University of British Columbia
CPSC 111, Intro to Computation
2009W2: Jan-Apr 2010

Tamara Munzner

Loops III

Lecture 19, Wed Mar 3 2010

borrowing from slides by Kurt Eiselt

http://www.cs.ubc.ca/~tmm/courses/111-10

Review: For Statement

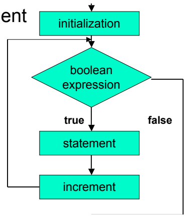
```
public class ForDemo
{
  public static void main (String[] args)
  {

    for (int counter = 1; counter <= 3; counter = counter + 1)
    {
      System.out.println("The square of " + counter +
                         " is " + (counter * counter));
    }
    System.out.println("End of demonstration");
  }
}
```

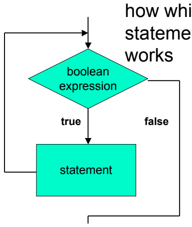
- Header has three parts, separated by semicolons
  - first: initialization: executed only one time, at start
  - second: boolean expression: evaluated just before loop body, like in **while**
  - third: increment: executed at end of loop body, arbitrary calculation allowed

For Versus While Statement

how for statement works



how while statement works

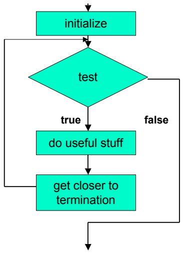


- flowcharts can be somewhat deceptive
  - need initialization and incrementing/modifying in while loop too
  - although syntax does not require it in specific spot

For Versus While Statement

- Anything that can be done with one type of loop can be done with another
  - **for** and **while** are equivalent
- **For** statement convenient when
  - loop should be executed specific number of times
  - number can be determined before loop starts
- **While** statement convenient when
  - don't know yet how many times to execute loop body
  - but can check if it's time to end loop as you go

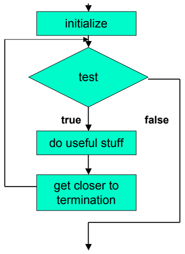
Four Things Needed In Any Loop



- Give starting values to one or more variables used in loop

how loops work in general

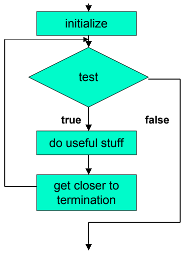
Four Things Needed In Any Loop



- Give starting values to one or more variables used in loop
- Test to see when looping stops

how loops work in general

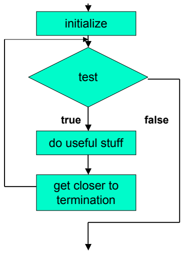
Four Things Needed In Any Loop



- Give starting values to one or more variables used in loop
- Test to see when looping stops
- One or more useful operations here

how loops work in general

Four Things Needed In Any Loop



- Give starting values to one or more variables used in loop
- Test to see when looping stops
- One or more useful operations here
- Change something to move process closer termination

how loops work in general

Yet Another Loop Statement

```
public class WhileDemo
{
  public static void main (String[] args)
  {
    int limit = 3;
    int counter = 1;

    while (counter <= limit)
    {
      System.out.println("The square of " + counter +
                         " is " + (counter * counter));
      counter = counter + 1;
    }
    System.out.println("End of demonstration");
  }
}
```

- **while** version

Yet Another Loop Statement

```
public class ForDemo
{
  public static void main (String[] args)
  {
    for (int counter = 1; counter <= 3; counter = counter + 1)
    {
      System.out.println("The square of " + counter +
                         " is " + (counter * counter));
    }
    System.out.println("End of demonstration");
  }
}
```

- **for** version

Yet Another Loop Statement

```
public class DoDemo
{
  public static void main (String[] args)
  {
    int limit = 3;
    int counter = 1;

    do
    {
      System.out.println("The square of " + counter +
                         " is " + (counter * counter));
      counter = counter + 1;
    } while (counter <= limit);

    System.out.println("End of demonstration");
  }
}
```

