Midterm 2 Review

CS 302 Spring 2013

Exam Tips

- Use the Reference Sheet
- Prioritize: Some problems take a lot longer than others to do!
- Plan Ahead: Leave space at the top of problems.
- Extra Sheets of Paper at the Back

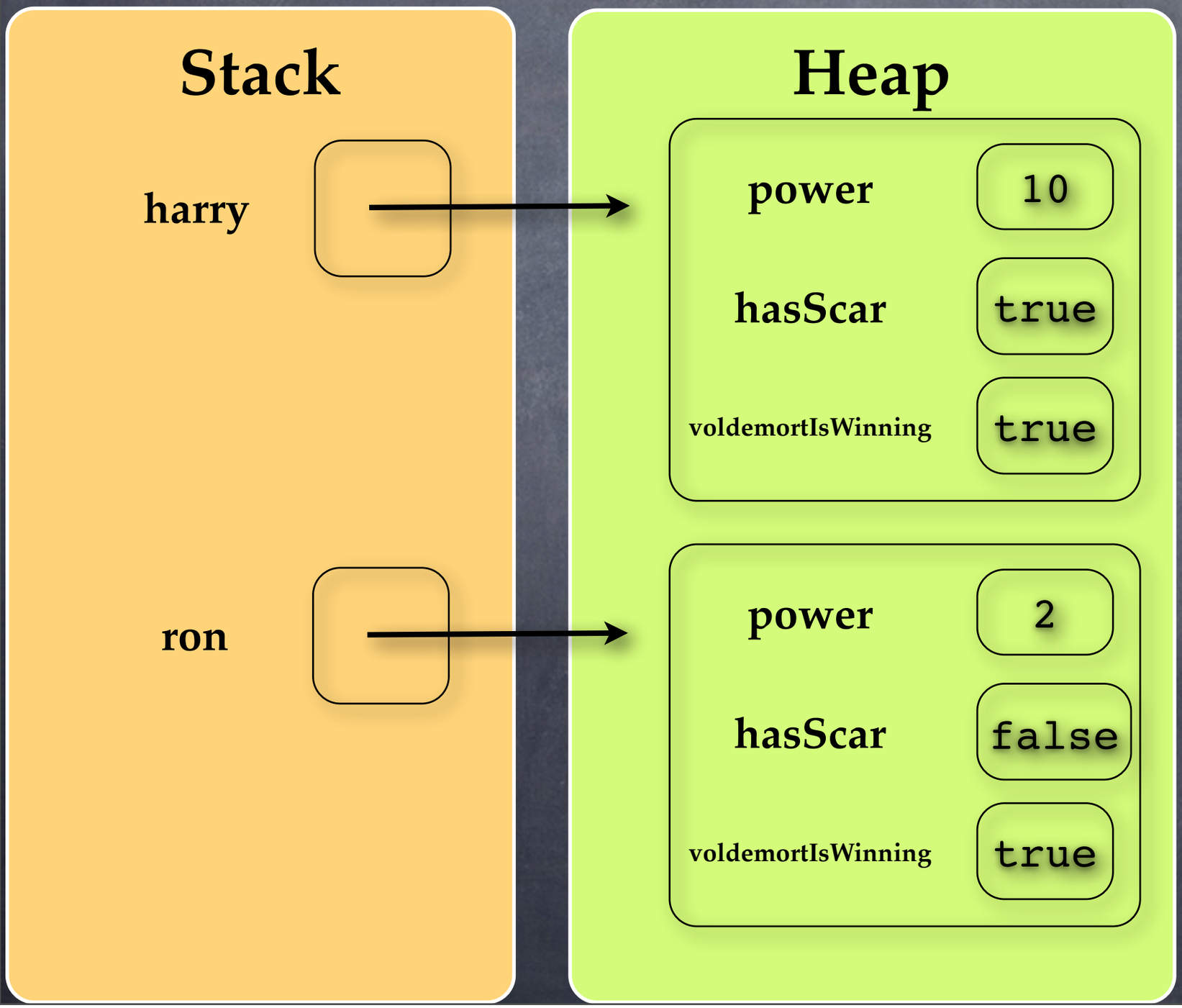
More Tips

- Write Legibly!
- Every character matters! We will take off for syntax mistakes.
- Do not use symbols: π, ≠, ÷, ≥, ÷, ●, etc.
- Do not write in all caps - uppercase/ lowercase matters!

