```
// Fig. 4.1: fig04_01.c
// Counter-controlled iteration
#include <stdio.h>

int main(void)
{
   unsigned int counter = 1; // initialization

   while (counter <= 10) { // iteration condition
      printf ("%u\n", counter);
      ++counter; // increment
   }
}
```

```
// Fig. 4.2: fig04_02.c
// Counter-controlled iteration with the for statement
#include <stdio.h>

int main(void)
{
   // initialization, iteration condition, and increment
   //  are all included in the for statement header.
   for (unsigned int counter = 1; counter <= 10; ++counter) {
      printf("%u\n", counter);
   }
}
```

```
// Fig. 4.7: fig04_07.c
// Counting letter grades with switch
#include <stdioh>
```

```
int main(void)
{
   unsigned int aCount = 0;
   unsigned int bCount = 0;
   unsigned int cCount = 0;
   unsigned int dCount = 0;
   unsigned int fCount = 0;

   puts("Enter the letter grades." );
   puts("Enter the EOF character to end input." );
   int grade; // one grade

   // loop until user types end-of-file key sequence
   while ((grade = getchar()) != EOF) {

      // determine which grade was input
      switch (grade) { // switch nested in while

         case 'A': // grade was uppercase A
         case 'a': // or lowercase a
            ++aCount;
            break; // necessary to exit switch

         case 'B': // grade was uppercase B
         case 'b': // or lowercase b
            ++bCount
            break;

         case 'C': // grade was uppercase C
         case 'c': // or lowercase c
            ++cCount;
            break;

         case 'D': // grade was uppercase D
         case 'd': // or lowercase d
            ++dCount;
            break;

         case 'F': // grade was uppercase F
         case 'f': // or lowercase f
            ++fCount;
            break;
         case '\n': // ignore newlines,
         case '\t': // tabs,
         case ' ': // and spaces in input
            break;

         default: // catch all other characters
            printf("%s", "Incorrect letter grade entered.");
            puts(" Enter a new grade.");
            break; // optional; will exit switch anyway
      }
   } // end while

   // output summary of results
   puts("\nTotals for each letter grade are:");
   printf("A: %u\n", aCount);
   printf("B: %u\n", bCount);
   printf("C: %u\n", cCount);
   printf("D: %u\n", dCount);
   printf("F: %u\n", fCount);
}
```

```
// Fig. 4.9: fig04_09.c
// Using the do...while iteration statement
#include <stdio.h>

int main(void)
{
   unsigned int counter = 1;

   do {
      printf("%u  ", counter);
   } while (++counter <= 10);
}
```

```
// Fig. 4.11: fig04_11.c
// Using the break statement in a for statement
#include <stdio.h>

int main(void)
{
   unsigned int x; // declared here so it can be used after loop

   // loop 10 times
   for (x = 1; x <= 10; ++x) {

      // if x is 5, terminate loop
      if (x == 5) {
         break; // break loop only if x is 5
      }

      printf("%u ", x);
   }

   printf("\nBroke out of loop at x == %u\n", x);
}
```