CIS 110: Intro to Computer Programming

Lecture 10

Interaction and Conditionals

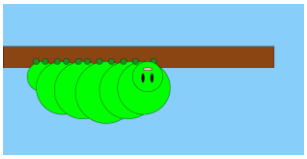
(§ 3.3, 4.1-4.2)

Outline

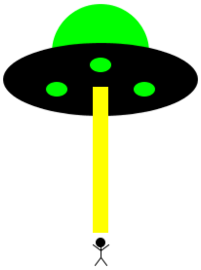
- The Scanner Object
- Introducing Conditional Statements
- Cumulative Algorithms

The Scanner Object

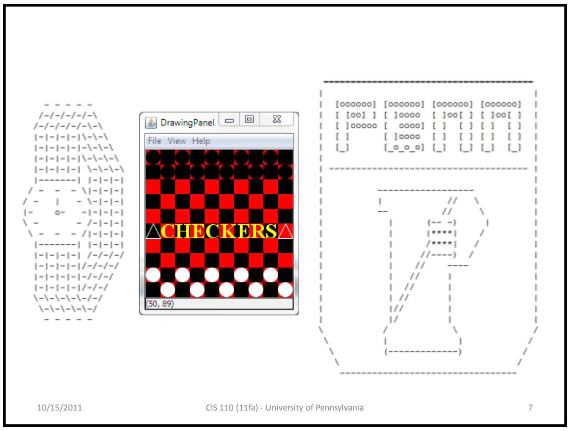
What Do Our Programs Look Like?



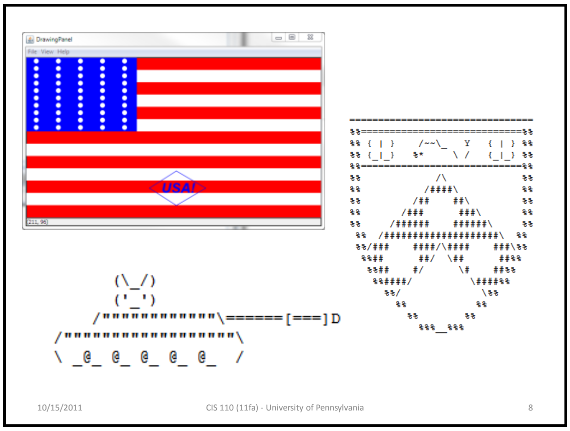
```
            ()
            ||##########
            ||~~~~~~#
            ||~~~~#
            ||~~#
            ||#
            ||
          |-----|
        |---------|
      |-------------|
    |-----------------|
    | [] [] [] [] []  |
    | [] [] [] [] []  |
    | [] [] [] [] []  |
    | [] [] [] [] []  |
    | [] [] [] [] []  |
    | [] [] [] [] []  |
    | [] [] [] [] []  |
    | [] [] [] [] []  |
    | [] [] [] [] []  |
    | [] [] [] [] []  |
    | [] [] [] [] []  |
    | [] [] [] [] []  |
    | [] [] [] [] []  |
    | [] [] [] [] []  |
    | [] [] [] [] []  |
  /=====================\
/=========================\
```



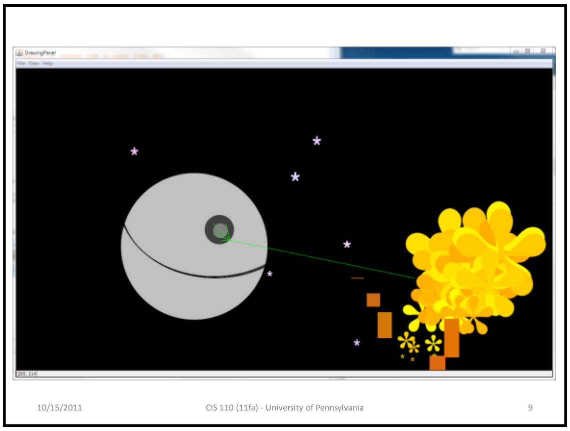
```
This program replicates a drawing
of the sinking Titanic and two
survivors in the water.
****
*****
******
*******
********
*********
**********
***********
***********    \o_   \o/
~^~^~^~^~^~^~^~^~^~^~^~^~^~^
~^~^~^~^~^~^~^~^~^~^~^~^~^~^
~^~^~^~^~^~^~^~^~^~^~^~^~^~^
~^~^~^~^~^~^~^~^~^~^~^~^~^~^
```



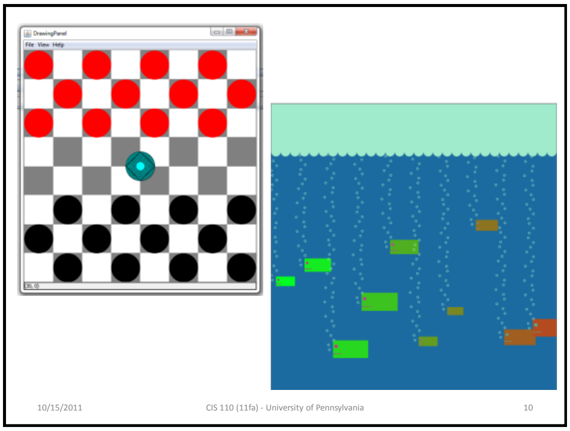
10/15/2011 CIS 110 (11fa) - University of Pennsylvania 7



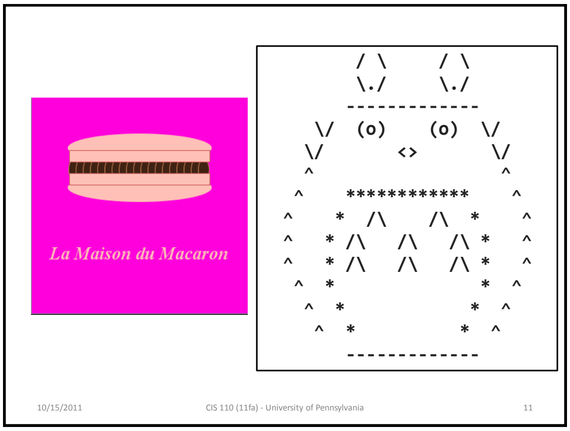
10/15/2011 CIS 110 (11fa) - University of Pennsylvania 8



