

- 1 Write the following code fragment as a **for** loop.

(1)

```
int i = 0;
while (i < 10) {
    ellipse(i, i, i, i);
    i++;
}
```

- 2 Write the following code fragment as a **while** loop.

(1)

```
for (int i = 0; i < width; i = i + 5)
    line(i, 0, i, height);
```

- 3 A common error beginning programmers face with the **for**-loop is illustrated in the code segment below:

(2)

```
int i = 0;
while (i < 100); {
    line(i, 0, i, height);
    i = i + 5;
}
```

- Identify the error in the code above.
- This type of error is actually a *semantic* error. That is, Processing accepts the code as valid; however, it will not do what appears to be intended by the programmer. What happens when the above code fragment is run and why?

- 4 Use nested loops to create a grid of Processing Bees based on your work on the Processing Bee program from Assignment #3.

(2)

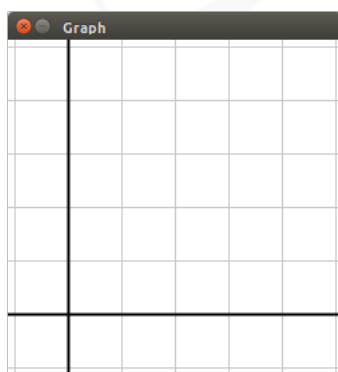
**Note:** Your program should include as many bees as possible based on the **width** and **height** of the drawing canvas. You may include some padding between consecutive bees for aesthetic reasons if you wish.

- 5 Redesign your **Dartboard** program from Assignment #3 to use loops wherever possible in order to drastically reduce the number of individual lines of code.

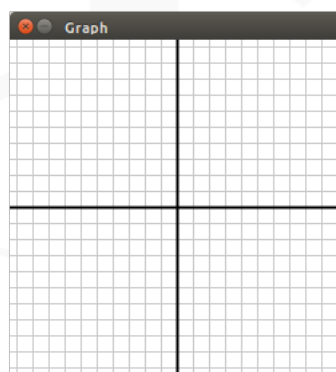
(4)

- 6 Although the Processing drawing canvas has a pixel coordinate statement wherein the top-left coordinate is always (0, 0) and the bottom-right coordinate is always (**width**, **height**), it is often important to offer different "views" of data, requiring a transformation of the coordinate system. Implement a program that will store values for **xMin**, **xMax**, **xStep**, **yMin**, **yMax**, and **yStep** and produces a coordinate grid, including coordinate axis. This will allow us to produce a "window" system similar to many popular graphing calculators. See below for a number of examples.

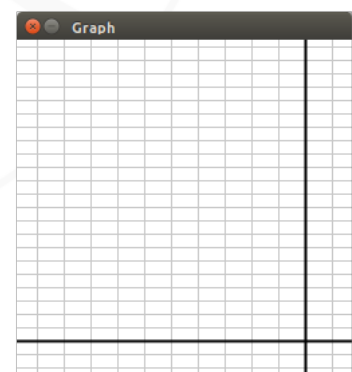
(6)



<b>xMin</b>	-5	<b>yMin</b>	-5
<b>xMax</b>	25	<b>yMax</b>	25
<b>xStep</b>	5	<b>yStep</b>	5



<b>xMin</b>	-10	<b>yMin</b>	-10
<b>xMax</b>	10	<b>yMax</b>	10
<b>xStep</b>	1	<b>yStep</b>	1



<b>xMin</b>	-100	<b>yMin</b>	0
<b>xMax</b>	10	<b>yMax</b>	100
<b>xStep</b>	10	<b>yStep</b>	5