

CSSE 220 Day 10

Two-dimensional arrays, Array Copying,
Software Engineering Techniques

Check out *TwoDArrays* from SVN

Questions?

Exam Coming!

- ▶ Test next Wednesday
- ▶ **Can start at 7:30 if you want extra time**
- ▶ Topics from Ch. 1–7
- ▶ Will include:
 - A paper part—logic, short answer, fill-in-the-blank
 - A programming part—a few small programs, unit tests provided
- ▶ Review in class Monday
 - Bring questions
 - I won't anything prepared but am happy to cover whatever you want, including working examples

Two Dimensional Arrays

- ▶ Consider:

- `final int ROWS = 3;`
`final int COLUMNS = 3;`
`String[][] board = new String[ROWS][COLUMNS];`

- ▶ What's the value of `board[1][2]` now?

- ▶ Need to fill the 2-d array:

- ```
for (int r=0; r < ROWS; r++) {
 for (int c=0; c < COLUMNS; c++) {
 board[r][c] = " ";
 }
}
```

# Exercise

- »» Complete the TODO items in TicTacToe and TicTacToeTest They're numbered; do 'em in order.

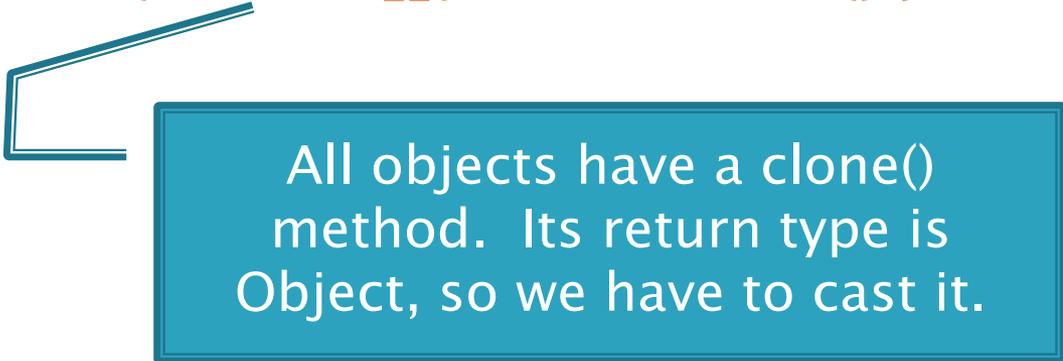
# Copying Arrays

- ▶ Assignment uses reference values:

- ```
double[] data = new double[4];  
for (int i=0; i < data.length; i++) {  
    data[i] = i * i;  
}  
double[] pieces = data;
```

- ▶ Can copy whole arrays:

- ```
double[] pizzas = (double []) data.clone();
```



All objects have a clone() method. Its return type is Object, so we have to cast it.

# Copying Part of an Array

- ▶ Use built-in function:
  - `System.arraycopy(fromArray, fromStart, toArray, toStart, count);`
- ▶ Copies
  - **count** values from **fromArray**,
  - beginning at index **fromStart**,
  - copying into array **toArray**,
  - beginning at index **toStart**

# Quality Tip

- ▶ “Avoid parallel arrays”
- ▶ We did this in ElectionSimulator
- ▶ Instead of storing:
  - `ArrayList<String> stateNames;`
  - `ArrayList<Integer> electoralVotes;`
  - `ArrayList<Double> candidateAOdds;`
  - `ArrayList<Double> candidateBOdds;`
- ▶ We used:
  - `ArrayList<State> states;`  
and put the 4 pieces of data inside a State object
- ▶ Why bother?

# Pick the Right Data Structure

- ▶ Array or ArrayList, that is the question
- ▶ General rule: use ArrayList
- ▶ Exceptions:
  - Lots of primitive data in time critical code
  - Two (or more) dimensional arrays

# Software Engineering Techniques

- ▶ Regression testing
  - ▶ Pair programming
  - ▶ Team version control
- 

# Regression Testing

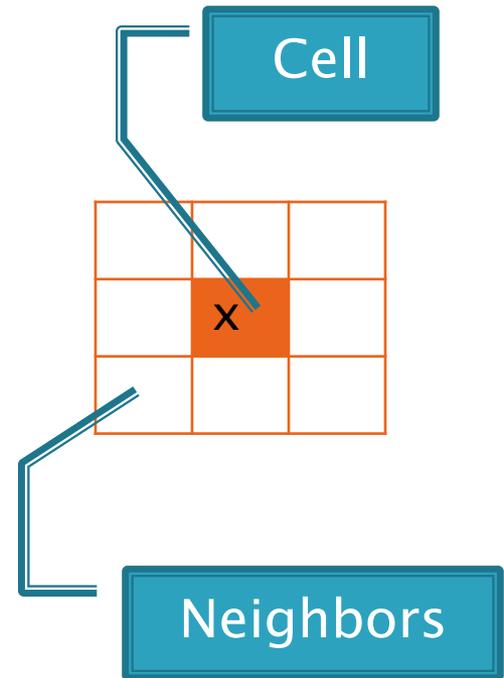
- ▶ Keep and run old test cases
- ▶ Create test cases for new bugs
  - Like antibodies, they keep a bug from coming back
- ▶ Remember:
  - You can right-click the project in Eclipse to run all the unit tests

# Pair Programming

»» Video

# Game of Life

1. A new cell is born on an empty square if it has exactly 3 neighbor cells
2. A cell dies of overcrowding if it is surrounded by 4 or more neighbor cells
3. A cell dies of loneliness if it has just 0 or 1 neighbor cells



# Team Version Control

- ▶ **Always:**
  - Update before working
  - Update again before committing
  - Commit often and with good messages
- ▶ **Communicate** with team mates so you don't edit the same code simultaneously
  - Pair programming eliminates this issue

# Game of Life Teams

| <i>n</i> |                   |
|----------|-------------------|
| 1        | Gardner and Alice |
| 2        | Cory and Joe      |
| 3        | Sam and Steve     |



Team number used in  
homework description