Pick up midterm exam at front of classroom

Reading for next Tuesday
- PLPJ Chapter 6: Runtime organization for Object-oriented languages
Today’s topics

• Midterm exam
  – mean 53, median 55, stdDev 11
  – go over solutions

• PA4 assignment

• OO runtime organization for miniJava
  – representation
  – mJAM support

• Finish lecture 14
Midterm score distribution
Simple classes, no inheritance

- Classes
  ```java
  class A {int x; void p() {x = 3;}}
  ```
  - runtime entity descriptions in AST
    - class A: \( S_A = \text{size of class A (# fields)} \)
    - field x: \( d_x = \text{displacement of field x} \)
    - method p: \( d_p = \text{displacement of code for p} \)

- Objects
  - instances are created on the heap
    ```java
    A a = new A();
    ```

```
<table>
<thead>
<tr>
<th>Type</th>
<th>Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classes</td>
<td>LB ( d_a ) \rightarrow\rightarrow mJAM runtime layout ( S_A ) \rightarrow \rightarrow object instance in heap</td>
</tr>
</tbody>
</table>
Basic miniJava operations

- mJAM code sequences

```java
A a = new A();  // (object creation)
a. x;           // (qualified reference)
a. p();         // (method invocation)
x = x + 3;      // (field upd within p() )
```

```
LOADL -1
LOADL S_A
CALL newobj
STORE d_a[LB]
```

```
LOAD d_a[LB]
LOADL d_x
CALL fieldref
```

```
LOAD d_a[LB]
CALLI d_p[CB]
```

```
LOAD d_x[OB]
LOADL 3
CALL ADD
STORE d_x[OB]
```

- Activation record on stack
- Object instance in heap
- miniJava runtime organization