Reading Questions

1. Class, Responsibilities, and Collaborators. Basically saying that one has made an outline to a class Superbot.

Classes

1. A piece of a program’s code that describes a specific group of objects.
2. Class is a group of objects and an object is one piece of a class. For example, UrRobot is a class and move is an object. Swimbot is a class and sMain is an object.
3. Class must be also called DinosaurBot and first letter must be capital.
4. Yes, so that one doesn’t have multiple classes of the same name that run in one file.
5. Instance variables consists of values stored in a program, a way the class can be created, and a function of the object. The difference between this and a field is that a field doesn’t have to be explicitly define.
6. State
7. public void run();
8. Yes, you can, but you will have to refer to color by their full file path name because you didn’t import, or make it available for local reference to your program.
9. No.
10. Comment at the top, import statements, class header, and class definition body.
11. No. They say this because it makes the code more readable.

Methods

1. It depends on what you seek to achieve, are you trying to change the color of all bugs to any color or all bugs to one color.
2. Temporary stored data from the program.
3. A Java interpreter that cleans out the temporary stored data that is not needed or readable anymore.
4. No, because null and false are built-in constructors.
5. Put a command on the main do function to tell an object to do something.
6. If we are running code, the object owns the method because the object is doing the method. If we are looking at the source code, the class owns the method because the method is part of the class.
7. Say void.
8. Use private.
9. It’s easier to maintain, document, and reuse an encapsulated class.

Inheritance

1. Reason 1: Clutters up the programs and makes it hard to read.

Reason 2: You can’t change the class, it’s private.

Reason 3: You can avoid more testing, if existing methods were already tested,

1. Methods and fields are inherited. Constructors are not.
2. No, because every class extends to the object class by default.