C Prog.

Lec 5

Structs

```c
struct Point {
    int x;
    int y;
};

int main() {
    struct Point p = {.x = 320, .y = 100};
    printf("X coordinate of p is %d", p.x);
    printf("Y coordinate of p is %d", p.y);
}
```

Structs simplified with typedef

```c
typedef struct Point {
    int x;
    int y;
} Point;

int main() {
    Point p = {.x = 320, .y = 100};
    printf("X coordinate of p is %d", p.x);
    printf("Y coordinate of p is %d", p.y);
}
```

Struct

```c
typedef struct Point {
    float x;
    float y;
} Point;

float distance_from_origin(Point p) {
    return sqrt(p.x*p.x + p.y*p.y);
}

Point move_point(Point p, float angle, float len) {
    Point p2 = {
        .x = p.x + len*cos(2*3.14*angle/360.0),
        .y = p.y - len*sin(2*3.14*angle/360.0)
    };
    return p2;
}
```

Struct: Anonymous instance

```
typedef struct Point {
    float x;
    float y;
} Point;

void distance_from_origin(Point p) {
    return sqrt(p.x*p.x + p.y*p.y);
}

Point move_point(Point p, float angle, float len) {
    return (Point){
        .x = p.x + len*cos(2*3.14*angle/360.0),
        .y = p.y + len*sin(2*3.14*angle/360.0)
    };
}
```

Struct for Tree Drawing

```
typedef struct Point {
    int x;
    int y;
} Point;
void drawTree(Point p, float angle,
              float len, int level) {
    Point p2 = {
        .x = p.x + len*cos(2*3.14*angle/360.0),
        .y = p.y + len*sin(2*3.14*angle/360.0)
    };
    ...
    if (level <= 10) {
        drawTree(p2, angle - 20, len, level+1);
        drawTree(p2, angle + 20, len, level +1);
    }
}
```

https://fiddle.skia.org/c/71b68e1656277054222b26734ca18610

Building Structs from struct

```c
typedef struct Circle {
    Point center;
    int radius;
} Circle;
```

Building Structs from struct

```
void drawCirce(Circle c) {
...
Circle
        c1 = {.center = {
                        .x = c.center.x + c.radius/2,
                .y = c.center.y},
              .radius = c.radius/2},
        c2 = {.center = {
                        .x = c.center.x - c.radius/2,
                        .y = c.center.y},
              .radius = c.radius/2},
        c3 ={.center = {
                        .x = c.center.x,
                        .y = c.center.y + c.radius/2},
             .radius = c.radius/2},
        c4 ={.center = {
                        .x = c.center.x,
                        .y = c.center.y - c.radius/2},
             .radius = c.radius/2};
...
}
```

https://fiddle.skia.org/c/9b7c1c2a7192611bc3f66c15162c0514

Array of Structs

```
Circle circles[4] = {
        {.center = {
          .x = c.center.x + c.radius/2,
          .y = c.center.y},
        .radius = c.radius/2},
        {.center = {
          .x = c.center.x - c.radius/2,
          .y = c.center.y},
        .radius = c.radius/2},
        {.center = {
          .x = c.center.x,
          .y = c.center.y + c.radius/2},
        .radius = c.radius/2},
        {.center = {
          .x = c.center.x,
          .y = c.center.y - c.radius/2},
        .radius = c.radius/2}
        };
for(int i = 0; i <4; i++) {
    drawCircle(circles[i], canvas, paint);
}
```

https://fiddle.skia.org/c/c888ec30005d8af42b3925e4a919336e

Pointer to struct

```c
void move_point(Point *p, float angle, float len) {
    p->x += len*cos(2*3.14*angle/360.0);
    p->y -= len*sin(2*3.14*angle/360.0);
}
```

https://fiddle.skia.org/c/7c4716225c006ba6a79b41d333b191f3