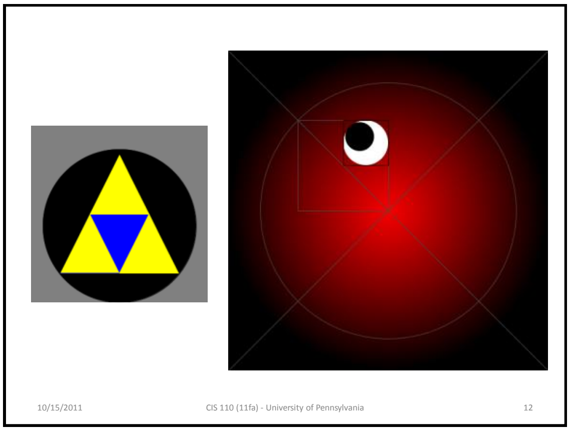
10/15/2011 CIS 110 (11fa) - University of Pennsylvania 9



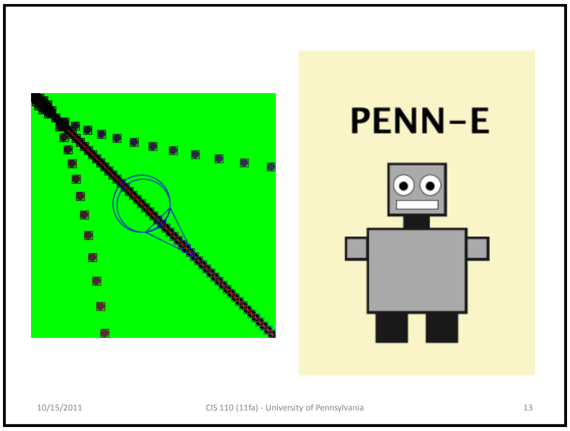
10/15/2011 CIS 110 (11fa) - University of Pennsylvania 10



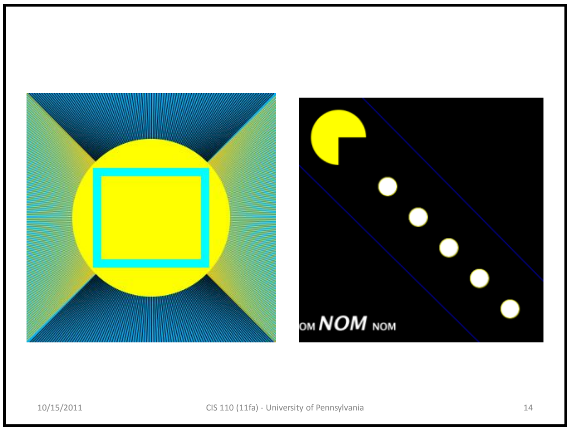
10/15/2011 CIS 110 (11fa) - University of Pennsylvania 11



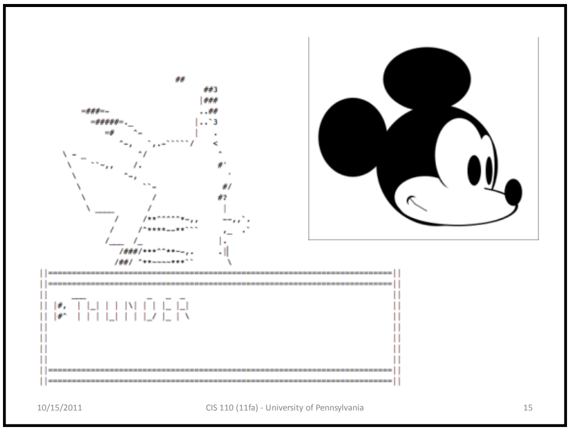
10/15/2011 CIS 110 (11fa) - University of Pennsylvania 12



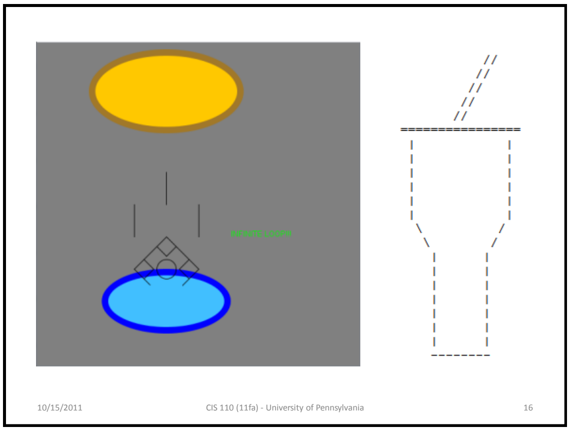
10/15/2011 CIS 110 (11fa) - University of Pennsylvania 13



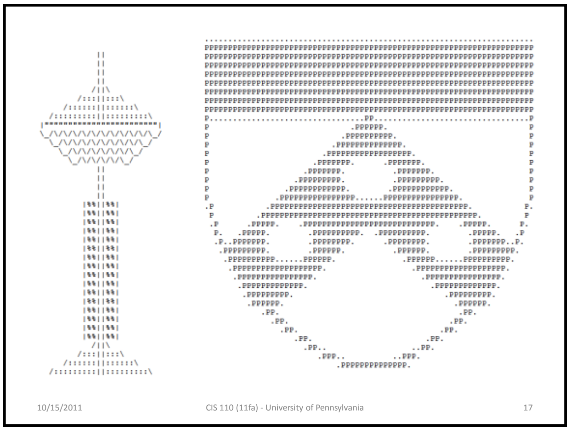
10/15/2011 CIS 110 (11fa) - University of Pennsylvania 14



