

Lecture 13 Assignment (Structures)

```
1  #include <stdio.h> //print and scan
2  #include <math.h> // math equations (e.g. sqrt() )
3
4
5  struct line { //main struct of this program
6
7      struct point{ // contains x&y coordinates
8          float x;
9          float y;
10     }point1, point2;
11     float midpoint[2]; //storing 2 float data type for midpoint x,y
12     float slope; //slope of the line
13     float distance; //distance between 2 points
14 }line1; //line structure contains another structure named line1
15
16 float solveslope (struct line line1){ //function solving and calling slope
17
18     //declared variables under this function
19
20     float solve_slope; //store the result of division between numerator and denominator
21     float y_numerator;
22     float x_denominator;
23
24     y_numerator = line1.point1.y - line1.point2.y; //solve for (y1-y2)
25     x_denominator = line1.point1.x - line1.point2.x; //solve for (x1-x2)
26     solve_slope = y_numerator / x_denominator; //computing the value of solve_slope
27
28     return solve_slope; // returning the value
29 };
30
31 float solveMidpoint (struct line line1){ //function solving midpoint
32     //declared array index [0] and [1] which solves the midpoint given the formula [(x1+x2)/2] and [(y1+y2)/2]
33     line1.midpoint[0] = ((line1.point1.x + line1.point2.x) / 2);
34     line1.midpoint[1] = ((line1.point1.y + line1.point2.y) / 2);
35
36     printf("Midpoint = %.2lf , %.2lf", line1.midpoint[0], line1.midpoint[1]); //printing the values of the midpoint
37
38 };
39
40 float solveDistance(struct line line1){ //function for solving distance between 2 points
41     //sqrt( (x2 - x1)*(x2 - x1) + (y2 - y1)*(y2 - y1) );
42     //this code can be shorten using pow() under the pre-processor directive
43     line1.distance = sqrt( (line1.point2.x - line1.point1.x)*(line1.point2.x - line1.point1.x) + (line1.point2.y - line1.point1.y)*(line1.point2.y - line1.point1.y) );
44     printf("Distance between two points : %.2lf", line1.distance); //printing the values
45 };
46
47 void getSlopeInterceptForm(struct line line1){ //function for solving the intercept form
48     float b = line1.point1.y - ((solveslope(line1))*(line1.point1.x));
49     printf("y = %.2lfx + %.2lf", solveslope(line1), b); //printing of the result
50 };
51
52 //main function ; some parts are added to achieve readability of the program
53 int main(void){
54     printf("\n<<This program solves for the following:>> \n\n[a]Slope\n[b] Midpoint\n[c] Distance between two points\n[d]Slope- intercept form\n");
55     printf("\nPoint 1: Please enter x and y: "); //prompting user for entering values (x and y) for points 1 and 2
56     scanf("%f %f", &line1.point1.x , &line1.point1.y);
57
58     printf("\nPoint 2: Please enter x and y: ");
59     scanf("%f %f", &line1.point2.x , &line1.point2.y);
60
61     printf("\n");
62     printf("Slope: %.2lf", solveslope(line1)); //printing and calling the returned value above of the solve
63     printf("\n"); //improves readability of the result
64     solveMidpoint(line1); //calling the function declared
65     printf("\n");
66     solveDistance(line1); //calling solveMidpoint function
67     printf("\n");
68     getSlopeInterceptForm(line1); //calling the function
69 }
70
```