CS 112 Introduction to Programming

Loop Examples and Coding Style;
Variable Scoping;
Nested Loops;

Yang (Richard) Yang
Computer Science Department
Yale University
208A Watson, Phone: 432-6400
Email: yry@cs.yale.edu
Outline

- Admin and recap
- for loops
  - basic syntax
  - CountDown as as loop and coding style example
  - variable scoping of loop variables
  - nested loops
Admin

- PS2 questions
- Coding style review?
Recap

- Primitive data types
  - why data types?
  - storage and representation
  - operations
  - expressions
  - data conversions
  - assignment as operator

- for loops
Recap: The for Statement: Syntax

The initialization portion is executed once before the loop begins.

The statement is executed until the condition becomes false.

Both semi-colons are always required.

The increment portion is executed at the end of each iteration.

Reserved word

for ( initialization ; condition ; increment )

statement;

The for statement: Syntax
Recap: Flexibility of for Loop with Counter

Loop counter:
- can use any name, not just i
- can start at any value, not just 1
- only valid in the loop

Compare loop counter with target:
- < less than
- <= less than or equal to
- > greater than
- >= greater than or equal to

Can increment, decrement, times, ...

```java
for (int i = 1; i <= 6; i ++)
{
    System.out.println("I am so smart");
}
```

Can be even empty
Example for Loop: Counting Down

- Write a program generating output
  T-minus 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, blastoff!

- Requirement: loop counter starts with 10 and counts down
Counting Down v1

- The **update** uses -- to count down.

```java
System.out.print("T-minus ");
for (int i = 10; i >= 1; i--) {
    System.out.print(i + ", ");
}
System.out.println("blastoff!");
```
Outline

- Admin and recap
- **for loops**
  - basic syntax
  - `CountDown` as as loop and coding style example
  - variable scoping of loop variables
  - nested loops
Counting Down v2

- Requirement: loop counter starts with 1 and counts up:

  T-minus 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, blastoff!

  System.out.println("T-minus ");
  for (int i = 1; i <= 10; i++) {
      // ???
  }
  System.out.println("blastoff!");
# Mapping Loop# to Target Pattern

*y = 11 - x*

<table>
<thead>
<tr>
<th>i</th>
<th>number to print</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Graph showing the relationship between Loop# (i) and the number to print (y).
Counting Down

System.out.print("T-minus ");
for (int i = 1; i <= 10; i++) {
    System.out.println(11 - i + ", ");
}
System.out.println("blastoff!");

y = 11 - x
An “IQ Test” Format

Loop
# i:

1  2  3  4  5  6  7  8  9  10
10  9  8  7  6  5  4  3  2  1

-1  -1  -1  -1  -1  -1  -1  -1  -1  -1

slope
An “IQ Test” Format

Loop # i: Value at 0

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

-1  -1  -1  -1  -1  -1  -1  -1  -1  -1

slope
An “IQ Test” Format

Loop # i:

Value at 0

11 10 9 8 7 6 5 4 3 2 1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

slope
An “IQ Test” Format

Loop # i:
Value at 0

1  2  3  4  5  6  7  8  9  10

11 10  9  8  7  6  5  4  3  2  1

-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

$y = 11 - 1 \times x$

slope
Practice: Mapping loop# to numbers

for (int count = 1; count <= 5; count++) {
    System.out.print( ... );
}

- What statement in the body would cause the loop to print:
  17 13 9 5 1
**Mapping loop# to numbers**

Loop# i: 1 2 3 4 5  
Target: 21 17 13 9 5 1  

21-4*i \(-4\ -4\ -4\ -4\ -4\)

```java
for (int count = 1; count <= 5; count++) {
    System.out.print(-4 * count + 21 + " ");
}
```
(Offline) Practice: Mapping loop# to numbers

for (int count = 1; count <= 5; count++) {
    System.out.print( ... );
}

○ What statement in the body would cause the loop to print:

4 7 10 13 16

for (int count = 1; count <= 5; count++) {
    System.out.print(3 * count + 1 + " ");
}
(Offline) Practice: Mapping loop# to numbers

for (int count = 1; count <= 5; count++) {
    System.out.print( ... );
}

- What statement in the body would cause the loop to print:
  2 7 12 17 22

for (int count = 1; count <= 5; count++) {
    System.out.print(5 * count - 3 + " ");
}
Counting Down v2a

- Requirement: loop counter starts with 0 and counts up:

  T-minus 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, blastoff!