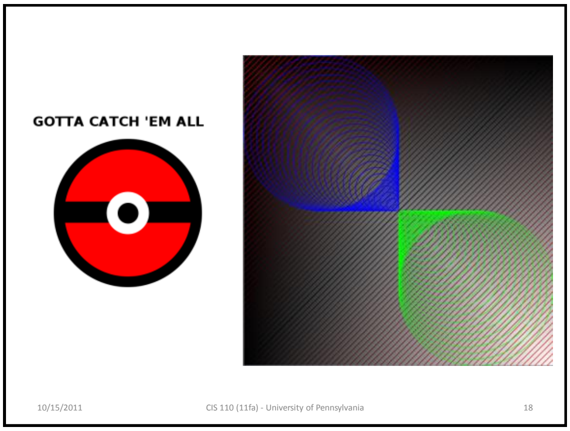
10/15/2011 CIS 110 (11fa) - University of Pennsylvania 15



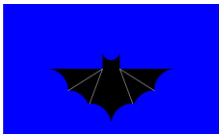
10/15/2011 CIS 110 (11fa) - University of Pennsylvania 16



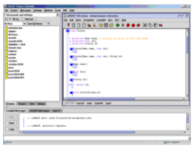
10/15/2011 CIS 110 (11fa) - University of Pennsylvania 17



10/15/2011 CIS 110 (11fa) - University of Pennsylvania 18



Real World Programs Out There



Real programs are **INTERACTIVE!**

Introducing the Scanner Class

- Object that lets you read input from the user.

Scanner is in the "java.util" package.

Creates a new Scanner that reads from some source.

```
import java.util.*;

Scanner in = new Scanner(System.in);
System.out.println(
   "Echo: " + in.nextLine());
```

"The keyboard"

Grabs the next line of input from the Scanner.

Reading Input From The User

```
// Reads in a double
double d = in.nextDouble();
// Reads in an integer
int n = in.nextInt();
// Reads in an entire line
String line = in.nextLine();
// Reads a token
String token = in.next();
```

Tokens

- *Tokens* are "chunks" of an input separated by a *delimiter* (here, whitespace).

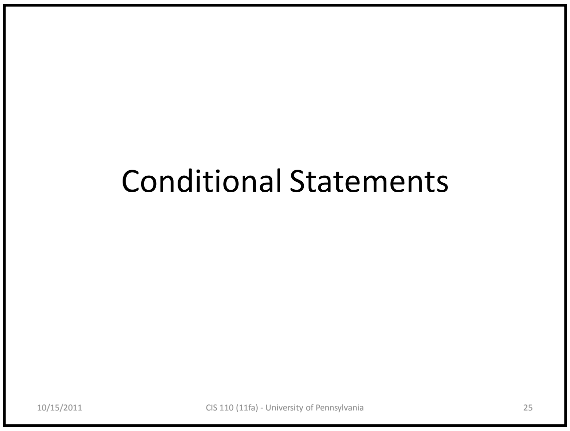
```
This is a string     with
  "some tokens" in           it!
```

- Tokens: this, is, a, string, with, "some, tokens", in, it!
  - Includes punctuation (e.g., quotes and bangs).

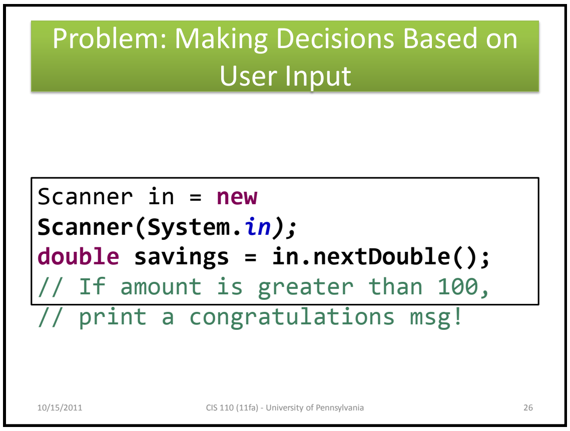
Aside: Packages and import

- Classes are bundled into sets called *packages*.
- The *import declaration* says that you wish to use classes found in a particular package.

```
// Make available all classes in java.util
import java.util.*;
// Make available just the Scanner class
import java.util.Scanner;
```

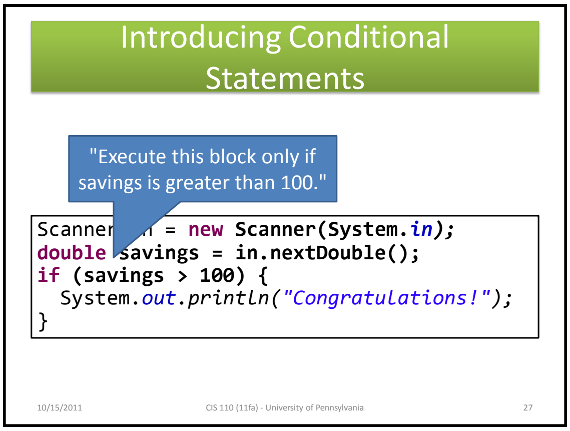


Conditional Statements



Problem: Making Decisions Based on User Input

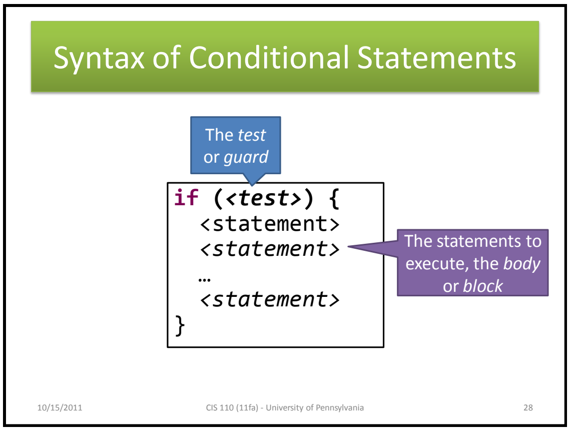
```
Scanner in = new
Scanner(System.in);
double savings = in.nextDouble();
// If amount is greater than 100,
// print a congratulations msg!
```



