

```
1.  /**
2.   * scanf-0.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Reads a number from the user into an int.
8.   *
9.   * Demonstrates scanf and address-of operator.
10.  */
11.
12. #include <stdio.h>
13.
14. int main(void)
15. {
16.     int x;
17.     printf("Number please: ");
18.     scanf("%i", &x);
19.     printf("Thanks for the %i!\n", x);
20. }
```

```
1. /**
2.  * scanf-1.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Reads a string from the user into memory it shouldn't.
8.  *
9.  * Demonstrates possible attack!
10. */
11.
12. #include <stdio.h>
13.
14. int main(void)
15. {
16.     char* buffer;
17.     printf("String please: ");
18.     scanf("%s", buffer);
19.     printf("Thanks for the %s!\n", buffer);
20. }
```

```
1.  /**
2.   * scanf-2.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Reads a string from the user into an array (dangerously).
8.   *
9.   * Demonstrates potential buffer overflow!
10.  */
11.
12. #include <stdio.h>
13.
14. int main(void)
15. {
16.     char buffer[16];
17.     printf("String please: ");
18.     scanf("%s", buffer);
19.     printf("Thanks for the %s!\n", buffer);
20. }
```

```
1.  /**
2.   * structs-0.c
3.   *
4.   * Computer Science 50
5.   * David J. Malan
6.   *
7.   * Demonstrates use of structs.
8.   */
9.
10. #include <cs50.h>
11. #include <stdio.h>
12. #include <string.h>
13.
14. #include "structs.h"
15.
16. // number of students
17. #define STUDENTS 3
18.
19. int main(void)
20. {
21.     // declare students
22.     student students[STUDENTS];
23.
24.     // populate students with user's input
25.     for (int i = 0; i < STUDENTS; i++)
26.     {
27.         printf("Student's name: ");
28.         students[i].name = GetString();
29.
30.         printf("Student's house: ");
31.         students[i].house = GetString();
32.     }
33.
34.     // now print students
35.     for (int i = 0; i < STUDENTS; i++)
36.     {
37.         printf("%s is in %s.\n", students[i].name, students[i].house);
38.     }
39.
40.     // free memory
41.     for (int i = 0; i < STUDENTS; i++)
42.     {
43.         free(students[i].name);
44.         free(students[i].house);
45.     }
46. }
```

```
1.  /**
2.   * structs-1.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Demonstrates use of file I/O.
8.   */
9.
10. #include <cs50.h>
11. #include <stdio.h>
12. #include <stdlib.h>
13. #include <string.h>
14.
15. #include "structs.h"
16.
17. // number of students
18. #define STUDENTS 3
19.
20. int main(void)
21. {
22.     // declare students
23.     student students[STUDENTS];
24.
25.     // populate students with user's input
26.     for (int i = 0; i < STUDENTS; i++)
27.     {
28.         printf("Student's name: ");
29.         students[i].name = GetString();
30.
31.         printf("Student's house: ");
32.         students[i].house = GetString();
33.     }
34.
35.     // save students to disk
36.     FILE* file = fopen("students.csv", "w");
37.     if (file != NULL)
38.     {
39.         for (int i = 0; i < STUDENTS; i++)
40.         {
41.             fprintf(file, "%s,%s\n", students[i].name, students[i].house);
42.         }
43.         fclose(file);
44.     }
45.
46.     // free memory
47.     for (int i = 0; i < STUDENTS; i++)
48.     {
```

```
49.         free(students[i].name);
50.         free(students[i].house);
51.     }
52. }
```

```
1.  /**
2.   * structs.h
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Defines a student.
8.   */
9.
10. #include <cs50.h>
11.
12. // structure representing a student
13. typedef struct
14. {
15.     string name;
16.     string house;
17. }
18. student;
```