JitterBug bug;

JitterBug jit; Declaring objects

float c=1; Set floating global variable

void setup(){ Setup Shell open bracket

size(600,600); Set window/console size(in pixels)

noStroke(); no borders around shapes

colorMode(HSB); Set color mode to HSB

bug = new JitterBug(width\*0.33,height/2,50,c); Create bug, a new copy of the jitterbug class

jit = new JitterBug(width\*0.66,height/2,10,c); Create jit, a new copy of the jitterbug class

} Setup Shell closed bracket

void draw(){ Draw the following within the brackets

if(c>=255) c=1; else c++; If c is greater than or equal to 255, reset c to 1; otherwise, add one to c for every loop

jit.move(); apply ‘move’ to ‘jit’

jit.display(); apply ‘display’ to ‘jit’

bug.move(); apply ‘move’ to ‘bug’

bug.display(); apply ‘display’ to ‘bug’

frameRate(c); make the framerate equal to c

} Draw function closed bracket

class JitterBug { Define Jitterbug class

float x; Creates local floats X and Y

float y;

int diameter; Initializes local diameter

float speed = 5; Creates local float speed, equal to 5

JitterBug(float tempX, float tempY,int tempDiameter,float c){Criteria for the Jitterbug Class

x=tempX Sets tempX equal to x

y=tempY; Sets tempY equal to y

diameter=tempDiameter; Sets tempDiameter equal to diameter

} End

void move(){ Define Method ‘move’

x+=random(-speed,speed); Add a random value between +- speed to x

y+=random(-speed,speed); Add a random value between +- speed to y

} End defining of method ‘move’

void display(){ Define Method ‘display’

fill(c,255,255); Fill with color with c, 255, 255

ellipse(x,y,diameter,diameter); Create ellipse at x y, with ‘diamter’ height and width

}} End define Method ‘move’ and jitterbug Class