CS111 – Introduction to C Programming
Fall 2025
Programming Assignment #11 (3%)
Due: Friday December 5 (23:59)
C Structures and Unions

Objectives

In this programming assignment, you will practice the use of C structures, unions, and enumerations in various situations.

Part I: Arithmetic Operation of Complex Numbers (1%)

Write a program to perform the addition and multiplication of two complex numbers. The program should first read the real and imaginary parts of the two number, store each in a C structure, named **`Complex`**, and pass them in turn to two functions, named **`addComplex()`** and **`multipyComplex()`**, which return the addition and product of the two complex numbers, respectively. The external behavior of your program should be as follows.

```
Input two parts of complex number 1 (real imag): 10 3
Input two parts of complex number 2 (real imag): 12 5

Sum of the two complex numbers:
22.00 + 8.00i

Product of the complex numbers:
105.00 + 86.00i
```

Part II: Date Comparison

Write the following C function, assuming that the C structure **`date`** contains three members:

```
int compareDates(struct date d1, struct date d2);
```

The function returns **`-1`** if **`d1`** is an earlier date than **`d2`**, **`+1`** if **`d1`** is a later date than **`d2`**, and **`0`** if **`d1`** and **`d2`** are the same.

Use the following definition of **`struct date`** to develop your code:

```
struct date {
   int month;
   int day;
   int year;
};
```

Write a main program to test the above function. Your main should prompt a user to input the day, month and year of the two dates, in that order, and then proceed to call **`compareDates()`**. It should print out the result with a meanful message.

Part III: Bug Fixing with C structures and unions

Study the following nested C structures with a nested union.

```
struct point { int x, y; };

struct shape {
    int shape_kind; /* RECTANGLE or CIRCLE */
    struct point center; /* coordinates of center */
    union {
        struct {
          int height, width;
        } rectangle;
        struct {
          int radius;
        } circle;
    } u;
} s;
```

If the value of **shape_kind** is **RECTANGLE**, the **height** and **width** members store the dimensions of a rectangle. If the value of **shape_kind** is **CIRCLE**, the **radius** member stores the radius of a circle. First, define constants **RECTANGLE** and **CIRCLE** with a C **enum**. Then place the following C instructions in a main program and correct the ones that do not work.

```
(a) s.shape_kind = RECTANGLE;
(b) s.center.x = 10;
(c) s.height = 25;
(d) s.u.rectangle.width = 8;
(e) s.u.circle = 5;
(f) s.u.radius = 5;
```

Marking and Assignment Submission

Once the program is completed, add a header documentation section and inline comments to explain how each program works. The header section should include information about the author and the general information about this assignment as well as about the use of AI as is described in our "AI policy". Each function should have its own header as the main function to describe the input and output the function and the algorithm used to compute output from the input. (This is called "proper documentation" below.) Name the programs for the three parts as **pa11p1, pa11p2** and **pa11p3.c,** respectively. Submit these three files as well as the zipped folder of the three files, called **pa11.zip**, as four separate files through Blackboard. Be prepared to explain to a TA in the next lab session about your solution to this assignment. This programming assignment is worth a total of 3% with 1% for each part.