

## Lab 6: Structs and Header Files in C

### Instructions:

In this lab, you will create a simple program to manage a list of students and their grades. You'll define a struct to store each student's details and use functions to populate and display student information. This lab will involve two files, names must match below:

- `lab6.c` (main program and function implementations)
- `student.h` (header file with struct definition and function prototypes).

### Goals:

By the end of this lab, you should be able to:

1. Create and use a header file to organize function prototypes and struct definitions.
2. Define and manipulate structures in C.
3. Use an array of structs to store and process multiple records.
4. Apply modular programming by separating interface (header file) and implementation (source file).

### Sample Output:

```
Enter the number of students: 2

Entering data for student 1
Enter student name: Alice
Enter student ID: 1001
Enter student grade: 85

Entering data for student 2
Enter student name: Bob
Enter student ID: 1002
Enter student grade: 90

Student Information:
Name: Alice, ID: 1001, Grade: 85.00
Name: Bob, ID: 1002, Grade: 90.00

Average Grade: 87.50
```

### Grading Criteria:

1. Program executes and achieves desired output: 100 points
  - a. Program should compile without errors (if it does not compile, 0 points).
2. Correct struct and function implementation: -10 points each
  - a. Deduct for incorrect or missing implementations of the Student struct, `input_student_data()`, `display_student_data()`, or `calculate_average_grade()`.
3. Output Formatting: -5 points
  - a. Deduct for mismatches in output formatting, as autograding relies on exact match.