Lecture 9: Conditionals Worksheet **Solutions**

1) What is the **value** returned by the following expressions?

`(2 + 5) * (6 - 4)`	**__14__**	`2 + 5 * 8`	**__42__**
`max(1/2, min(-1*3, 1-3))`	**__0.5__**	`9 / 3 > 1 + 2`	**_false_**
`0.5 != -0.5`	**_true_**	`'h' == 'H'`	**_false_**

2) What is the **data type** of the value returned by the following expressions? Assume the variables `x` and `y` are defined as `float`. Check the Processing Reference (online) for functions that you don't know.

`sqrt(x)`	**_float_**	`abs(x) != sq(y)`	**boolean**
`!false`	**boolean**	`floor(0.1 * y)`	**__int__**

3) The **modulus** operator (`x % y`) returns the *remainder* of `x` divided by `y`. What value is returned by the following expressions?

`0 % 3`	**__0__**	`6 % 3`	**__0__**	`-2 % 3`	**_-2__**
`2 % 3`	**__2__**	`8 % 3`	**__2__**	`-4 % 3`	**_-1__**
`4 % 3`	**__1___**	`10 % 3`	**__1__**	`-6 % 3`	**__0__**

4) Type the following into Processing and press Play. Explain what you see.

```
void draw() {
  background(0, frameCount % 255, 0);
}
```

The frame count increments the green component of the background by 1 each time `draw()` is run (every 1/60 sec), but the modulus cycles it back from fully green to black.

5) Fill in the following *truth tables* for the logical operators given `boolean x` and `y`:

NOT (`!`)

x	!x
false	true
true	false

OR (`||`)

x	y	x || y
false	false	false
false	true	true
true	false	true
true	true	true

AND (`&&`)

x	y	x && y
false	false	false
false	true	false
true	false	false
true	true	true

6) What is the **value** returned by the following expressions?

`true || false`	**_true_**	`true && true && false`	**_false_**
`!(true == false)`	**_true_**	`(3 >= 1) && (3 < 10))`	**_true_**

For the following questions, we will use *static* Processing code (*i.e.* no `setup()` or `draw()`). Start a new Processing file and add the following code. Make sure that the canvas is blue when you press Play.

```
int x = 120;
if ( x > 0 ) {
  background(0, 0, 255);
}
```

7) Change the initial value of `x` in your code so that the canvas no longer turns blue. What value of `x` did you use and what color is the canvas now?

   `x`: _anything negative_ canvas color: _light gray (the default color)_

8) Now add an `else` clause after the `if` clause so that the canvas turns **red** instead of the color you saw in Question 7. Press Play to verify that it works now (changing `x` back to 120 should revert the canvas to blue).

```
if ( x > 0 ) {
  background(0, 0, 255);
} else {
  background(255, 0, 0);
}
```

9) Now add an `else if` clause *in-between* the `if` and `else` clauses that turns the canvas **green** if `x` is less than or equal to `-2`. Predict what color the canvas will be for the following values of `x` and then verify in Processing:

```
if ( x > 0 ) {
  background(0, 0, 255);
} else if ( x <= -2 ) {
  background(0, 255, 0);
} else {
  background(255, 0, 0);
}
```

x	canvas color
-3	green
-2	green
-1	red
0	red
1	blue