Static Fields

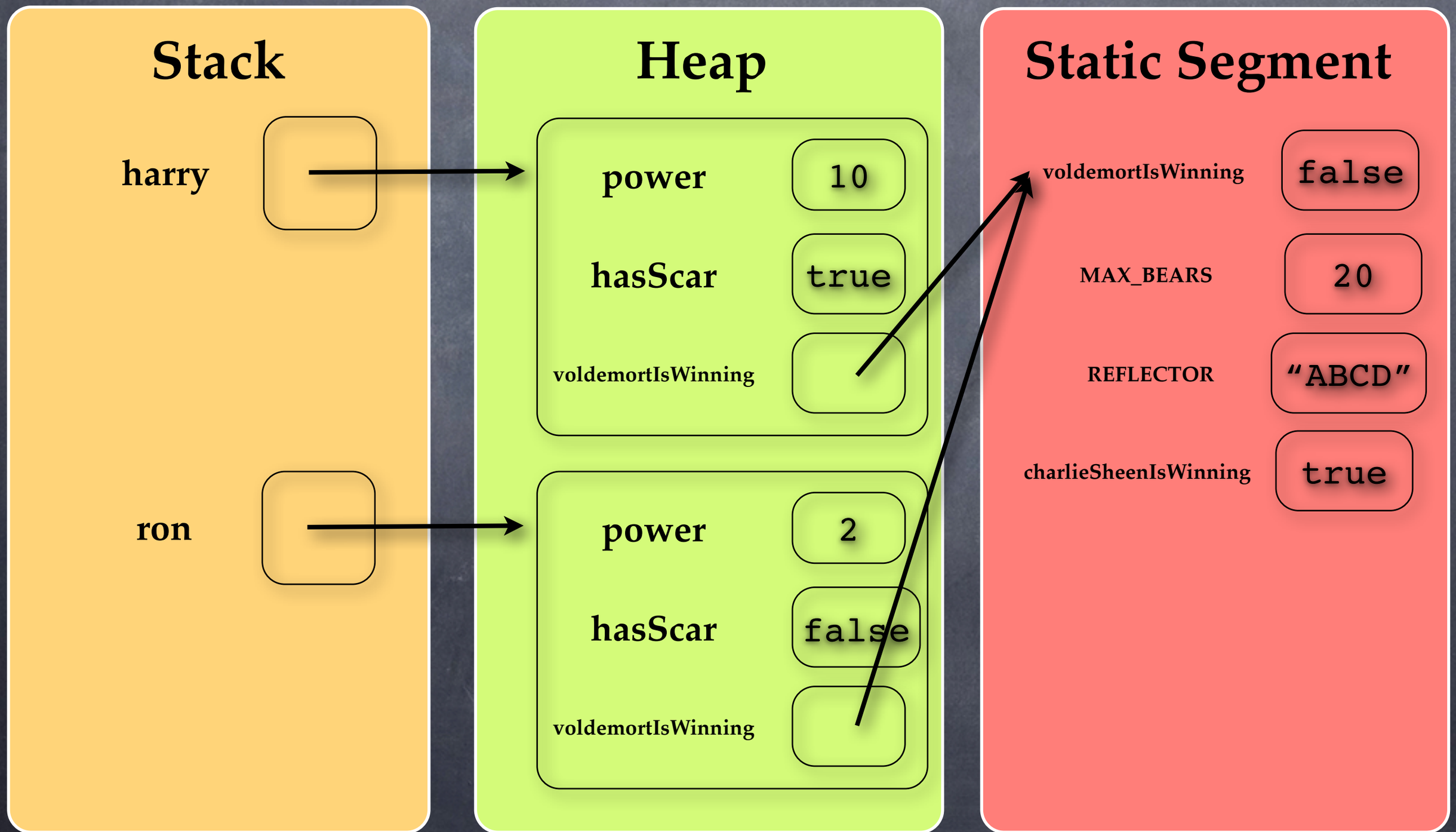
```
public class Wizard{
      private int power;
      private boolean hasScar;
      private boolean voldemortIsWinning;
}
```

```
Wizard harry = new Wizard();
Wizard ron = new Wizard();
```



```
public class Wizard{
      private int power;
      private boolean hasScar;
      private static boolean voldemortIsWinning;
}
```

```
Wizard harry = new Wizard();
Wizard ron = new Wizard();
```



Static Members Must be Accessed via the Class Name

```
public class Wizard{
      public int power;
      public static boolean voldemortIsWinning;
}
```

```
public static void main(String[] args){
   Wizard harry = new Wizard();
   Wizard ron = new Wizard();
   harry.power = 4;
   ron.power = 6;
   Wizard.voldemortIsWinning = true;
}
```

Encapsulation

- The process of separating public interface from private implementation.
- Enables the idea of controlled access to managed state.

