```
function Ghost(color, x, y){
//if the user declares a ghost with coordinates outside of the 16x16, it will place it where it would
//have wrapped. It will add 1 to the coordinates if they are above 16 because the 16th coordinate
//exists but zeroth does not, so (x%32) would be zero and anything after that will be 1 off. So
//adding 1 takes care of that issue. 17 & 32 will both return 1.

  while (x > 16){
    if (x > 31){
      x++; }
    x = (x%16);
}
  while (y > 16){
    if (y > 31){
      y++; }
    y = (y%16);
  }
  this.ghostColor = color;
  this.cordX = x;
  this.cordY = y;
}
Ghost.prototype.up = function () {
  this.cordY++;
  if(this.cordY == 17){
      this.cordY = 1; }
}
Ghost.prototype.down = function () {
  this.cordY = (this.cordY - 1);
  if(this.cordY == 0){
      this.cordY = 16; }
}
Ghost.prototype.left = function () {
  this.cordX = (this.cordX - 1);
  if(this.cordX == 0){
      this.cordX = 16; }
}
Ghost.prototype.right = function () {
  this.cordX++;
  if(this.cordX == 17){
      this.cordX = 1; }
}
Ghost.prototype.toString = function () {
  return ("The " + this.ghostColor + " ghost is at (" + this.cordX + ", " + this.cordY + ")"); }
```

```
> var yellow = new Ghost ("yellow", 1, 1)
< undefined
> yellow.toString()
< "The yellow ghost is at (1, 1)"
> yellow.down()
< undefined
> yellow.toString()
< "The yellow ghost is at (1, 16)"
> yellow.left()
< undefined
> yellow.toString()
< "The yellow ghost is at (16, 16)"
> yellow.right()
< undefined
> yellow.toString()
< "The yellow ghost is at (1, 16)"
> yellow.up()
< undefined
> yellow.toString()
< "The yellow ghost is at (1, 1)"
```