Introducing Conditional Statements

"Execute this block only if savings is greater than 100."

```
Scanner in = new Scanner(System.in);
double savings = in.nextDouble();
if (savings > 100) {
  System.out.println("Congratulations!");
}
```

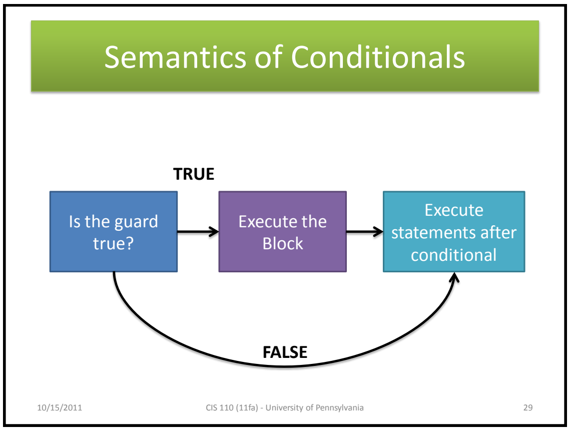


Syntax of Conditional Statements

The *test* or *guard*

```
if (<test>) {
  <statement>
  <statement>
  …
  <statement>
}
```

The statements to execute, the *body* or *block*



Semantics of Conditionals

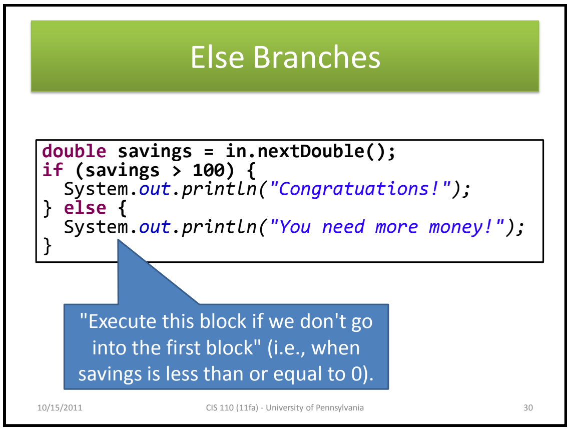
TRUE

Is the guard true?

Execute the Block

Execute statements after conditional

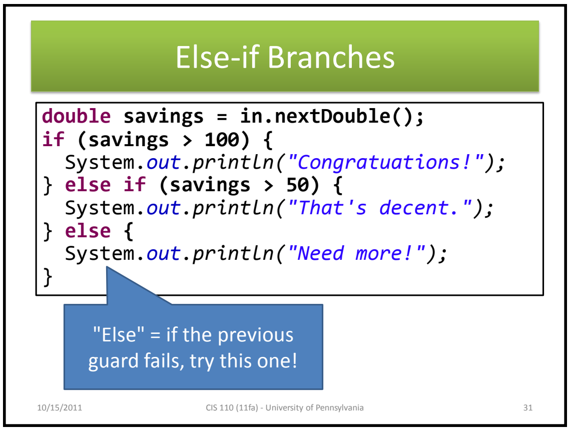
FALSE



Else Branches

```
double savings = in.nextDouble();
if (savings > 100) {
  System.out.println("Congratuations!");
} else {
  System.out.println("You need more money!");
}
```

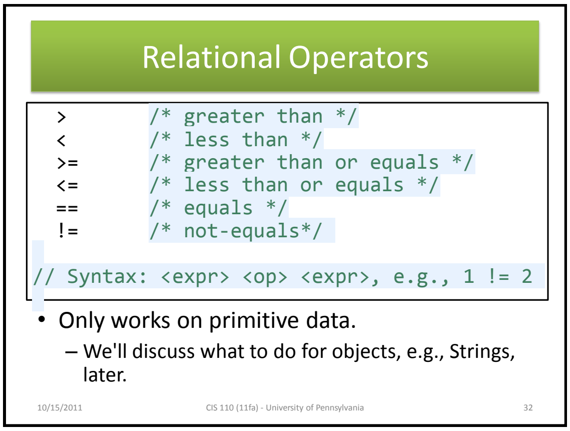
"Execute this block if we don't go into the first block" (i.e., when savings is less than or equal to 0).



Else-if Branches

```
double savings = in.nextDouble();
if (savings > 100) {
  System.out.println("Congratuations!");
} else if (savings > 50) {
  System.out.println("That's decent.");
} else {
  System.out.println("Need more!");
}
```

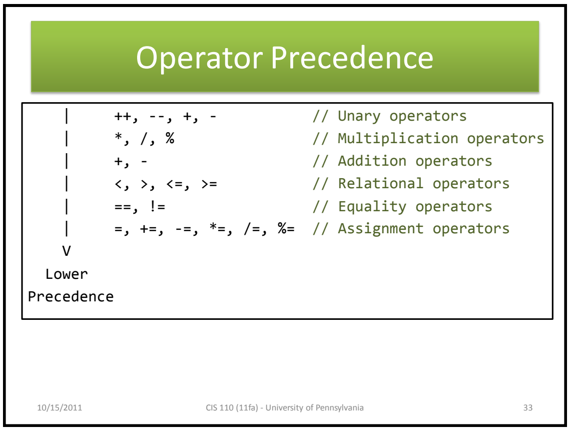
"Else" = if the previous guard fails, try this one!



Relational Operators

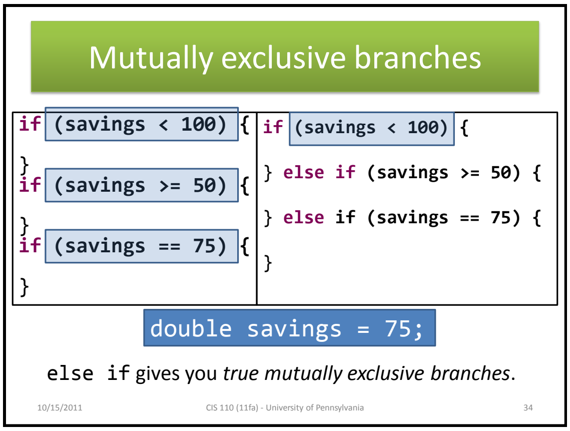
```
  >         /* greater than */
  <         /* less than */
  >=        /* greater than or equals */
  <=        /* less than or equals */
  ==        /* equals */
  !=        /* not-equals*/

// Syntax: <expr> <op> <expr>, e.g., 1 != 2
```

- Only works on primitive data.
  - We'll discuss what to do for objects, e.g., Strings, later.