  System.out.print("T-minus ");
  for (int i = 0; i __ 10; i++) {
      System.out.println(_____ + ", ");
  }
  System.out.println("blastoff!");
An “IQ Test” Format

Loop

# i:

0 1 2 3 4 5 6 7 8 9

10 9 8 7 6 5 4 3 2 1

-1 -1 -1 -1 -1 -1 -1 -1 -1 -1

Already knows value at 0
Counting down or up, starting w/ 0 or 1 is mostly a personal style.

```java
System.out.print("T-minus ");
for (int i = 10; i >= 1; i--) {
    System.out.print(i + ", ");
}
System.out.println("blastoff!");

System.out.print("T-minus ");
for (int i = 1; i <= 10; i++) {
    System.out.print(i + ", ");
}
System.out.println("blastoff!");

System.out.print("T-minus ");
for (int i = 0; i < 10; i++) {
    System.out.print(10-i + ", ");
}
System.out.println("blastoff!");
```
If I want to count down from 12, what should we change?

- T-minus 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, blastoff!

```java
System.out.print("T-minus ");
for (int i = 1; i <= 10; i++) {
    System.out.print(11-i + ", ");
}
System.out.println("blastoff!");
```
Problem: The code has two “magic” numbers 11 and 10, but they are not independent

```java
System.out.print("T-minus ");
for (int i = 1; i <= 10; i++) {
    System.out.print(11-i + ", ");
}
System.out.println("blastoff!");
```
int N = 10;
System.out.print("T-minus ");
for (int i = 1; i <= N; i++) {
    System.out.print(N+1-i + "", "");
}
System.out.println("blastoff!");

Minimize # of magic numbers (make change easier).
Does the following program give the correct countdown?:

```java
int N = 10;
System.out.print("T-minus ");
for (int i = 1; i <= N; N++) {
    System.out.print(N+1-i + "", "");
}
System.out.println("blastoff!");
```
Counting Down: v4

Does the following program give the correct countdown?

```java
int N = 10;
System.out.print("T-minus ");
for (int i = 1; i <= N; N++) {
    System.out.print(N+1-i + ", ");
}
System.out.println("blastoff!");
```

Answer: No. There is a typo (N for i)

Q: can the computer help me to find it (read my mind?)
Constant

- Use keywords to tell computer your intention
- If there is a `final` before a variable declaration, it is your promise to the computer that you will not modify it after declaration
- If you break your promise, the compiler will catch you
Good Program Style

```java
final int N = 10;
System.out.print("T-minus ");
for (int i = 1; i <= N; i++) {
    System.out.print(N+1-i + " , ");
}
System.out.println("blastoff!");
```
Outline

- Admin and recap
- for loops
  - basic syntax
  - CountDown as as loop and coding style example
  - variable scoping of loop variables
  - nested loops
Counting Down: Code Puzzle

What if we want to print out the values of N after the loop:

```java
final int N = 10;
System.out.print("T-minus ");
for (int i = 1; i <= N; i++) {
    System.out.print(N+1-i + ", ");
}
System.out.println("blastoff! ");
System.out.println("N = " + N); //?
```
Counting Down: Code Puzzle

What if we want to print out the values of \( i \) after the loop:

```java
final int N = 10;
System.out.print("T-minus ");
for (int i = 1; i <= N; i++) {
    System.out.print(N+1-i + ", ");
}
System.out.println("blastoff!");

System.out.println("Final i =" + i); //?
```
Counting Down: Code Puzzle

% javac CountDownValue.java
CountDownValue.java:25: cannot find symbol
  symbol   : variable i
  location: class CountDownValue
             System.out.println( "Final i = " + i );
                      ^
  1 error
Variable Scope

- **Scope**: The part of a program where a variable exists.

