

CIS 110: Introduction to Computer Programming

Lecture 11

Text Processing and More On Design (§ 4.2-4.3)

Outline

- More on Cumulative Algorithms
- Processing Text
- Tackling Programming Problems

More Cumulative Algorithms

Review: Interactive Sum

- Can you write a program that computes the sum of numbers from 0 to the user's input minus 1?

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

Some storage declared outside the loop.

```
int sum = 0;
```

```
for (int i = 0; i < n; i++) {
```

```
    sum += i;
```

Update the storage inside of the loop.

```
}  
System.out.println(  
    "Sum of 1 to " + n + " is " + sum + ".");
```


Use the updated storage after the loop.

Interactive Sum Trace (1)



```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```


Interactive Sum Trace (2)



```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

in	...
----	-----


Interactive Sum Trace (3)



```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

in	...
----	-----

Interactive Sum Trace (4)



```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

in	...
----	-----

n	<u>5</u>
---	----------

Interactive Sum Trace (5)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```

in

...

n

5

Interactive Sum Trace (6)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



in	...
----	-----

n	5
---	---

sum	<u>0</u>
-----	----------

Interactive Sum Trace (7)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



i	<u>0</u>
---	----------

in	...
----	-----

n	5
---	---

sum	0
-----	---

Interactive Sum Trace (8)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



i	0
---	---

in	...
----	-----

n	5
---	---

sum	<u>0</u>
-----	----------

Interactive Sum Trace (9)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



i	<u>1</u>
---	----------

in	...
----	-----

n	5
---	---

sum	0
-----	---

Interactive Sum Trace (10)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



i	1
---	---

in	...
----	-----

n	5
---	---

sum	0
-----	---

Interactive Sum Trace (11)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



i	1
---	---

in	...
----	-----

n	5
---	---

sum	<u>1</u>
-----	----------

Interactive Sum Trace (12)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



i

2

in

...

n

5

sum

1

Interactive Sum Trace (13)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



i	2
---	---

in	...
----	-----

n	5
---	---

sum	<u>3</u>
-----	----------

Interactive Sum Trace (13)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



i	3
---	---

in	...
----	-----

n	5
---	---

sum	3
-----	---

Interactive Sum Trace (14)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



i	3
---	---

in	...
----	-----

n	5
---	---

sum	<u>6</u>
-----	----------

Interactive Sum Trace (15)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



i

4

in

...

n

5

sum

6

Interactive Sum Trace (16)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



i	4
---	---

in	...
----	-----

n	5
---	---

sum	<u>10</u>
-----	-----------

Interactive Sum Trace (17)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



i	5
---	---

in	...
----	-----

n	5
---	---

sum	10
-----	----

Interactive Sum Trace (18)

```
01 Scanner in = new Scanner(System.in);
02 System.out.print("n? ");
03 int n = in.nextInt();
04 System.out.println();
05
06 int sum = 0;
07 for (int i = 0; i < n; i++) {
08     sum += i;
09 }
10 System.out.println("Sum of 1 to " +
    n + " is " + sum + ".");
```



in	...
----	-----

n	5
---	---

sum	<u>10</u>
-----	-----------

Cumulative Algorithms and Conditionals

- Problem: calculate the minimum of 10 numbers entered by the user.

```
public static int promptForNumber(Scanner in, int i) {
    System.out.print("Enter number " + i + ": ");
    return in.nextInt();
}

public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    int min = promptForNumber(in, 1);           // Storage
    for (int i = 2; i <= 10; i++) {
        int num = promptForNumber(in, i);     // Modify storage
        if (num < min) {
            min = num;
        }
    }
    // Use storage
    System.out.println("The minimum is " + min + ".");
}
```


Text Processing

Recall: charAt(index)

```
Scanner in = new Scanner(System.in);
String line = in.nextLine();
for (int i = 0; i < line.length(); i++) {
    // charAt(i) returns the ith character of the string
    // Remember string indicies start at 0...
    System.out.println(line.charAt(i));
}
```

The char Type

- A *primitive* that represents a single character.
 - Represented as a *16-bit* integer (*character value*).
 - All our primitive operations on numbers apply!

```
char c1 = 'c';
if (c1 == 'c') {
    System.out.println(c1 + " is c!");
}
// Int + Char = Int which is 32 bits, so we need to
// cast to tell the compiler we don't mind the truncation.
char c2 = (char) (c1 + 5);
System.out.println("5 chars from " + c1 + " is " + c2);
// Tests to see if c2 is a lowercase number.
if (c2 >= 'a' || c2 <= 'z') {
    System.out.println(c2 + " is between 'a' and 'z'!");
}
```

The Character class

- Helpful static methods for dealing with chars.

```
int five = Character.getNumericValue('5');
System.out.println("The number value of 5 is " + five);
System.out.println("Is 6 a digit? " + Character.isDigit('6'));
System.out.println("Is q a letter? " + Character.isLetter('q'));
System.out.println("Is z lowercase? " + Character.isLowerCase('z'));
System.out.println("Is R uppercase? " + Character.isUpperCase('R'));
System.out.println("Lowercase Z is " + Character.toLowerCase('Z'));
System.out.println("Uppercase a is " + Character.toUpperCase('a'));
```

Cumulative Text Algorithms

- Similar to our numeric cumulative algorithms.

```
// Echoes the line of text, alternating capital case of the input.
Scanner in = new Scanner(System.in);
System.out.print("Enter a line of text: ");
String line = in.nextLine();
String result = "";
for (int i = 0; i < line.length(); i++) {
    char letter = line.charAt(i);
    if (i % 2 == 0) {
        letter = Character.toUpperCase(letter);
    }
    result += letter;
}
System.out.println(result);
```

System.out.printf

- An alternative to println/print that lets you *format* the output.

A *format specifier*.
A placeholder for a
thing to print.

```
System.out.printf("Example of printf: %d %.2f %s",  
                 12, 1.241, "Chowder");  
> Example of printf: 12 1.24 Chowder
```

Specifiers have the form:
%<formatting><type>

Need to provide one argument
per format specifier. They are
consumed in-order.

See p. 260 of the book for
more information about
format specifiers.

System.out.printf (example)

```
Scanner in = new Scanner(System.in);
for (int i = 0; i < 3; i++) {
    System.out.print("Enter the employee's name: ");
    String name = in.nextLine();
    System.out.print("Enter the employee's salary: ");
    double salary = in.nextDouble();
    System.out.printf("%10s -> %6.2f\n", name, salary);
    in.nextLine();// flush the buffer
}
```

```
Enter the employee's name: Joe
Enter the employee's salary: 1410.14
    Joe -> 1410.14
Enter the employee's name: Lamont
Enter the employee's salary: 4104.41
    Lamont -> 4104.41
Enter the employee's name: Fred
Enter the employee's salary: 10041.4190482
    Fred -> 10041.42
```

Tackling Program Problems

See `PalindromeChecker.java`

Sample Problem

- Problem: write a program that reads in a String from the user, checks to see if that String is a palindrome, and informs the user of the results of the check.

Example output

```
> Enter a string to check:  
> abba  
> The reverse of the line is: abba  
> The line is a palindrome!
```

Our Methodology

1. Try some example inputs to get a feel for the problem.
2. Start with a skeleton of the solution.
3. Decompose the problem into sub-problems.
4. Make helper methods to solve the sub-problems.
5. Use those helper methods to solve your main problem.