Four Things Needed In Any Loop

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

Four Things Needed In Any Loop

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

Do Statement

- Body always executed at least once

Nested Loops

- Very simple for loop

Nested Loops

- Very simple for loop

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
- Test to see when looping stops
- Do useful stuff
- Change something to move process closer termination

Yet Another Loop Statement

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

 Yet Another Loop Statement

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

Do Statement

- Body always executed at least once

Order of four things can change, but need them all

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

While

- statement convenient when
  - loop should be executed specific number of times
  - number can be determined before loop starts

For

- 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

Do

- 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
**Nested Loops**

- Very simple for loop

```java
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**

- Put a loop inside a loop

```java
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();
        }
    }
}
```

- Need another loop to print numbers in row
1 2 3
2 4 6
3 6 9

**Nested Loops**

- Put a loop inside a loop

```java
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();
        }
    }
}
```

- Trace to see how it works

**Nested Loops**

- Put a loop inside a loop

```java
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();
        }
    }
}
```

- Trace to see how it works
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

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

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

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

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();
        }
    }
}

System.out.println();
Nested Loops

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

```java
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();
        }
    }
}
```

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