

```
1. /**
2.  * cough-0.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Coughs three times.
8.  *
9.  * Demonstrates suboptimal design (and coughing).
10. */
11.
12. #include <stdio.h>
13.
14. int main(void)
15. {
16.     // cough three times
17.     printf("cough\n");
18.     printf("cough\n");
19.     printf("cough\n");
20. }
```

```
1. /**
2.  * cough-1.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Coughs three times.
8.  *
9.  * Demonstrates better design via a loop.
10. */
11.
12. #include <stdio.h>
13.
14. int main(void)
15. {
16.     // cough three times
17.     for (int i = 0; i < 3; i++)
18.     {
19.         printf("cough\n");
20.     }
21. }
```

```
1. /**
2.  * cough-2.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Coughs three times.
8.  *
9.  * Demonstrates abstraction and hierarchical decomposition.
10. */
11.
12. #include <stdio.h>
13.
14. // prototype
15. void cough(void);
16.
17. int main(void)
18. {
19.     // cough three times
20.     for (int i = 0; i < 3; i++)
21.     {
22.         cough();
23.     }
24. }
25.
26. /**
27.  * Coughs once.
28.  */
29. void cough(void)
30. {
31.     printf("cough\n");
32. }
```

```
1. /**
2.  * cough-3.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Coughs three times.
8.  *
9.  * Demonstrates parameterization.
10. */
11.
12. #include <stdio.h>
13.
14. // prototype
15. void cough(int n);
16.
17. int main(void)
18. {
19.     // cough three times
20.     cough(3);
21. }
22.
23. /**
24.  * Coughs n times.
25.  */
26. void cough(int n)
27. {
28.     for (int i = 0; i < n; i++)
29.     {
30.         printf("cough\n");
31.     }
32. }
```

```
1. /**
2.  * cough-4.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Coughs three times and sneezes three times.
8.  *
9.  * Demonstrates further abstraction.
10. */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15. // prototypes
16. void cough(int n);
17. void say(string word, int n);
18. void sneeze(int n);
19.
20. int main(void)
21. {
22.     // cough three times
23.     cough(3);
24.
25.     // sneeze three times
26.     sneeze(3);
27.
28. }
29.
30. /**
31.  * Coughs n times.
32.  */
33. void cough(int n)
34. {
35.     say("cough", n);
36. }
37.
38. /**
39.  * Says word n times.
40.  */
41. void say(string word, int n)
42. {
43.     for (int i = 0; i < n; i++)
44.     {
45.         printf("%s\n", word);
46.     }
47. }
48.
```

```
49. /**
50.  * Sneezes n times.
51.  */
52. void sneeze(int n)
53. {
54.     say("achoo", n);
55. }
```

```
1. /**
2.  * f2c.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Converts Fahrenheit to Celsius.
8.  *
9.  * Demonstrates arithmetic.
10. */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15. int main(void)
16. {
17.     // ask user user for temperature in Fahrenheit
18.     printf("Temperature in F: ");
19.     float f = GetFloat();
20.
21.     // convert F to C
22.     float c = 5.0 / 9.0 * (f - 32.0);
23.
24.     // display result to one decimal place
25.     printf("%.1f\n", c);
26. }
```

```
1. /**
2.  * function-0.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Prints a user's name.
8.  *
9.  * Demonstrates a function (not from a library) with a side effect.
10. */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15. // prototype
16. void PrintName(string name);
17.
18. int main(void)
19. {
20.     printf("Your name: ");
21.     string s = GetString();
22.     PrintName(s);
23. }
24.
25. /**
26.  * Says hello to someone by name.
27.  */
28. void PrintName(string name)
29. {
30.     printf("hello, %s\n", name);
31. }
```



```
1. /**
2.  * function-1.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Demands that user provide a positive integer.
8.  *
9.  * Demonstrates use of a function (not from a library) with a return value.
10. */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15. // prototype
16. int GetPositiveInt();
17.
18. int main(void)
19. {
20.     int n = GetPositiveInt();
21.     printf("Thanks for the %i!\n", n);
22. }
23.
24. /**
25.  * Gets a positive integer from a user.
26.  */
27. int GetPositiveInt(void)
28. {
29.     int n;
30.     do
31.     {
32.         printf("Please give me a positive int: ");
33.         n = GetInt();
34.     }
35.     while (n < 1);
36.     return n;
37. }
```

```
1. /**
2.  * imprecision.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Divides one floating-point value by another.
8.  *
9.  * Demonstrates imprecision of floating-point values.
10. */
11.
12. #include <stdio.h>
13.
14. int main(void)
15. {
16.     printf("%.29f\n", 1.0 / 10.0);
17. }
```

```
1. /*****
2.  * return.c
3.  *
4.  * David J. Malan
5.  * malan@harvard.edu
6.  *
7.  * Cubes a variable.
8.  *
9.  * Demonstrates use of parameter and return value.
10. *****/
11.
12. #include <stdio.h>
13.
14. // function prototype
15. int cube(int a);
16.
17. int main(void)
18. {
19.     int x = 2;
20.     printf("x is now %i\n", x);
21.     printf("Cubing...\n");
22.     x = cube(x);
23.     printf("Cubed!\n");
24.     printf("x is now %i\n", x);
25. }
26.
27. /**
28.  * Cubes argument.
29.  */
30. int cube(int n)
31. {
32.     return n * n * n;
33. }
```

```
1.  /**
2.   * switch.c
3.   *
4.   * David J. Malan
5.   * malan@harvard.edu
6.   *
7.   * Assesses the size of user's input.
8.   *
9.   * Demonstrates use of a switch.
10.  */
11.
12. #include <cs50.h>
13. #include <stdio.h>
14.
15. int main(void)
16. {
17.     // ask user for an integer
18.     printf("Give me an integer between 1 and 10: ");
19.     int n = GetInt();
20.
21.     // judge user's input
22.     switch (n)
23.     {
24.         case 1:
25.         case 2:
26.         case 3:
27.             printf("You picked a small number.\n");
28.             break;
29.
30.         case 4:
31.         case 5:
32.         case 6:
33.             printf("You picked a medium number.\n");
34.             break;
35.
36.         case 7:
37.         case 8:
38.         case 9:
39.         case 10:
40.             printf("You picked a big number.\n");
41.             break;
42.
43.         default:
44.             printf("You picked an invalid number.\n");
45.             break;
46.     }
47. }
```