The Structure of `totaler` Statements

The `totaler` construct allows the programmer to keep running totals of various types in a program. It can be used to add or subtract a preset value or a calculated value to another final variable. The first totaler we have used is the ++ and -- operator. Notice that in all the examples of totalers, the variable is **initialized** before it is used. This is crucial for the totaler to behave as expected.

```
int total = 0;                      // Initialized to zero
total++;                            // This adds 1 to total
total = total + 1;                  // This also adds 1 to total


int total = 20;                     // Initialized to 20 in this case
total--;                            // This subtracts 1 from total
total = total - 1;                  // This also subtracts 1 from total
```

Fancier `totaler` statements can be used to add or subtract numbers larger than 1 as follows:

```
int total = 10;
total+=5;                           // This adds 5 to total
total = total + 5;                  // This also adds 5 to total

int total = 20;
total-=5;                           // This subtracts 5 from total
total = total - 5;                  // This also subtracts 5 from total

int total = 1;
total*=2;                           // This multiplies total by 2
total = total * 2;                  // This also multiplies total by 2
```

A loop can make this very useful when coding. Say we want to add the first 10 numbers together:

```
int total = 0;                      // You must initialize totalers
for(int i = 1; i <= 10; i++)        // for them to work as expected
{
  total = total + i;
}
cout << "The sum of the numbers from 1 to 10 is: " << total;
```

or say we want to add the first 10 *even* numbers together:

```
int total = 2;                      //Note fancy initialization!!
for(int i = 4; i <= 20; i=i+2)      //Note this goes up by 2 each time
{
  total = total + i;
}
cout << "The sum of the first 10 even numbers: " << total;
```