```
1 #include <stdio.h>
 2
 3 // DEFINING STRUCT
 4 struct MyData
 5 {
 6
        int i;
 7
        float f;
 8
        double d;
 9 };
10
11 int main(void)
12 {
        //variable declarations
13
14
        int i_size;
15
        int f_size;
        int d_size;
16
17
        int struct_MyData_size;
        int pointer_to_struct_MyData_size;
18
19
20
        struct MyData *pData = NULL;
21
        //code
22
23
        printf("\n\n");
24
25
        pData = (struct MyData *)malloc(sizeof(struct MyData));
26
        if (pData == NULL)
27
        {
28
            printf("FAILED TO ALLOCATE MEMORY TO 'sturct MyData' !!! EXITTING NOW ... →
              n'n;
29
            exit(0);
30
        }
31
        else
            printf("SUCCESSFULLY ALLOCATED MEMORY TO 'sturct MyData' !!!\n\n");
32
33
34
35
        //Assigning Data Values To The Data Members Of 'struct MyData'
36
        (*pData).i = 30;
        (*pData).f = 11.45f;
37
38
        (*pData).d = 1.2995;
39
40
        //Displaying Values Of The Data Members Of 'struct MyData'
41
        printf("\n\n");
        printf("DATA MEMBERS OF 'struct MyData' ARE : \n\n");
42
        printf("i = %d\n", (*pData).i);
43
        printf("f = %f\n", (*pData).f);
44
        printf("d = %lf\n", (*pData).d);
45
46
47
        //Calculating Sizes (In Bytes) Of The Data Members Of 'struct MyData'
48
        i_size = sizeof((*pData).i);
19
        f_size = sizeof((*pData).f);
50
        d_size = sizeof((*pData).d);
51
```

```
...ts\01-SimpleStructPointer\Method_01\SimpleStructPointer.c
```

```
2
```

```
//Displaying Sizes (In Bytes) Of The Data Members Of 'struct MyData'
53
        printf("\n\n");
        printf("SIZES (in bytes) OF DATA MEMBERS OF 'struct MyData' ARE : \n\n");
54
55
        printf("Size of 'i' = %d bytes\n", i_size);
        printf("Size of 'f' = %d bytes\n", f_size);
56
        printf("Size of 'd' = %d bytes\n", d_size);
57
58
59
        //Calculating Size (In Bytes) Of the entire 'struct Mydata'
60
        struct_MyData_size = sizeof(struct MyData);
61
        pointer_to_struct_MyData_size = sizeof(struct MyData *);
62
63
      //Displaying Sizes (In Bytes) Of the entire 'struct Mydata'
        printf("\n\n");
64
        printf("Size of 'struct MyData' : %d bytes\n\n", struct_MyData_size);
65
66
        printf("Size of pointer to 'struct MyData' : %d bytes\n\n",
          pointer_to_struct_MyData_size);
67
        if (pData)
68
69
        {
70
            free(pData);
71
           pData = NULL;
72
           printf("MEMORY ALLOCATED TO 'struct MyData' HAS BEEN SUCCESSFULLY
              FREED !!!\n\n");
73
        }
74
75
        return(0);
76 }
77
78
```