Encapsulation Example

Public Interface

Private Implementation

```
public class Barn{
    private ArrayList<Chicken> chickenCoop;

    public int getNumChickens(){
        return chickenCoop.size();
    }

    public void addChicken(Chicken chicken){
        if(chicken.isAGoodChicken()){
            chickenCoop.add(chicken);
        }
    }
}
```

Think: Cheshire Cat

Public Interface

Benefits:

- Hiding of Complexity
- Guarding of State

```
public class Barn{

    public int getNumChickens(){

    public void addChicken(Chicken chicken){

}
```

Why is main() public static void?

Why is main() public static void?

- public so that other classes may call it
- static because it does not operate on any instance
- void because it does not return anything

Syntax Gotchas

```
int[] array = new int[20];
ArrayList<Integer> arrayList = new ArrayList<Integer>();
String str = "Hello World!";
```

```
int a = array.length
int b = arrayList.size();
int c = str.length();
```

Declaration & Assignment

```
// Declares a new ArrayList variable and initializes it to null
ArrayList<String> arrList1 = null;

// Declares a new ArrayList and initializes it to refer
// to a newly-constructed ArrayList
ArrayList<String> arrList2 = new ArrayList<String>();

// Assigns arrList2 to refer to a different ArrayList.
arrList2 = new ArrayList<String>();
```

What's the difference between public and static?

Public vs. Static ?

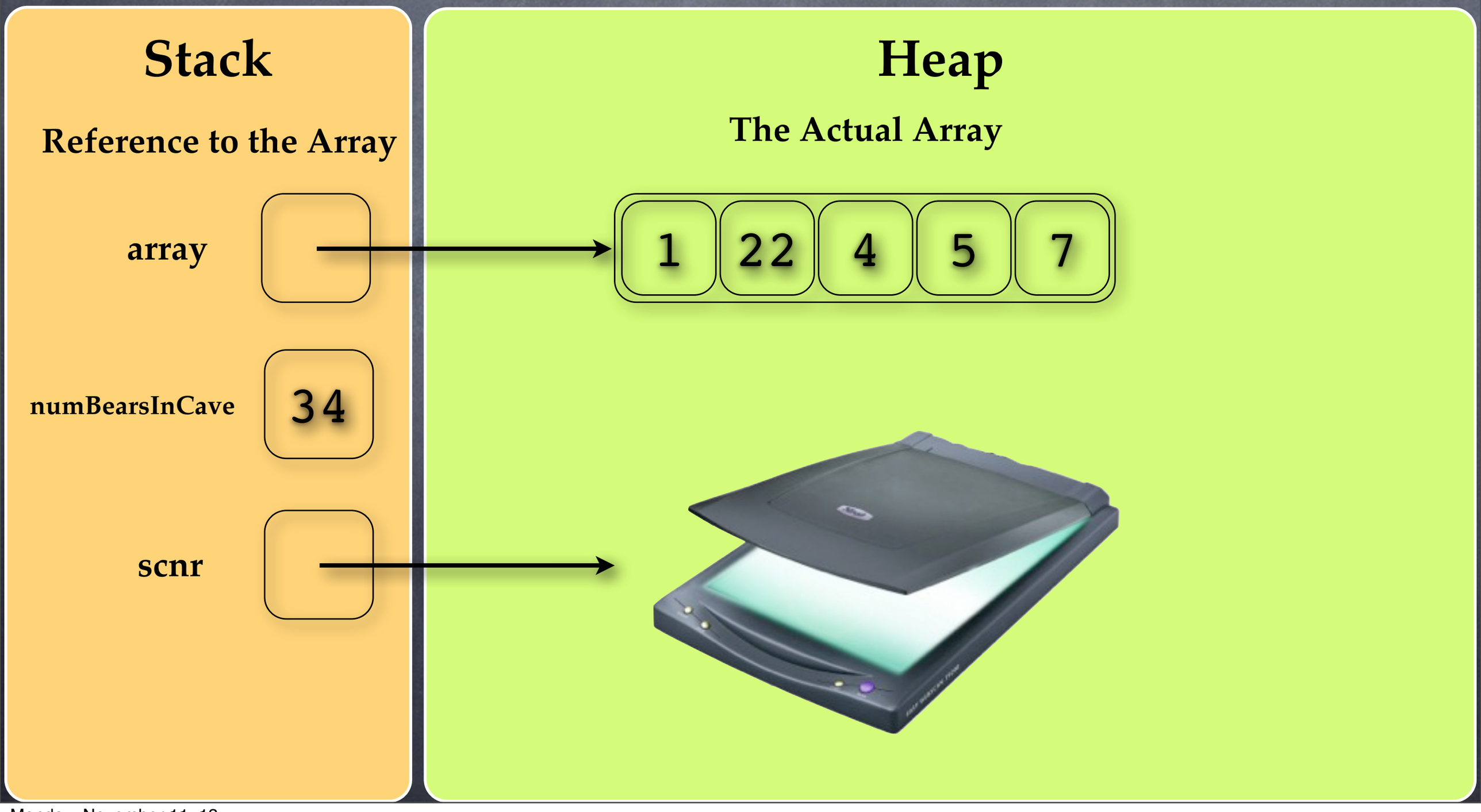
- public is an access specifier. Access specifiers such as public and private specify who can access a class.
- static forces a method to become a class method, which disassociates it from any given instance.

