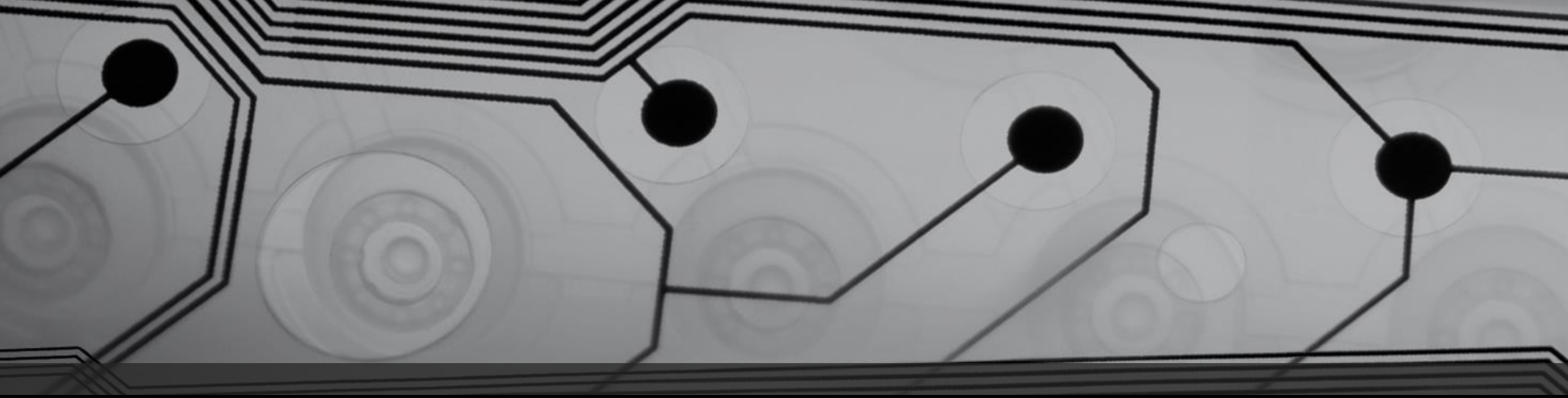

CMSC 131

Fall 2018

Announcements

- Project #3 has been posted
- Exam will be returned in discussion on Wed/Thurs
- Median score was XXXX

Recall Dog class. Review:

Suppose we put this instance method in the Dog class:

```
public void bark() {

    Dog d = new Dog();

    // which of the following are OK?
    …weight…
    …d.weight…
    …Dog.weight…
    …MIN_DOG_WEIGHT…
    …d.MIN_DOG_WEIGHT…
    …Dog.MIN_DOG_WEIGHT…
```

Review

Suppose we put this static method in the Dog class:

```
public static void foo() {

    Dog d = new Dog();

    // which of the following are OK?
    …weight…
    …d.weight…
    …Dog.weight…
    …MIN_DOG_WEIGHT…
    …d.MIN_DOG_WEIGHT…
    …Dog.MIN_DOG_WEIGHT…
```

Project #3

Let's look at the project together…

- Overview (Let's run it!)
- Soldier class (you will write this)
- BattleField API
- Contest
- This project is a bit hard to "test" (so submit early and often!)

- Example of implementing a method in soldier class:

```
public boolean isEnemyAboveMe() {

}
```

Is our Code “Correct”?

- Formal Verification
  - What’s good about it?
  - What’s bad about it?
  - Where might this be used?

- Testing
  - What’s good about it?
  - What’s bad about it?
  - Where might this be used?
  - What is “Unit Testing”?
  - What is “Integrated Testing?

JUnit

- What is JUnit?
  - Used on our Submission Server
- When are tests written?
- Where are they kept?
- How long are they kept?

Assertions

- assertTrue(<boolean expression>)
- assertFalse(<boolean expression>)
- assertEquals(x, y)
- Many others

Begin Demonstration: FunnyIntegerSet

This class has some bugs. Let's test it with JUnit!

```
public FunnyIntegerSet()
```

Instantiates an "empty" set

```
public add(int x)
```

Adds a value to the set

```
public findClosest(int x)
```

Returns the value in the set that is "closest" to x