Operator Precedence

```
    |     ++, --, +, -             // Unary operators
    |     *, /, %                  // Multiplication operators
    |     +, -                     // Addition operators
    |     <, >, <=, >=             // Relational operators
    |     ==, !=                   // Equality operators
    |     =, +=, -=, *=, /=, %=    // Assignment operators
    V
  Lower
Precedence
```



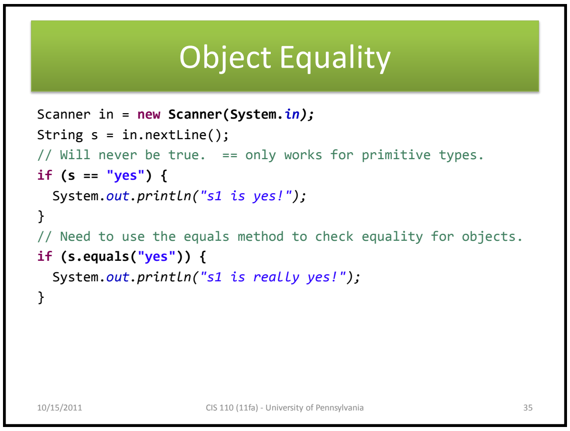
Mutually exclusive branches

```
if (savings < 100) {
}
if (savings >= 50) {
}
if (savings == 75) {
}
```

```
if (savings < 100) {
} else if (savings >= 50) {
} else if (savings == 75) {
}
```

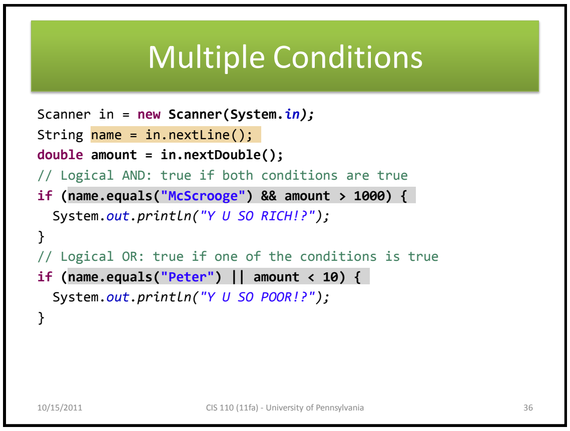
```
double savings = 75;
```

`else if` gives you *true mutually exclusive branches*.



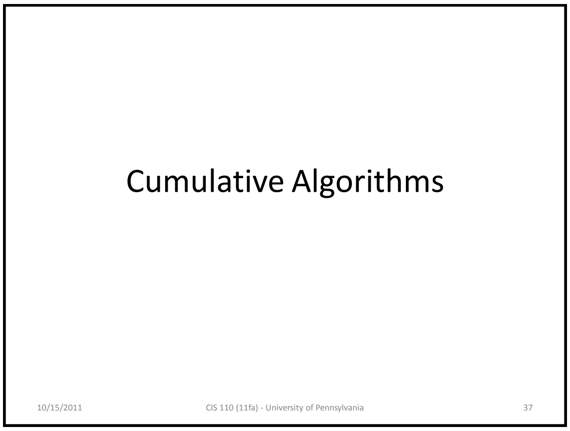
Object Equality

```
Scanner in = new Scanner(System.in);
String s = in.nextLine();
// Will never be true.  == only works for primitive types.
if (s == "yes") {
  System.out.println("s1 is yes!");
}
// Need to use the equals method to check equality for objects.
if (s.equals("yes")) {
  System.out.println("s1 is really yes!");
}
```

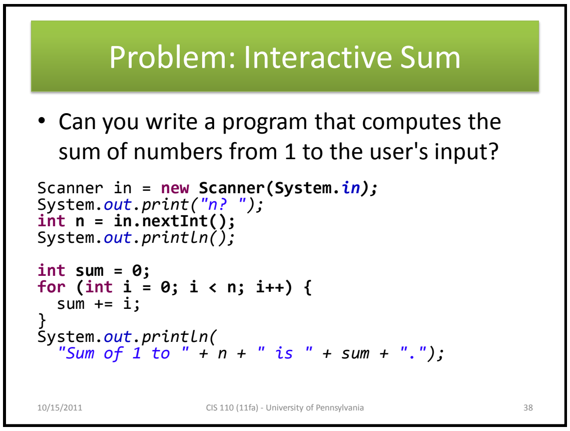


Multiple Conditions

```
Scanner in = new Scanner(System.in);
String name = in.nextLine();
double amount = in.nextDouble();
// Logical AND: true if both conditions are true
if (name.equals("McScrooge") && amount > 1000) {
  System.out.println("Y U SO RICH!?");
}
// Logical OR: true if one of the conditions is true
if (name.equals("Peter") || amount < 10) {
  System.out.println("Y U SO POOR!?");
}
```