The Special Methods

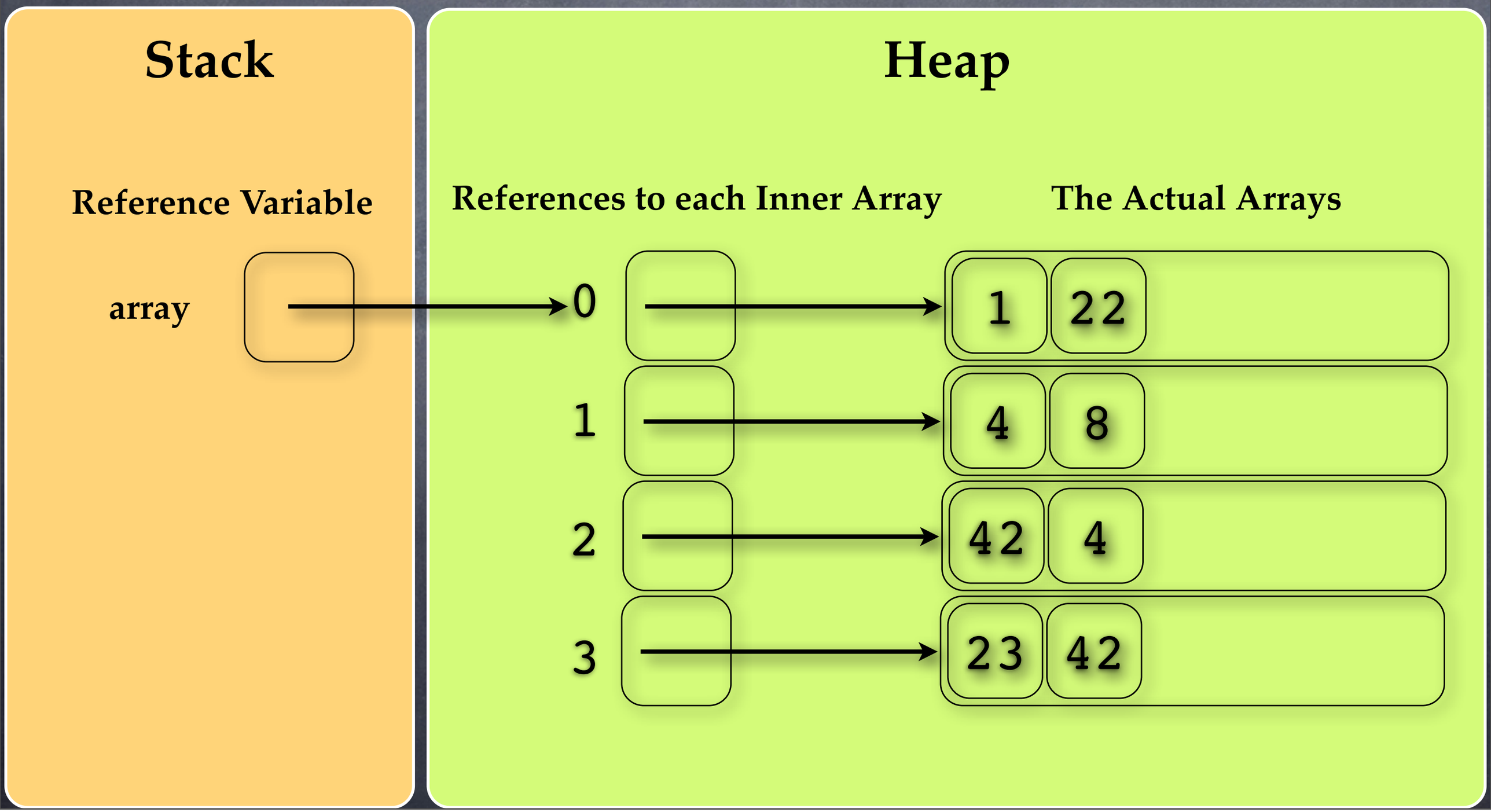
- Two methods which every class automatically supports:
- boolean equals(Object other)
  - Used for comparing instances with each other
- String toString()
  - Used commonly for debugging
  - Automatically called when an object is passed in where a String is expected

