

```
1  #include <stdio.h>
2
3  // DEFINING STRUCT
4  struct MyData
5  {
6      int i;
7      float f;
8      double d;
9      char c;
10 };
11
12 int main(void)
13 {
14     //variable declarations
15
16     // 35 will be assigned to 'i' of 'data_one'
17     // 3.9 will be assigned to 'f' of 'data_one'
18     // 1.23765 will be assigned to 'd' of 'data_one'
19     // 'A' will be assigned to 'c' of 'data_one'
20     struct MyData data_one = { 35, 3.9f, 1.23765, 'A'}; //Inline Initialization
21
22     // 'P' will be assigned to 'i' of 'data_two' ... but 'P' is a character (char) ↗
23     // and 'i' is an integer ... so 'P' is converted into its decimal integer ASCII ↗
24     // value (80) and 80 is assigned to 'i' of data_two
25     // 6.2 will be assigned to 'f' of 'data_two'
26     // 12.199523 will be assigned to 'd' of 'data_two'
27     // 68 will be assigned to 'c' of 'data_two' ... but 68 is an integer (int) and ↗
28     // 'c' is a 'char' ... so 68 is considered as a decimal ASCII value and its ↗
29     // corresponding character ( 'D' ) is assigned to 'c' of data_two
30     struct MyData data_two = { 'P', 6.2f, 12.199523, 68 }; //Inline Initialization
31
32     // 36 will be assigned to 'i' of 'data_three'
33     // 'G' is 'char', but 'f' of 'data_three' is 'float'...hence, 'G' is converted ↗
34     // to its decimal integer ASCII value (71) and this in turn is converted to ↗
35     // 'float' (71.000000) and then it will be assigned to 'f' of 'data_three'
36     // 0.000000 will be assigned to 'd' of 'data_three'
37     // No character will be assigned to 'c' of 'data_three'
38     struct MyData data_three = { 36, 'G' }; //Inline Initialization
39
40     // 79 will be assigned to 'i' of 'data_four'
41     // 0.000000 will be assigned to 'f' of 'data_four'
42     // 0.000000 will be assigned to 'd' of 'data_four'
43     // No character will be assigned to 'c' of 'data_four'
44     struct MyData data_four = { 79 }; //Inline Initialization
45
46     //code
47     //Displaying Values Of The Data Members Of 'struct MyData'
48     printf("\n\n");
49     printf("DATA MEMBERS OF 'struct MyData data_one' ARE : \n\n");
50     printf("i = %d\n", data_one.i);
51     printf("f = %f\n", data_one.f);
52     printf("d = %lf\n", data_one.d);
```

```
47     printf("c = %c\n", data_one.c);
48
49     printf("\n\n");
50     printf("DATA MEMBERS OF 'struct MyData data_two' ARE : \n\n");
51     printf("i = %d\n", data_two.i);
52     printf("f = %f\n", data_two.f);
53     printf("d = %lf\n", data_two.d);
54     printf("c = %c\n", data_two.c);
55
56     printf("\n\n");
57     printf("DATA MEMBERS OF 'struct MyData data_three' ARE : \n\n");
58     printf("i = %d\n", data_three.i);
59     printf("f = %f\n", data_three.f);
60     printf("d = %lf\n", data_three.d);
61     printf("c = %c\n", data_three.c);
62
63     printf("\n\n");
64     printf("DATA MEMBERS OF 'struct MyData data_four' ARE : \n\n");
65     printf("i = %d\n", data_four.i);
66     printf("f = %f\n", data_four.f);
67     printf("d = %lf\n", data_four.d);
68     printf("c = %c\n", data_four.c);
69
70     return(0);
71 }
72
73
```