CS1007 lecture #5 notes

tue 5 feb 2002

- news
- mathematical operators
- logical operators
- truth tables
- relational operators
- the `if` branching statement
- flowcharts
- reading: *ch 3.1-3.5*

news.

- the LAST day to CHANGE or SIGN UP for a RECITATION is this Friday Feb 8 by 6AM
- here's what you do:
  - go to the "human help" link on the class web page
  - find a recitation that you can attend and get the TA's email address
  - if you are NEW and first signing up for a recitation,
    send email to the TA and CC me
  - if you are CHANGING recitation,
    send email to BOTH the old TA and the new TA, and CC me

mathematical operators.

+	unary plus
−	unary minus
+	addition
−	subtraction
*	multiplication
/	division
%	modulo

example:

```
int x, y;
x = -5;
y = x * 7;
y = y + 3;
x = x * -2;
y = x / 19;
```

what are `x` and `y` equal to?

modulo means “remainder after integer division”

coercion or type casting.

- remember from last time: data of type `char` is stored as a number — which is really an index into the ASCII table
- a declaration like this:

```
char y = 'A';
```

  really stores a 65 (the ASCII value of 'A') in a memory location that is labeled `y`
- you can do math on that 65 by *coercing* (aka *type casting*) the `char` to an `int`
- for example:

```
char y = 'A';    // initialize variable y to store an A
int  x = (int)y; // initialize variable x to store 65
x = x + 1;       // increment x (to 66)
y = (char)x;     // coerce x from an int to a char ('B')
```

boolean expressions.

- boolean variables: `true` (1) or `false` (0)
- logical operators:

!	not
&&	and
||	or

example:

```
boolean a, b;
x = 1; // true
y = 0; // false
System.out.println( "x && y is false" );
System.out.println( "x || y is true" );
System.out.println( "x && !y is true" );
```

truth tables.

a	!a
false	true
true	false

a	b	a && b
true	true	true
true	false	false
false	true	false
false	false	false

a	b	a || b
true	true	true
true	false	true
false	true	true
false	false	false

relational operators.

==	equality
!=	inequality
>	greater than
<	less than
>=	greater than or equal to
<=	Less than or equal to

example:

```
int x, y;
x = -5;
y = 7;
```

some truths:

(x < y)	true
(x == y)	false
(x >= y)	false

the `if` branching statement.

```
if ( x < y ) {
  x = y;
}
```

```
if ( x < y ) {
  x = y;
}
else {
  x = 91;
}
```