Start a project and import JavaGUI.java from this folder.

* When you run the program, there are four buttons
  +  - draw a randomly colored dot on the screen in a random location.
  +  - draws five of them.
  +  - this button doesn’t do anything yet.
  +  - clears the screen.
* There’s a problem with this program. The sequence of statements in the box below appears on line 24 and again on line 33 of the program. This sequence draws one randomly colored dot at a random position on the screen. While the program works as planned, it is considered bad form to have duplicated blocks of code in a program. What if the block needs to change for some reason? Who will remember to change BOTH of them? What if you had this block of code scattered all throughout a software system and it needed to change? That would be bad. Keep reading for how to fix it.

// draw on the screen

int x = (int)(Math.random()\*getWidth())-10;

int y = (int)(Math.random()\*getHeight())-10;

Color c = new Color((int)(Math.random()\*256), (int)(Math.random()\*256), (int)(Math.random()\*256));

g.setColor(c);

g.fillOval(x, y, 20, 20);

* Make a java **method** out of this sequence of statements. That way the sequence won’t appear twice in the code anymore. We can call it up any time we want by calling the **method.** Follow these specifications:
  + Write a method block: give it a meaningful name – it should be a verb that describes what the block of code above does. The first letter should be a lower case letter.
  + Don’t use *static* on the method header like we did in the demo. JavaGUI is not running as a static class like the demo is (we’ll learn more about this later).
  + It doesn’t need to return a value, so use void.
  + It does not need to receive any parameters, so keep the parenthesis empty.
  + Remove the duplicated code from both places, and paste a single copy of it inside the method you just created.
* Now, *call* the new method from the places where the original code was. Then, make sure the program works just like it did before.
* Now, write the code for the 3rd button. It should draw 100,000 dots on the screen (by repeatedly calling your new method of course!)