1. Circle all of the statements that are true.
   
   a) A variable stores a value into the computer's memory so that it can be used later in the program.
   b) A single variable can be used many times.
   c) The value that is stored in a variable can change over time.
   d) All of the above.

2. Below are several variable declarations in processing. Each of the declarations uses mathematical expressions to determine the value that the variable will be set to. Determine what value will be in each variable.

   int big_num = 8 / 4 + 2 - 10 + 100;

   int coordinate = (2 - (6 + 4) * 2) / 2 + 3;

   int medium_num = big_num * 10 * 4 + 2 / 4;

   int coord2 = coordinate + big_num - medium_num * 2;
3. Below are a few small programs that use variables. For each one, draw a picture with pen/pencil/marker to determine what the canvas will look like, and indicate what color the various parts of the canvas will be. After doing this, you can try running the code to see if you were correct!

```cpp
size(300, 300);
int valueA = 250;
rect(25, 25, valueA, valueA);
int valueB = valueA / 5;
valueB = valueB * 5 + 50;
ellipse(150, 150, valueB, valueA);

size(400, 300);
int valueA = 20;
rect(valueA * 2, valueA * 5, 320, 100);
rect(150, 40, valueA * 5, valueA * 11);
int x = valueA - 10;
ellipse(x * 20, x * 15, 40, 40);

size(300, 300);
int a = 10;
int b = 3;
int c = 25;
int location = a * b * 5;
ellipse(location, location, 200, 200);
line(c, c, c * 11, c * 11);
line(c, c * 11, c * 11, c);
```

(Section 4 problems are continued on the class webpage.)
4. In this problem, you will write a processing program that uses your mouse position. You should write a program that draws four circles around your mouse, as shown in the picture. The circles should follow your mouse around the canvas.

![Image of four circles around a mouse cursor](image)

5. (A) Write a processing program that looks like the processing canvas below. You should use only one `rect()` function in your program. Use a for-loop to repeat the shape.

![Image of a row of rectangles](image)

(B) Modify the program from part A to make the canvas look like the canvas below. Again, you can only use one call to `rect()` and must use a loop.

![Image of a colored row of rectangles](image)
(C) Modify the program from part A to make the canvas look like the canvas below. The same rules apply.

(D) Modify the program from part A to make the canvas look like the canvas below. The same rules apply.