

Unit 8

- Introduction to `for` loops



The `for` Loop

- `for` is a loop that combines in one statement initialization, condition, and change:

```
for ( initialization; condition; change )  
{  
    statement1;  
    statement2;  
    ...  
    statementN;  
}
```



The `for` Loop

- for loops are great for repeating something a certain amount of times



The for Loop (cont'd)

- Java allows you to declare the loop control variable in the for statement itself. For example:

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

i is undefined outside
the loop



The for Loop (cont'd)

- “Repeat n times” idiom:

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

or

```
for (int count = 1; count <= n ; count++)  
{  
    ...  
}
```



The for Loop (cont'd)

- Example:

```
// Returns the sum of the first
// n integers
public int sumAll (int n)
{
    int sum = 0;
    for (int p = 1; p <=n; p++)
    {
        sum += p;
    }
    return sum;
}
```

Initialization

Condition

Change



The for Loop (cont'd)

- Example: What does this method compute?

```
public int mystery (int n)
{
    int m = 1;

    for (int k = 2; k <= n; k++)
        m *= k;

    return m;
}
```



break and return in Loops

- break in a loop instructs the program to immediately quit the current iteration and go to the first statement following the loop.
- return in a loop instructs the program to immediately quit the current method and return to the calling method.
- A break or return must be inside an if or an else, otherwise the code after it in the body of the loop will be unreachable.



break in Loops

- Example:

```
int d = n - 1;

while (d > 0)
{
    if (n % d == 0)
        break;
    d--;
}

if ( d > 0 ) // if found a divisor
    ...
```



Nested Loops

- A loop within a loop is called nested.

```
// Draw a 5 by 3 grid:
```

```
for (int x = 0; x < 50; x += 10)
{
    for (int y = 0; y < 30; y += 10)
    {
        g.fillRect(x, y, 8, 8);
    }
}
```

