

Pointers and Structures

Pointers and Structures

- Accessing a structure instance's members via pointer
- Syntax
 - The structure pointer operator (->)
 - p->x accesses the member x of the structure instance pointed to by p
 - This syntax is shorthand for (*p).x

```
struct point {  
    double x;  
    double y;  
};
```

```
//Declaring instance of struct point
```

```
struct point p1;
```

```
//Declaring pointer to a structure instance
```

```
struct point *pointer_to_p1 = &p1;
```

```
//Accessing member of structure instance
```

```
pointer_to_p1->x; // (*pointer_to_p1).x
```

```
pointer_to_p1->y // (*pointer_to_p1).y
```

Code Example

```
#include <stdio.h>
#include <math.h>

struct point {
    double x;
    double y;
};

void initialize_point(struct point *point, double x, double y);

void print_point(const struct point *point);

void distance_and_slope(const struct point *p1, const struct point *p2,
                        double *distance, double *slope);
```

- `initialize_point()`
 - Given a pointer to a struct point structure instance, set its x and y members to be the provided values.
- `print_point()`
 - Print a point with the format (x,y). The point is passed via pointer.
- `distance_and_slope()`
 - Calculate the distance and slope between two points. The points are passed via pointers, and the distance and slope are return via pointers.

initialize_point()

```
int main() {  
    struct point p1;  
    struct point p2;  
  
    initialize_point(&p1, 1.0, 2.5);  
    initialize_point(&p2, 2.0, 5.0);  
  
    double distance, slope;  
  
    distance_and_slope(&p1, &p2, &distance, &slope);  
  
    printf("The distance between ");  
    print_point(&p1);  
    printf(" and ");  
    print_point(&p2);  
    printf("\nis %lf. The slope is %lf\n", distance, slope);  
  
    return 0;  
}
```

```
void initialize_point(struct point *point, double x, double y) {  
    point->x = x;  
    point->y = y;  
}
```

distance_and_slope()

```
int main() {
    struct point p1;
    struct point p2;

    initialize_point(&p1, 1.0, 2.5);
    initialize_point(&p2, 2.0, 5.0);

    double distance, slope;

    distance_and_slope(&p1, &p2, &distance, &slope);

    printf("The distance between ");
    print_point(&p1);
    printf(" and ");
    print_point(&p2);
    printf("\nis %lf. The slope is %lf\n", distance, slope);

    return 0;
}
```

```
void distance_and_slope(const struct point *p1, const struct point *p2,
                        double *distance, double *slope) {
    double x_distance = p2->x - p1->x;
    double y_distance = p2->y - p1->y;

    *distance = sqrt(x_distance * x_distance + y_distance * y_distance);
    *slope = y_distance / x_distance;
}
```

print_point()

```
int main() {
    struct point p1;
    struct point p2;

    initialize_point(&p1, 1.0, 2.5);
    initialize_point(&p2, 2.0, 5.0);

    double distance, slope;

    distance_and_slope(&p1, &p2, &distance, &slope);

    printf("The distance between ");
    print_point(&p1);
    printf(" and ");
    print_point(&p2);
    printf("\nis %lf. The slope is %lf\n", distance, slope);

    return 0;
}
```

```
void print_point(const struct point *point) {
    printf("(%lf,%lf)", point->x, point->y);
}
```