Questions about Lab 1

[ ] [O] [ ] [O] [ ] [ ] [O] [ ] [ ]
0 1 2 3 4 5 6 7 8

Which coin do you want to move? 3
How far?

---

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
</table>
Review

- **Invariants**
  - Enforced by `protected` state, `accessors`, and `final`
- **Interfaces** are stateless abstractions of properties
- **static** members/methods on the class instead of an instance
- `java.util.Random`
- Strategies for storing state in arrays
public class NumberBag {
    protected int[] data;

    public NumberBag(int n) {
        data = new int[n];
        data[3] = 62;
        data[data.length - 1] = 4;
        System.out.println(data[0]);
    }
}
Today

1. Histogram
2. String
3. Assertions
HISTOGRAM LIVE CODING DEMO
Essential String Methods

https://docs.oracle.com/javase/8/docs/api/java/lang/String.html

int length()
char charAt(int)
String substring(int, int)
int indexOf(String, int)
Assertions

assert $Expression_1 : Expression_2$;

assert data != null : “No array provided”;

assert x > data.length :
    “Out of bounds x = “ + x +
    “ data.length = “ + data.length;
Design Summary

- **Design** your program on paper, in English
  - Nouns = state => **members**
  - Verbs = computation => **methods**
  - Group into **classes**
- A class should be a **reusable** collection of state + computation
- **main()** is your particular program (like top level in Python)
- Look at sample programs
- Read documentation
Monday

- **Contracts**: invariants, pre/post conditions, and interfaces revisited
- **Associations**
- **Vectors**: arrays that can change length

*Lab #1 due Monday night!*