Lecture topic: arrays. They’re objects; declaring them; 0-based indexing system; the length constant; it’s NOT length(); Off-by-one errors.

Some problems:
Arrays:
Declare an array of 12 booleans called boo
Declare an array of 19 doubles called nums
Declare an array of 5 Infants called littleOnes

```java
public class CoffeeMug{
    String who; // whose mug is it?
    int capacity; // oz held in mug
    boolean full; // is cup full or not – only full or empty

    public CoffeeMug(String w, int cap, boolean f){
        who = w;
        capacity = cap;
        full = f;
    }
    public String getWho(){ return who;}
    public int getCapacity(){ return capacity;}
    public boolean getFull(){ return full;}
    public void setWho(String other){who = other;}
    public void setFullStatus(boolean isFilled)
        {full=isFilled;}
}
```

create an array of 5 mugs called myMugs.

Now: fill myMugs with cups that are all jill’s, first 3: 10 oz, last 2: 12 oz. – all full
Next: empty them all
Next: fill them all
Next: print a report about each mug- owner/capacity/full?
Next – write a method that’s passed an array of mugs, and returns the number that are full.
Opoly works this way: The board is a straight linear track of variable length. There is only one playing piece, which begins the game just off the board, at position 0. Thus, if the board length is 20, then the board positions start at position 1 and end at position 20. To finish the game, the piece must land exactly on the last cell of the board.

The object of the game is to acquire reward. The reward amount is initialized to 10. If you board piece lands on a board cell that is divisible by 4, your reward doubles. However, if your piece lands one cell shy of the final board cell, your reward is reduced to 1/3 of its current value, and your piece must go back to the start position - position 0.

In Opoly the game piece advances via a spinner - a device that takes on the values 1-2-3-4-5 at random, with each advance value equally likely.

Two additional rules:
1) if a spin would move the piece beyond the end of the board, the piece should not advance at all.
2) if the next to last board location is divisible by 4, and if the piece lands on this location, the reward is reduced to 1/3 of its current value only - the reward is NOT also doubled.

> java OpolyDriver

Enter an int - the size of the board

Board Size: 20

*O**************************** 10
***O************************** 20
********O******************** 40
**********O******************* 40
****************O************* 80
******************O********** 80
*******************O********* 80
*********************O***** 160

game over

rounds of play 7

final reward 160

A requirement: your Opoly class must include the following methods,

spin - generates a integer value from 1 to 5 at random

move - advances the piece

spinAndMove - spins the spinner and then advances the piece according to the rules of the game (uses spin, move methods)

isGameOver - checks if game termination condition has been met

drawBoard - draws the board using *'s and an O to mark the current board position

displayReport - reports the end of the game, and gives the number of rounds of play, and the final reward

One more wrinkle. You should use the Rows class from chapter 5 of the textbook. You can download it from the tab at the top of the book labeled "Source Files". If you compile this class, you can use its methods directly for drawing the board.