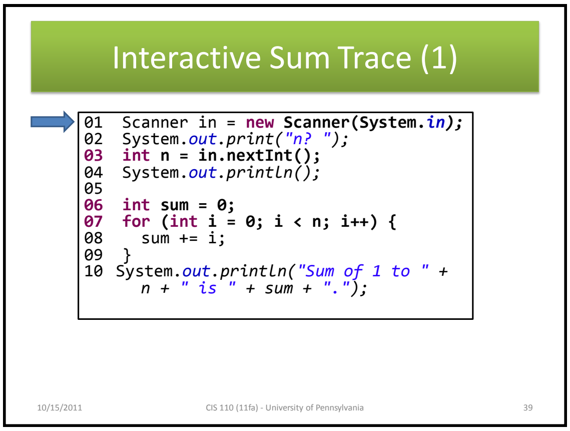
Cumulative Algorithms



Problem: Interactive Sum

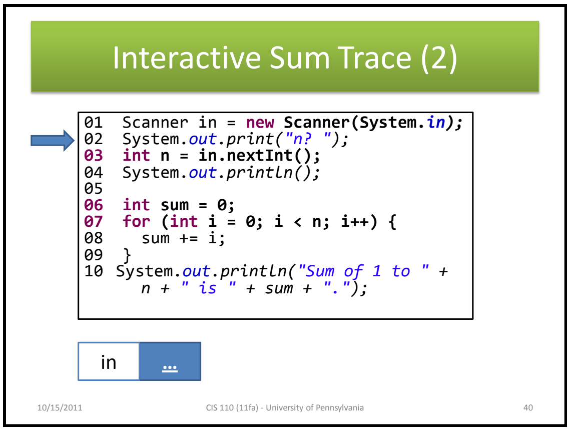
- Can you write a program that computes the sum of numbers from 1 to the user's input?

```java
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println(
  "Sum of 1 to " + n + " is " + sum + ".");
```



Interactive Sum Trace (1)

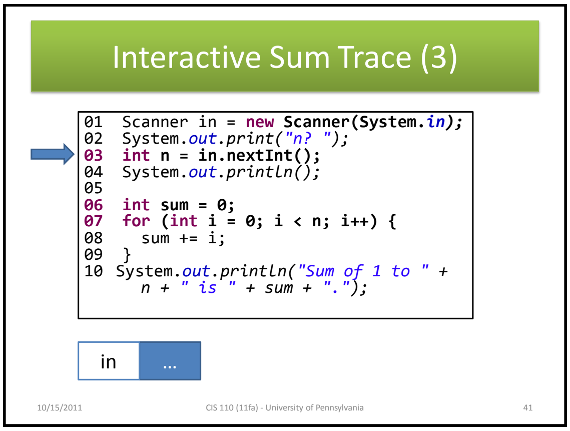
```java
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



Interactive Sum Trace (2)

```java
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

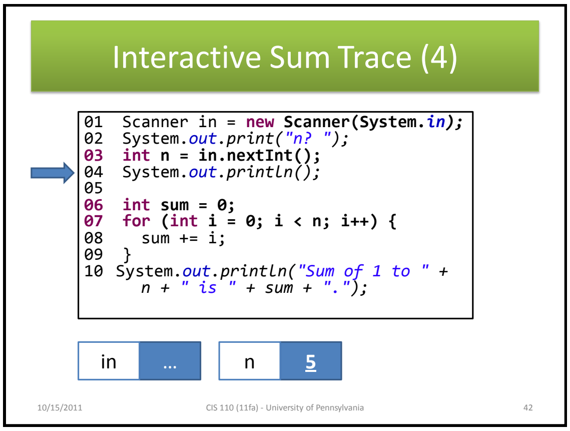
in ...



Interactive Sum Trace (3)

```java
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

in ...

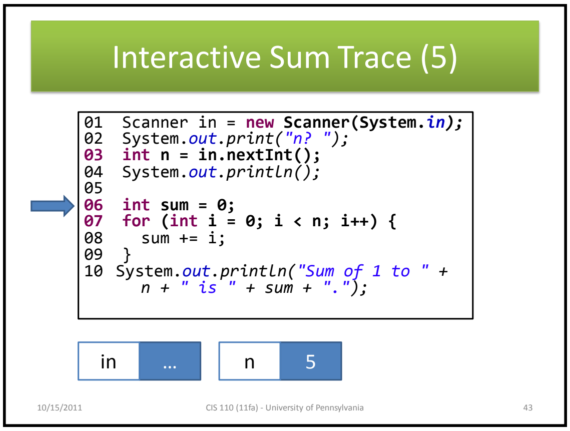


Interactive Sum Trace (4)

```java
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

in ...

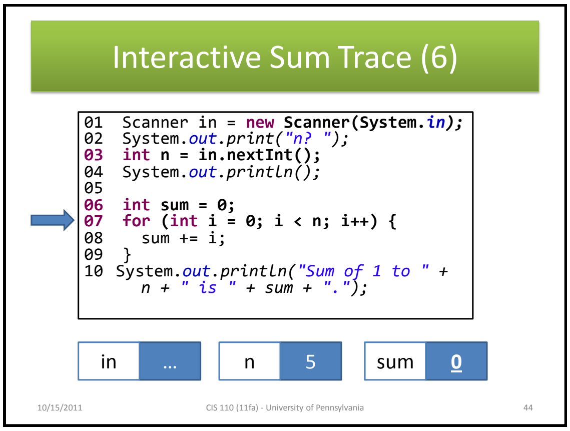
n 5



Interactive Sum Trace (5)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
   n + " is " + sum + ".");
```

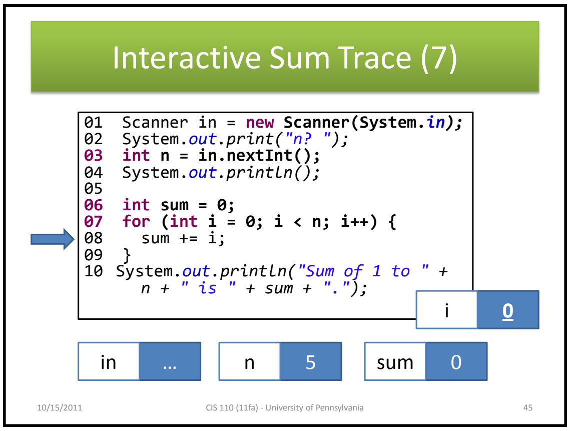
in	…	n	5



Interactive Sum Trace (6)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
   n + " is " + sum + ".");
```

