

Announcements/Follow-ups

- Lec04
 - Execution flow with method calls
 - Switch statements (days in month example)
- Lab02
 - Scanner `nextLine()` and buffers
 - Multiple calls to `Math.random()`
- Keep an eye out for Project 1 in the next day or two
- Email and Piazza
- 30 minute quiz tomorrow (study answers posted)
- Midterm #1 next Friday
 - Hardware/software, Data types, expressions, control flow, basics of object oriented programming
 - For DSS accommodations or religious obligations let me know by this Friday, as per the syllabus

Control Flow

Loops

Hard-coding iteration

```
//Print "Hello" 100 times  
1.  System.out.println("Hello");  
2.  System.out.println("Hello");  
3.  System.out.println("Hello");  
...  
98. System.out.println("Hello");  
99. System.out.println("Hello");  
100. System.out.println("Hello");
```

Hard-coding iteration

```
//Print the 1st 100 odds  
1.  System.out.println(1);  
2.  System.out.println(3);  
3.  System.out.println(5);  
...  
98. System.out.println(195);  
99. System.out.println(197);  
100. System.out.println(199);
```

Hard-coding iteration

```
//Add the 1st 100 odds
int sum = 0;
1.    sum += 1;
2.    sum += 3;
3.    sum += 5;
...
98.   sum += 195;
99.   sum += 197;
100.  sum += 199;
```

Hard-coding iteration

- Tedious (at best) to write
 - `//Print 1st 1,000,000,000 odds`
- Number of iterations may be unknown at compile time
 - `//Print 1st sc.nextInt() odds`
- Bloats the executable file
- Fast

Enter loops

- How can we abbreviate the iterative process?
 - **Where to start counting**
 - When to stop counting
 - What to do at each step

Counter:	0		1		2		3		4		5		6
Odds:	1	+	3	+	5	+	7	+	9	+	11	+	13
Sum:	1		4		9		16		25		36		49

Enter loops

- How can we abbreviate the iterative process?
 - Where to start counting
 - **When to stop counting**
 - What to do at each step

Counter:	0		1		2		3		4		5		6
Odds:	1	+	3	+	5	+	7	+	9	+	11	+	13
Sum:	1		4		9		16		25		36		49

Enter loops

- How can we abbreviate the iterative process?
 - Where to start counting
 - **(When to keep counting)**
 - What to do at each step

Counter:	0		1		2		3		4		5		6
Odds:	1	+	3	+	5	+	7	+	9	+	11	+	13
Sum:	1		4		9		16		25		36		49

Enter loops

- How can we abbreviate the iterative process?
 - Where to start counting
 - (When to keep counting)
 - **What to do at each step**

Counter:	0		1		2		3		4		5		6
Odds:	1	+	3	+	5	+	7	+	9	+	11	+	13
Sum:	1		4		9		16		25		36		49

Enter loops

- How can we abbreviate the iterative process?
 - Where to start counting
 - (When to keep counting)
 - **What to do at each step**

Counter: 0 1 2 3 4 5 6



Odds: **1 + 3 + 5 + 7 + 9 + 11 + 13**

Enter loops

- How can we abbreviate the iterative process?
 - Where to start counting
 - (When to keep counting)
 - **What to do at each step**

Counter: 0 1 2 3 4 5 6



$2 * \text{counter} + 1$

Odds: **1 + 3 + 5 + 7 + 9 + 11 + 13**

Enter loops

- How can we abbreviate the iterative process?
 - How to count

		+1		+1		+1		+1		+1		+1	
Counter:	0		1		2		3		4		5		6
Odds:	1	+	3	+	5	+	7	+	9	+	11	+	13
Sum:	1		4		9		16		25		36		49

“For Loops”

```
for(init; condition; update) {  
    do something  
}
```

- *Init* : initialize a counter variable
- *condition* : if this condition is true, repeat the loop body
- *Update* how to change the counter variable for the next iteration
- *do something* : execute a sequence of commands (typically involving the counter)

“For Loops”

//Typical

```
for(int i = start; i < stop; i++) {  
    doSomething(i);  
}
```

//Atypical (but valid)

```
for(float f = 1.0f; f > 1e-9; f *= 0.1) {  
    doSomething(f);  
}
```

Loops and Sums

$$\sum_{i=0}^{99} 2i + 1$$

```
int sum = 0;
for(int i = 0; i < 100; i++) {
    int odd = 2*i + 1;
    sum += odd;
}
```

Execution flow

```
int sum = 0;
for(int i = 0; i < 100; i++) {
    int odd = 2*i + 1;
    sum += odd;
}
System.out.println(sum);
```

Execution flow

```
int sum = 0;
```

```
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}
```

```
System.out.println(sum);
```

sum

0

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 0

i 0

Execution flow



```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 0

i 0

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 0

i 0

odd 1

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 1 i 0 odd 1

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 1

i 1

Execution flow



```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 1

i 1

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 1 i 1 odd 3

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 4 i 1 odd 3

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 4

i 2

Execution flow



```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 4

i 2

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 4

i 2

odd 5

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 9 i 2 odd 5

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 9

i 3

Execution flow



```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 9

i 3

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 9

i 3

odd 7

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 16 i 3 odd 7

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 16

i 4

Execution flow

```
int sum = 0;
for(int i = 0; i < 100; i++) {
    int odd = 2*i + 1;
    sum += odd;
}
System.out.println(sum);
```

sum 16

i 4

Execution flow



```
int sum = 0;
for(int i = 0; i < 100; i++) {
    int odd = 2*i + 1;
    sum += odd;
}
System.out.println(sum);
```

sum

9801

i

99

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum

9801

i

99

odd

199

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum

10000

i

99

odd

199

Execution flow

```
int sum = 0;
for(int i = 0; i < 100; i++) {
    int odd = 2*i + 1;
    sum += odd;
}
System.out.println(sum);
```

sum

10000

i

100

Execution flow



```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum 10000

i 100

Execution flow

```
int sum = 0;  
for(int i = 0; i < 100; i++) {  
    int odd = 2*i + 1;  
    sum += odd;  
}  
System.out.println(sum);
```

sum

10000

i

100

Examples

- Print (sum) the first 10 integers
- Print the full set of signed byte values
- Print the full Unicode alphabet
- Print an integer in base 2
- Check if an integer is prime