Previous projects

1. Complete application, sequential processing programmed with straight line code, programming artistic graphic designs by thinking about a Turtle's location on a World and how a Turtle's methods change its location.
2. Everything from Project 01 plus additions:
   - Making and calling your own methods you add to the Turtle class,
   - making and calling your own methods with two or more parameters,
   - variables, belonging to your methods, that are parameters,
   - variables used for loop control
     the syntax
     ```
     while ( test ) { body statement or statements }
     ```
     and loop pattern
     ```
     int howManyMore;
     howManyMore = HOW_MANY;
     while( howManyMore > 0 )
     {
     //code to do something ONCE
     //(the first time or again)
     howManyMore = howManyMore - 1;
     }
     ```
   - variables used to compute numbers from previously computed numbers,
   - in addition to sequential processing programmed by straight line code inside the main method---
   - sequential processing programmed with loops inside the main method,
   - programming with method calls inside the main method,
   - programming with straight line code inside your own methods
   - "casting" a numeric value that was best computed in double precision floating point arithmetic into an integer (whole number) value because an existing software interface (G&E's Turtle's forward(int distance)) required an int type of parameter.

Now here is what you will do in Project 03:

You will, by first analyzing textbook, lecture and lab solutions, and then trial, error, debugging, testing, observing, more textbook studying, more trials, etc., and then, final test and software release for grading, develop your own Picture manipulations, one that can be done with a Turtle living in the Picture and the others are variations of the manipulations explained in Chapter 4 of G&E's textbook. A total of four (4) variations are assigned for you to do for full credit. (Extra credit will be given for work that demonstrates mastery of the new topics from Chapter 5.)

You will "encapsulate" every single added Picture manipulation into a new method you make yourself and add to the Picture class.

The first of these methods must, just like in the book, manipulate every Pixel in the Picture. It will therefore manipulate the whole Picture, exactly how the textbook's examples do.
The first method you add to the **Picture** class must, similar Program 10, page 114 of the book, have the name and parameters given exactly by:

```java
public void changeWhole( double amount )
{
    //Write loopy code to manipulate "this" Picture here.
    //The value of amount MUST control the amount of the effect, so
    //0.0 makes the smallest amount and 1.0 the maximum amount.
}
```

But it MUST NOT JUST BE A COPY OF `changeRed`! Pick any of the ideas of what can be manipulated, or a combination of them, from Chapter 4, and program something that's not exactly the same as the book's or other examples you have seen.

The second method you add to the **Picture** class must have the name, parameters, and structure exactly this:

```java
public boolean scribble( int xPos, int yPos, double scale )
{
    Turtle tu = new Turtle( this );
    //this refers to the Picture on, for or with scribble() is called.
    //A GE Turtle can live in a Picture as well as a World object.
    tu.penUp( );
    tu.moveTo( xPos, yPos );
    //Purpose of the last 2 statements:
    //They make the Turtle start at location (xPos, yPos) without
    //making any marks on the Picture.

    //Your code to program making the scribble goes here.
    //Of course, it must call tu.penDown( ) at least once!
    //And, it CAN use tu.moveTo(), not just tu.forward().
    //Finally, the size of the scribble MUST be controlled by the
    //value of scale.

    return true;  //Return may be done from elsewhere, and the value
                   //might be false if you do the extra credit version.
}
```

Pick any ideas from the course or you think up of, but do not program something that is the same as any code you have seen before. The objective is to improve (in your opinion!) what your method scribbles onto a Picture until you are have a result that of course works, and you are happy with!