in	…	n	5	sum	0

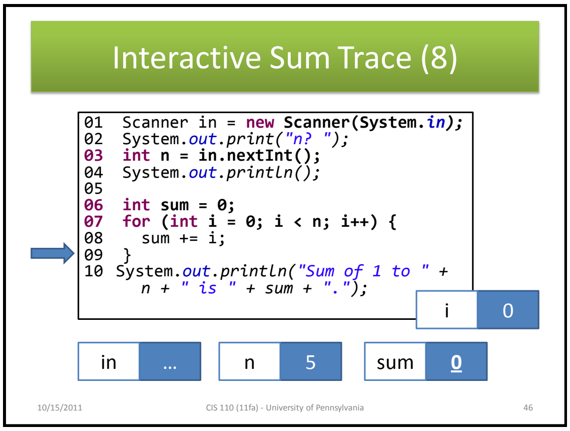


Interactive Sum Trace (7)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
   n + " is " + sum + ".");
```

i	0

in	…	n	5	sum	0

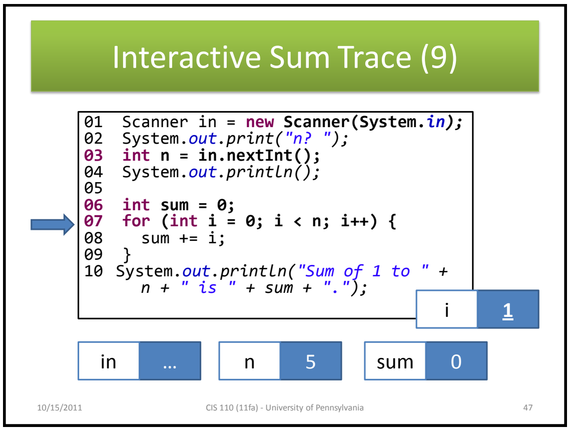


Interactive Sum Trace (8)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
   n + " is " + sum + ".");
```

i	0

in	…	n	5	sum	0

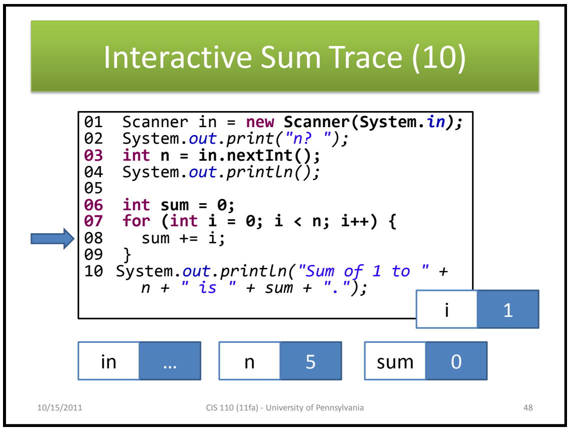


Interactive Sum Trace (9)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
   n + " is " + sum + ".");
```

i	1

in	…	n	5	sum	0

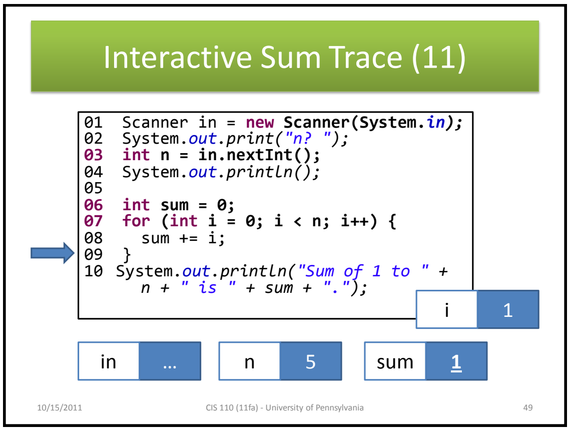


Interactive Sum Trace (10)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
  sum += i;
}
System.out.println("Sum of 1 to " +
   n + " is " + sum + ".");
```

i	1

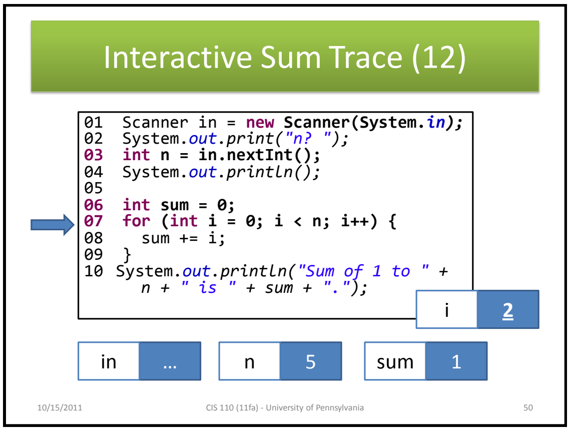
in	…	n	5	sum	0



Interactive Sum Trace (11)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
   sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

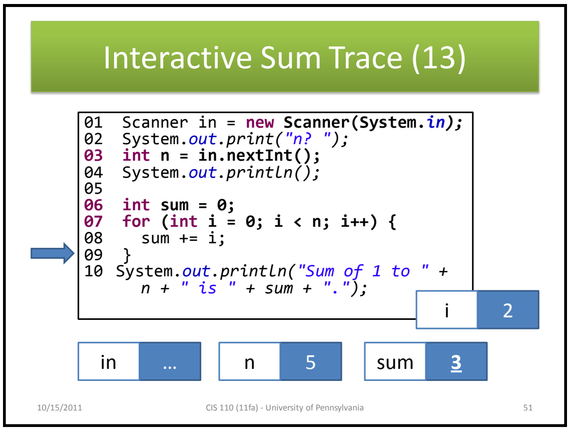
i	1						
in	...	n	5	sum	1		



Interactive Sum Trace (12)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
   sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

