# CS 24000 Lab Section L04 Week 5

Structs, Complex Numbers, and the Mandelbrot Set

# Structs

- Some of you have been using them in your homeworks
- Like arrays, but more versatile
- Contain ***ordered*** information

Example:

```
struct animal {

        char *name;

        int num_legs;

        char **species;

};
```

# Struct Usage

```
struct animal {
        char *name;
        int num_legs;
        char **species;
};
```

```
struct animal Cat = {“cat”, 4, {“felicius domestic”, NULL}};

// Access basic information
printf(“Name: %s, Legs: %d\n”, Cat.name, Cat.num_legs);

// Access an array inside a struct
for (int i = 0; Cat.species[i] != NULL; i++) {
        printf(“%s\n”, Cat.species[i]);
}
```

```
struct animal {
        char *name;
        int num_legs;
        char **species;
};
```

# Struct Pointer Usage

```
// Similar to a normal struct, but with -> instead of .

struct animal *Cat = malloc(sizeof(struct animal));
*Cat = (struct animal) {“cat”, 4, {“felicius domestic”, NULL}};

// Access basic information
printf(“Name: %s, Legs: %d\n”, Cat->name, Cat->num_legs);

// Access an array inside a struct
for (int i = 0; Cat->species[i] != NULL; i++) {
        printf(“%s\n”, Cat->species[i]);
}
```

# Complex Numbers

- Consist of two components: real (x) and imaginary (y)
  - Perfect for turning into a struct
- In HW4, create a struct with two ints, x and y
- You may find typedef useful for working with struct pointers

# Mandelbrot

- You've probably seen it before
- Perfect use of complex numbers
- Formula is weird if you've never seen it before

  $Z = Z^2 + C$

- You'll need several formulas to make it work:
  1) Add
  2) Negate
  3) Subtract (add + negate)
  4) Dot
  5) Inverse
  6) Multiply
  7) Divide
  8) Exp