imag);
     c = a;
     d.real = a.real + b.real;
     d.imag = a.imag + b.imag;
     e.real = a.real - b.real;
     e.imag = a.imag - b.imag;
     printf("C = %.21f + %.21fi\n", c.real, c.imag);
     printf("D = %.21f + %.21fi\n", d.real, d.imag);
     printf("E = %.21f + %.21fi\n", e.real, e.imag);
     return 0;
}
```

6. **Example:** Write the following functions that use the StudentInfo structure

- Given a pointer to a single StudentInfo variable, print all of the student info to the screen using the following format:
  - o Michael J. Geiger
  - o ID #12345678
  - o GPA: 1.23

• Given an array of StudentInfo variables, compute and return the average GPA of all students in the list

• Prompt the user to enter 3 lines of input (using the format below), read the appropriate values into StudentInfo elements, and return a value of type StudentInfo

o Format (user input <u>underlined</u>)

o Enter name: Michael J. Geiger

o Enter ID #:  $\overline{12345678}$ 

o Enter GPA: 1.23