Memory Diagrams

```
int[] array = {1, 22, 4, 5, 7};
int numBearsInCave = 34;
Scanner scnr = new Scanner(System.in);
```

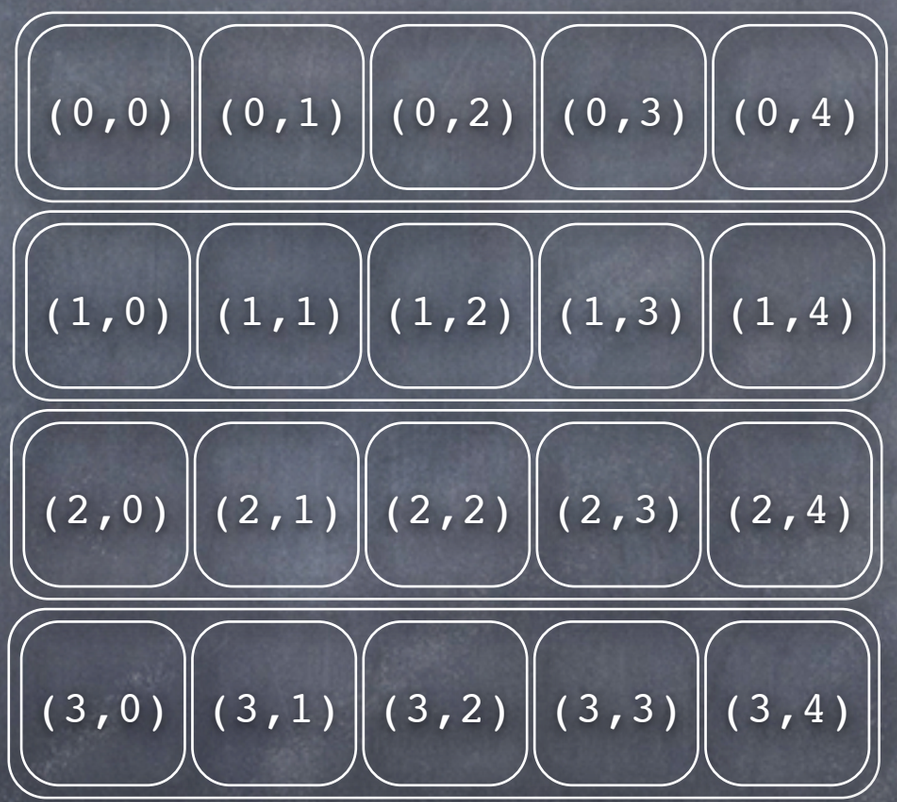


```
int[][] array = new int[4][2];
array[0][0] = 1;
array[0][1] = 22;
etc.
```



2D Array Indices

CS-Style Array Indices



Remember Input Validation?

```
int userInput = 0;
System.out.println("Enter an integer between 7 and 11");
do{
   while( !scnr.hasNextInt()) {
      System.out.println("Error: not an integer!");
      scnr.next();
   }

   userInput = scnr.nextInt();

   if (userInput < 7 || userInput > 11){
      System.out.println("Error: not between 7 and 11");
   }
}while(userInput < 7 || userInput > 11);
```

Reverse an ArrayList

```
public static ArrayList<String> reverse(ArrayList<String> arr){

}
```

Study Tips

- Try the Sample Exam (And Answers)
- Look at Lecture Notes (on website)
- Look through examples
- Practice with writing P3