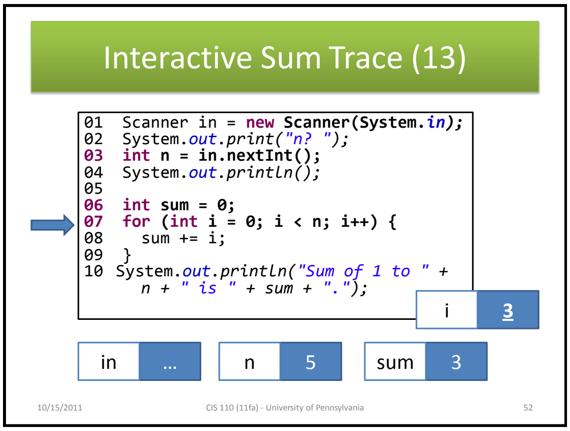
i	2				
in	...	n	5	sum	1



Interactive Sum Trace (13)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
   sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

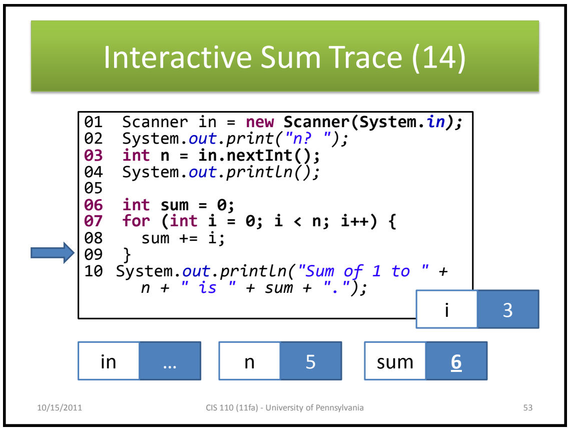
i	2				
in	...	n	5	sum	3



Interactive Sum Trace (13)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
   sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

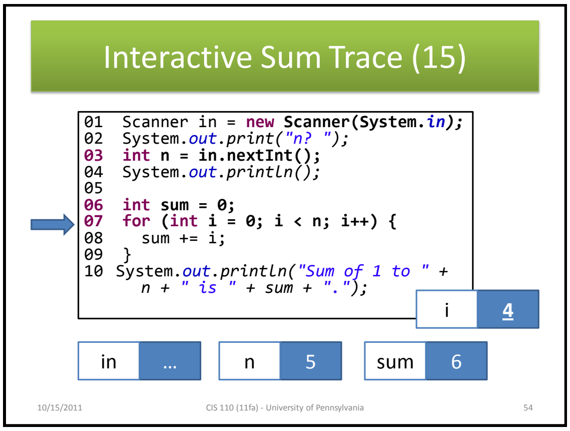
i	3				
in	...	n	5	sum	3



Interactive Sum Trace (14)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
   sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

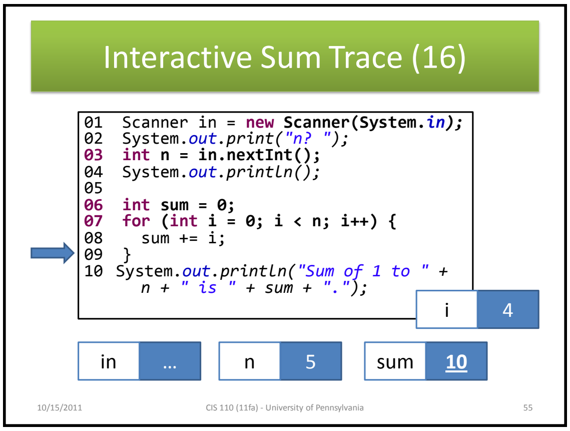
i	3				
in	...	n	5	sum	6



Interactive Sum Trace (15)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
   sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

i	4				
in	...	n	5	sum	6



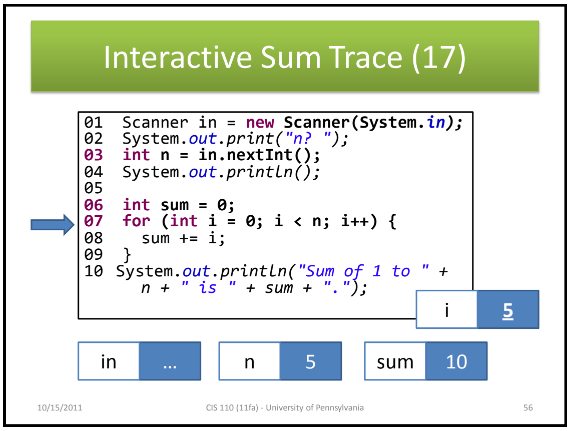
Interactive Sum Trace (16)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
   sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

i 4

in ... n 5 sum 10

10/15/2011 CIS 110 (11fa) - University of Pennsylvania 55



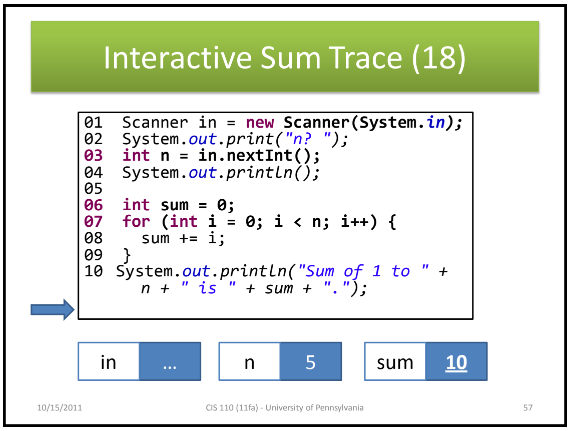
Interactive Sum Trace (17)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
   sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

i 5

in ... n 5 sum 10

10/15/2011 CIS 110 (11fa) - University of Pennsylvania 56



Interactive Sum Trace (18)

```
Scanner in = new Scanner(System.in);
System.out.print("n? ");
int n = in.nextInt();
System.out.println();

int sum = 0;
for (int i = 0; i < n; i++) {
   sum += i;
}
System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

in ... n 5 sum 10

10/15/2011 CIS 110 (11fa) - University of Pennsylvania 57