

## Pointers.c

```
1 //*****
2 // Name      Pointers.c
3 // Author    Computer Science, Pacific University
4 // Date
5 // Class     CS300
6 // Description Pointers, Pointers, and more Pointers
7 // NOTE      This code is used with the lecture "More Pointers" Students
8 //           will write and test each function from that set of slides.
9 //           Delete solutions and testing in main before committing what
10 //          students will download.
11 //*****
12
13 #include <stdio.h>
14 #include <stdlib.h>
15 #include <string.h>
16
17 typedef struct Person1
18 {
19     char szName[5];
20     int age;
21 } Person1;
22
23 typedef struct Person2
24 {
25     char *pszName;
26     int age;
27 } Person2;
28
29 //void myIntCpy (int *pIntDest, int *pIntSrc);
30 //void myStrCpy (char *pszDest, char *pszSrc);
31 void myMemCpy (void *pDest, void *pSrc, int size);
32
33 int main (void)
34 {
35     char szStr1[] = "hi", szStr2[] = "hit", szStr3[10];
36     int i = 5, j;
37     double x = 3.14, y;
38     char *pszStr = (char *) malloc (5);
39     Person1 sPerson1 = {"Jill", 10}, sPerson2;
40
41     puts("Pointers Start");
42
43     myMemCpy (szStr3, szStr1, strlen (szStr1) + 1);
44     puts (szStr3);
45
46     myMemCpy (szStr3, szStr2, strlen (szStr2) + 1);
47     puts (szStr3);
48
49     myMemCpy (&j, &i, sizeof (i));
50     printf ("%d\n", j);
51
52     myMemCpy (&y, &x, sizeof (double));
53     printf ("%8.2f\n", y);
54
55     myMemCpy (&szStr3, &szStr1, strlen (szStr1) + 1);
56     puts (szStr3);
```

```
57
58 myMemCpy (pszStr, &szStr1, strlen (szStr1) + 1);
59 puts (pszStr);
60
61 puts("Pointers End");
62
63 return EXIT_SUCCESS;
64 }
65
66 void myMemCpy (void *pDest, void *pSrc, int size)
67 {
68     int i;
69
70     for (i = 0; i < size; ++i)
71     {
72         *(char *) (pDest + i) = *(char *) (pSrc + i);
73     }
74 }
75
```