

```
1 // Logical operators
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Prompt user for answer
9     char c = get_char("Answer: ");
10
11     // Check answer
12     if (c == 'Y' || c == 'y')
13     {
14         printf("yes\n");
15     }
16     else if (c == 'N' || c == 'n')
17     {
18         printf("no\n");
19     }
20 }
```

```
1 // Conditions and relational operators
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Prompt user for x
9     int x = get_int("x: ");
10
11     // Prompt user for y
12     int y = get_int("y: ");
13
14     // Compare x and y
15     if (x < y)
16     {
17         printf("x is less than y\n");
18     }
19     else if (x > y)
20     {
21         printf("x is greater than y\n");
22     }
23     else
24     {
25         printf("x is equal to y\n");
26     }
27 }
```

```
1 // Opportunity for better design
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     printf("cough\n");
8     printf("cough\n");
9     printf("cough\n");
10 }
```

```
1 // Better design
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     for (int i = 0; i < 3; i++)
8     {
9         printf("cough\n");
10    }
11 }
```

```
1 // Abstraction
2
3 #include <stdio.h>
4
5 void cough(void);
6
7 int main(void)
8 {
9     for (int i = 0; i < 3; i++)
10    {
11        cough();
12    }
13 }
14
15 // Cough once
16 void cough(void)
17 {
18     printf("cough\n");
19 }
```

```
1 // Abstraction with parameterization
2
3 #include <stdio.h>
4
5 void cough(int n);
6
7 int main(void)
8 {
9     cough(3);
10 }
11
12 // Cough some number of times
13 void cough(int n)
14 {
15     for (int i = 0; i < n; i++)
16     {
17         printf("cough\n");
18     }
19 }
```

```
1 // Floating-point arithmetic with double
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Prompt user for x
9     double x = get_double("x: ");
10
11     // Prompt user for y
12     double y = get_double("y: ");
13
14     // Perform division
15     printf("x / y = %.50f\n", x / y);
16 }
```

```
1 // get_float and printf with %f
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     float f = get_float("Float: ");
9     printf("hello, %f\n", f);
10 }
```



```
1 // Floating-point arithmetic with float
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Prompt user for x
9     float x = get_float("x: ");
10
11     // Prompt user for y
12     float y = get_float("y: ");
13
14     // Perform division
15     printf("x / y = %.50f\n", x / y);
16 }
```

```
1 // A program
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     printf("hello, world\n");
8 }
```

```
1 // get_int and printf with %i
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     int i = get_int("Integer: ");
9     printf("hello, %i\n", i);
10 }
```

```
1 // Integer arithmetic
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Prompt user for x
9     int x = get_int("x: ");
10
11     // Prompt user for y
12     int y = get_int("y: ");
13
14     // Perform arithmetic
15     printf("x + y = %i\n", x + y);
16     printf("x - y = %i\n", x - y);
17     printf("x * y = %i\n", x * y);
18     printf("x / y = %i\n", x / y);
19     printf("x mod y = %i\n", x % y);
20 }
```

```
1 // Integer overflow
2
3 #include <stdio.h>
4 #include <unistd.h>
5
6 int main(void)
7 {
8     // Iteratively double i
9     for (int i = 1; ; i *= 2)
10    {
11        printf("%i\n", i);
12        sleep(1);
13    }
14 }
```

```
1 // Remainder operation
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Prompt user for integer
9     int n = get_int("n: ");
10
11     // Check parity of integer
12     if (n % 2 == 0)
13     {
14         printf("even\n");
15     }
16     else
17     {
18         printf("odd\n");
19     }
20 }
```

```
1 // Abstraction and scope
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int get_positive_int(string prompt);
7
8 int main(void)
9 {
10     int i = get_positive_int("Positive integer: ");
11     printf("%i\n", i);
12 }
13
14 // Prompt user for positive integer
15 int get_positive_int(string prompt)
16 {
17     int n;
18     do
19     {
20         n = get_int("%s", prompt);
21     }
22     while (n < 1);
23     return n;
24 }
```

```
1 // get_string and printf with %s
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     string s = get_string("Name: ");
9     printf("hello, %s\n", s);
10 }
```