C Programming

Arithmetic Operators

What are they?

+

=

/

-

*

Those ones behave as you would expect (almost)

Examples

```
float price = 9.30;
price = price + 4.8;
price = price * 3;
price = price - 4.2;
```

Almost???

```
int price = 8;
price = price / 3;

// price is 2, not 2.6667
```

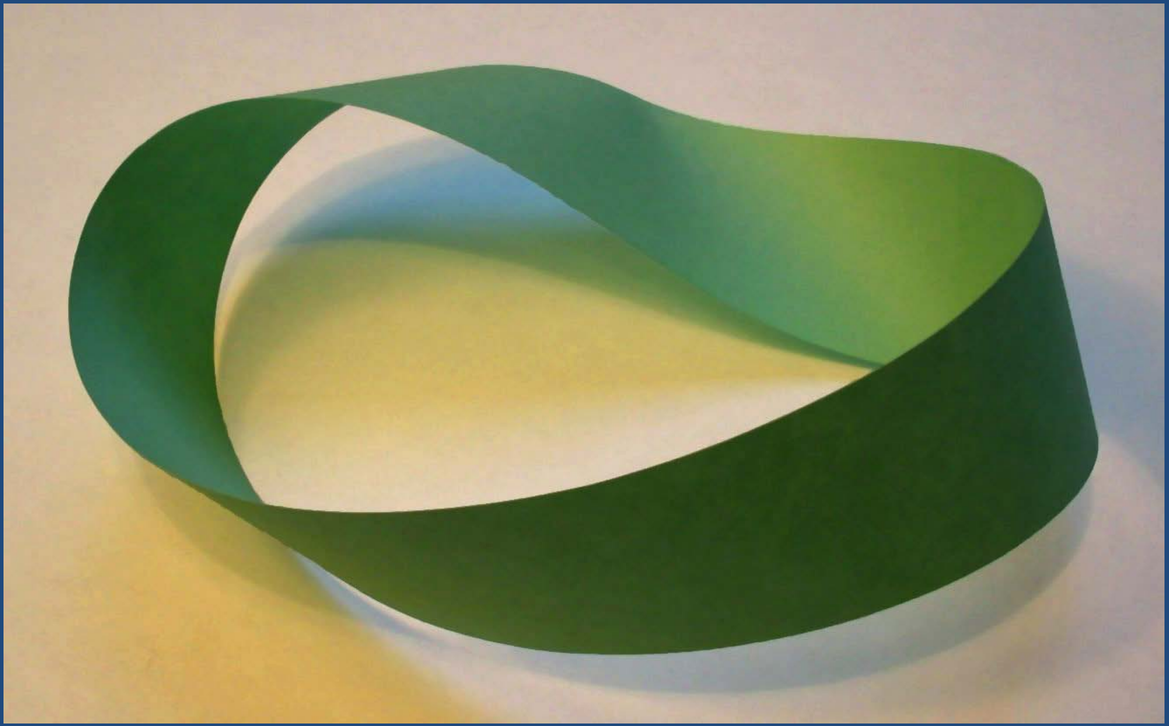
LOST
CONFUSED
UNSURE
UNCLEAR
PERPLEXED
DISORIENTED
BEWILDERED

Integer arithmetic truncates fractions

(you'll see this in Assignment #1)

And "Almost" again???

Data types can cause wrapping



e.g. if a is a char variable with value 127 and you add 1 to it, you do **not** get 128!

- The value wraps around to -128.

Watch for this in assignment #1.

Pre-increment and Post-increment

++

(add 1 to a variable)

++a is different from a++

… sometimes …

If on a line by itself, the two versions behave the same

Example of Pre-increment and Post-increment, combining with printf()

```
int a = 8;
printf("%d", a);       8
printf("%d", ++a);     9
printf("%d", a);       9
printf("%d", a++);     9
printf("%d", a);       10
```

```
int a = 8;
printf("%d", a);       8
printf("%d", a++);     8
printf("%d", a);       9
printf("%d", ++a);     10
printf("%d", a);       10
```

Also note that we've put an arithmetic expression in a printf() argument.

--

behaves the same way

+=

allows for increasing a variable by more than 1

```
a = a + 8;
```

$\Downarrow$

```
a += 8;
```

and …

+=

*=

-=

and more

/=

And one you may not have heard of ...

%

is modulo

or remainder

e.g. If b = 15, b % 12 produces 3 because 3 is left over when you divide 15 by 12.

Most obvious applications

12-hour or 24-hour time

converting minutes to hours and minutes

- Remember this for the airline assignment (#4)

Summary

1. A lot of the usual arithmetic operators work as you'd expect.
2. But some don't.
3. And some are new.