- **Basic rule**: from its declaration to the end of the enclosing `{ }` braces

- **Examples**
  - A variable declared in a for loop exists only in that loop.
  - A variable declared in a specific method exists only in that method.
  - A variable declared not inside any method but in a class is said to have class scope.
public class CountSum {
    static int N = 10;
    public static void main(String[] args) {
        countSum();
    }

    public static void countSum() {
        System.out.print("T-minus ");
        int sum = 0;
        for (int i = 1; i <= N; i++) {
            System.out.println(i);
            sum += i;
        }
        System.out.println("N: \+ N");
        System.out.println("Sum: \+ sum");
    } // end of countSum
} // end of class
Why Scope?

- Encapsulation
  - e.g., different methods can use the same variable name without the need for coordination
  - many analogies: folders allow same file name so long in different folders

```java
public static void aMethod() {
    int x = 1;
    ...
}
```

```java
public static void bMethod() {
    int x = 2;
    ...
}
```
Loop Example

Does the following code work?

```java
public static void main() {
    final int N = 10;
    for (int i = 1; i <= N; i++) {
        System.out.print(N+1-i + " ");
    }
    System.out.println();

    for (int i = 1; i <= N; i++) {
        System.out.print(N+1-i + " ");
    }
}
```

Output:

```
10 9 8 7 6 5 4 3 2 1
10 9 8 7 6 5 4 3 2 1
```
Does the following code work?

```java
public static void main() {
    final int N = 10;
    int i;
    for (i = 1; i <= N; i++) {
        System.out.print( N+1-i + " " );
    }
    System.out.println();

    for (int i = 1; i <= N; i++) {
        System.out.print( N+1-i + " ");
    }
}
```
Loop Example

Does the following code work?

```java
public static void main() {
    final int N = 10;
    int i;
    for (i = 1; i <= N; i++) {
        System.out.print( N+1-i + " ");
    }
    System.out.println();

    for (i = 1; i <= N; i++) {
        System.out.print( N+1-i + " ");
    }
}
```
public static void main() {
    final int N = 10;
    int i;
    for (i = 1; i <= N; i++) {
        System.out.print( N+1-i + " ");
    }
    System.out.println();
    for (i = 1; i <= N; i++) {
        System.out.print( N+1-i + " ");
    }
}

Personally, I prefer this one as it declares variables with minimal scope.
Loop Example

Does the following code work?

```java
for (int set = 1; set <= 5; set++) {
    for (int rps = 1; rps <= set; rps++) {
        System.out.print("*");
    }
    System.out.println();
}
```

Output:

```
* 
** 
*** 
**** 
***** 
```
Outline

- Admin and recap
- for loops
  - basic syntax
  - CountDown as as loop and coding style example
  - variable scoping of loop variables
  - nested loops
Nested Loop

```java
for (int set = 1; set <= 5; set++) {
    for (int rps = 1; rps <= set; rps++) {
        System.out.print("*");
    }
    System.out.println();
}
```

- There can be another loop inside one loop to form a nested loop.
- The #loop times of the inner loop can depend on the outer loop variable.
- Nested loops are common, important programming patterns.
Practice: Nested for loop example

What is the output of the following nested for loops?

```java
for (int i = 1; i <= 5; i++) {
    for (int j = 1; j <= i; j++) {
        System.out.print(i);
    }
    System.out.println();
}
```

Output:

```
1
22
333
4444
55555
```
What is the output of the following nested for loops?

```java
for (int i = 1; i <= 9; i++) {
    for (int j = 1; j <= i; j++) {
        System.out.print(i*j + "\t");
    }
    System.out.println();
}
```
Nested Loop Design Example: Drawing A Complex Pattern

- Use nested `for` loops to draw ASCII X
- A size SIZE ASCII X has 2 * SIZE rows
- Why draw ASCII art?
  - Real graphics will require some finesse (shortly)
  - ASCII art has complex patterns
  - Can focus on the algorithms