- **do** version

Do Statement

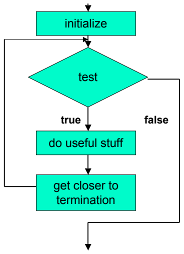
```
public class DoDemo
{
  public static void main (String[] args)
  {
    int limit = 3;
    int counter = 1;

    do
    {
      System.out.println("The square of " + counter +
                         " is " + (counter * counter));
      counter = counter + 1;
    } while (counter <= limit);

    System.out.println("End of demonstration");
  }
}
```

- **do** version: not quite equivalent
  - termination test at end, so body executed at least once

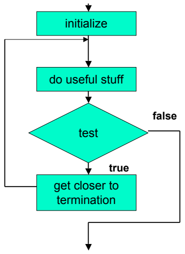
Four Things Needed In Any Loop



- Give starting values to one or more variables used in loop
- Test to see when looping stops
- One or more useful operations here
- Change something to move process closer termination

how loops work in general

Do Statement



- Body always executed at least once

order of four things can change, but need them all

Nested Loops

- Very simple for loop

```
public class SimpleLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      System.out.println(i);
    }
  }
}
```

- What does it do?

Nested Loops

- Very simple for loop

```
public class SimpleLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      System.out.println(i);
    }
  }
}
```

- What does it do? Prints

```
1
2
3
```

Nested Loops

- Very simple for loop

```
public class SimpleLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      System.out.println(i);
    }
  }
}
```

- What if for every number below, want multiplication table of value times 2, x3, etc?

```
1 2 3
2 4 6
3 6 9
```

Nested Loops

- Very simple for loop

```
public class SimpleLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      System.out.println(i);
    }
  }
}
```

- For every number printed by loop above

```
1 2 3
2 4 6
3 6 9
```

Nested Loops

- Very simple for loop

```
public class SimpleLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      System.out.println(i);
    }
  }
}
```

- For every number printed by loop above
  - need another loop to print numbers in row

```
1 2 3
2 4 6
3 6 9
```

Nested Loops

- Very simple for loop

```
public class SimpleLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      System.out.println(i);
    }
  }
}
```

- For every number printed by loop above
  - need another loop to print numbers in row

```
1 2 3
2 4 6
3 6 9
```

How do we do that?

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 1

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 1

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 1

j 1

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 1

j 1

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 1

j 1

```
1_
```

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 1

j 2

```
1_
```

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 1

j 2

```
1_
```

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 1

j 2

```
1  2_
```

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 1

j 3

```
1  2_
```

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 1

j 3

```
1  2_
```

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 1

j 3

```
1  2  3_
```

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 1

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 1

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 1

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3
_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 2

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3
_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 2

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3
_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 2

j 1

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3
_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 2

j 1

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3
_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 2

j 1

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3
2_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 2

j 2

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3
2_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 2

j 2

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3
2_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 2

j 1

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3
2   4_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 2

j 3

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3
2   4_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 2

j 3

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3
2   4_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 2

j 3

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3
2   4   6_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 2

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3
2   4   6_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 2

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1   2   3
2   4   6_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 2

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 3

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 3

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 3

j 1

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 3

j 1

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 3

j 1

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
3_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 3

j 2

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
3_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 3

j 2

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
3_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 3

j 2

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
3  6_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 3

j 3

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
3  6_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 3

j 3

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
3  6_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 3

j 3

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
3  6  9_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 3

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
3  6  9_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 3

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
3  6  9_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 3

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
3  6  9
_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 4

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
3  6  9
_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 4

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
3  6  9
_
```

Nested Loops

- Put a loop inside a loop
  - trace to see how it works

i 4

j 4

```
public class NestedLoop
{
  public static void main (String[] args)
  {
    for (int i = 1; i <= 3; i++)
    {
      for (int j = 1; j <= 3; j++)
      {
        System.out.print((i * j) + "  ");
      }
      System.out.println();
    }
  }
}
```

```
1  2  3
2  4  6
3  6  9
_
```

Exit!

Practice Problem

- Write program using loop to simulate flipping a coin one million times
  - keep track of how many times it's heads up and how many heads down
  - print results
- Make version for each loop type
  - `while`, `for`, `do`