

# COMP 504: Graduate Object-Oriented Programming and Design

## Lecture 3: Infrastructure Setup

Mack Joyner ([mjoyner@rice.edu](mailto:mjoyner@rice.edu))

<https://www.clear.rice.edu/comp504>



# Worksheet #2: Composite Design Review

Assume: Shapes are opaque

Is there a problem with trying to use the composite design pattern in the model given the current definition of the abstract class AShape?

Explain why or why not?

```
9  public abstract class AShape {
10      protected Point loc;
11      protected String color;
12
13      /**
14       * Get the shape name
15       * @return shape name
16       */
17      public abstract String getName();
18
19      /**
20       * Get the shape color.
21       * @return shape color
22       */
23      public String getColor() {return this.color; }
24
25
26      /**
27       * Paint or repaint the shape at a location. The
28       * lefthand corner of the shape.
29       */
30      public abstract void paint(Point loc, String c);
31  }
```



# Worksheet #2: Composite Design

---

- Solution: Yes, there is a problem!
- The composite object's children know nothing about each other
  - There's no coordination between children
- The composite operation calls each child's operation method
  - The composite object should not influence the children's behavior
  - problem: all children are drawn at specific location
- The design of one leaf operation should not affect the design of any other leaf operation
  - problem: size now matters to view all shapes



# Design Issue

---

There's a design issue in the abstract *paint* method in AShape

- The location and color arguments impose behavior on composite children
- Location and color shouldn't be passed to paint method
- Use another name



# Announcements & Reminders

---

- HW #1 is available now, due **Fri, Sep 4th by 11:59pm**
- Use Piazza (public or private posts, as appropriate) for all communications re. COMP 504
  - Do not include code in a public post (**could be considered an honor code violation**)
- See course web site for syllabus, work assignments, due dates, office hours schedule.



# In-Class Exercises

---

- We'll have 1 for each class lecture
- Based on material presented in the lecture
- Grade rubric
  - 1 = submit before end of class
  - 1 = submit before beginning on next class
- Submit in-class exercises in git repo



# In-Class Exercise 1 GitHub Repo

---

GitHub ex1 repo: <https://classroom.github.com/a/mt44mSy6>



# In-Class Exercise 1: Infrastructure Setup

---

## Exercise 1 handout is in Piazza

- Piazza Setup
- Java 11 installation
- Maven installation
- IntelliJ installation
- GitHub classroom integration
- JUnit
- Introduction to Spark Java
- Optional: Print out JSON data of a circle that you create in hw #1 in hw1 repo

