



Reading

- This week: Chapter 6 all (6.1-6.4)
 - second edition: Chap 7

2

News

- Welcome back!
- Midterms returned last time
 - get yours after class if you didn't already
- Department news

3

Events this week	Events next week
Resume & Cover Letter Drop-In Session	Townhall Meeting – Combined
Date: Wed., Mar 3	Majors/Honours, BA, B.Comm in CS
Time: 12 – 3 pm (20 mins. sessions)	Date: Thurs., Mar 11
Location: Rm 255, ICICS/CS	Time: 12:30 – 2 pm
	Location: DMP 310
	Lunch will be provided!
Find a Job Fast! Info Session	CS Distinguished Lecture Series –
Date: Thurs., Mar 4	Featuring David Parkes
Time: 12:30 – 1:45 pm	Title: Incentive Mechanism
Location: DMP 201	Engineering in the Internet Age
Registration: Email dianejohn@cs.ubc.ca	Date: Thurs., Mar 11
Townhall Meeting – 1st Year CS Students	Time: 3:30 – 4:50 pm
Date: Thurs., Mar 4	Location: DMP 110
Time: 12:30 – 2 pm	
Location: DMP 310	CSSS Movie Night –
Lunch will be provided!	“Zombieland” & “Iron Man”
Faculty Talk – Son Vuong	Date: Thurs., Mar 11
Title: Mobile Learning via LIVES	Time: 6 – 10 pm
Date: Thurs., Mar 4	Location: DMP 310
Time: 12:30 – 1:45 pm	Free pop & popcorn!
Location: DMP 201	

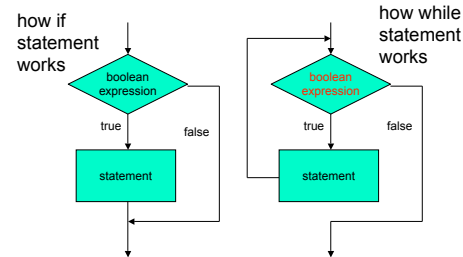
4

Recap: while Statement

while (boolean expression)
body

- Simplest form of loop in Java
- Body of loop can be
 - single statement
 - whole block of many statements in curly braces
- Control flow
 - body executed if expression is true
 - then boolean expression evaluated again
 - if expression still true, body executed again
 - repetition continues until expression false
 - then processing continues with next statement after loop⁵

Recap: If Versus while Statements



- How can loop boolean change from false to true?

6

Using while Statements

```
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 statement

7

Using while Statements

```
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");
    }
}
```

- boolean expression

8

Using while Statements

```
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 statement body

9

Using while Statements

```
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");
    }
}
```

- statement after while
 - control flow resumes here when boolean is false

10

Using while Statements

```
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");
    }
}
```

- trace what happens when execute

11

Using while Statements

```
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");
    }
}
```

limit 3

12

Using while Statements

```
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");
    }
}
```

limit 3 counter 1

13

Using while Statements

```
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");
    }
}
```

limit 3 counter 1 Is counter <= limit? yes

14

Using while Statements

```
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");
    }
}
```

limit 3 counter 1 Is counter <= limit? yes

"The square of 1 is 1" printed on monitor

15

Using while Statements

```
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");
    }
}
```

limit 3 counter 2

16

Using while Statements

```
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");
    }
}
```

limit **3** counter **2** Is counter <= limit? yes

17

Using while Statements

```
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");
    }
}
```

limit **3** counter **2** Is counter <= limit? yes

"The square of 2 is 4" printed on monitor

18

Using while Statements

```
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");
    }
}
```

limit **3** counter **3**

19

Using while Statements

```
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");
    }
}
```

limit **3** counter **3** Is counter <= limit? yes

20

Using while Statements

```
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");
    }
}
```

limit **3** counter **3** Is counter <= limit? yes

"The square of 3 is 9" printed on monitor

21

Using while Statements

```
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");
    }
}
```

limit **3** counter **4**

22

Using while Statements

```
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");
    }
}
```

limit **3** counter **4** Is counter <= limit? NO!

23

Using while Statements

```
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");
    }
}
```

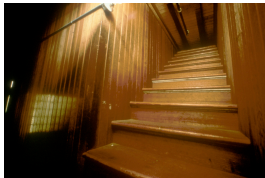
limit **3** counter **4** Is counter <= limit? NO!

"End of demonstration" printed on monitor

24

Climbing Stairs Again

- Am I at the top of the stairs?
- No.
- Climb up one step.
- Am I at the top of the stairs?
- No.
- Climb up one step.
- Am I at the top of the stairs?
- No.
- Climb up one step.
- Am I at the top of the stairs?
- No.
- Climb up one step.
- ...and so on...



25

Climbing Stairs Again

```
while (I'm not at the top of the stairs)
{
    Climb up one step
}
```

- Climbing stairs is a while loop!



26

Using while Statements

```
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");
    }
}
```

- change termination condition

27

Using while Statements

```
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");
    }
}
```

- change termination condition
 - body of loop never executed

28

Using while Statements

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

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

- change termination condition
 - always true

29

Infinite Loops

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

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

- if termination condition always true, loop never ends
 - infinite loop goes forever

30

Infinite Loops

```
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");
    }
}
```

- good termination condition
- but process never gets closer to condition

31

Infinite Loops

```
public class WhileDemo
{
    public static void main (String[] args)
    {
        int limit = 9;
        int counter = 0;

        while (counter != limit)
        {
            System.out.println("The square of " + counter +
                " is " + (counter * counter));
            counter = counter + 2;
        }
        System.out.println("End of demonstration");
    }
}
```

- process gets closer to termination condition
- but never satisfies condition, keeps going past it

32

Another while Example

```
public class PrintFactorials
{
    public static void main (String[] args)
    {
        int limit = 10;
        int counter = 1;
        int product = 1;

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

- accumulate product

33

Fun With Loops

```
public class BeerSong
{
    public static void main (String[] args)
    {
        int beerNum = 99;
        String word = "bottles";
        while (beerNum > 0)
        {
            if (beerNum == 1)
            {
                word = "bottle";
            }

            System.out.println(beerNum + " " + word + " of beer on the wall.");
            System.out.println(beerNum + " " + word + " of beer.");
            System.out.println("If one of the bottles");
            System.out.println("should happen to fall...");
            beerNum = beerNum - 1;
        }

        if (beerNum < 1)
        {
            System.out.println("No more bottles of beer on the wall.");
        }
    }
}
```

34

Fun With Loops

```
import java.util.Scanner;

public class BeerSong2
{
    public static void main (String[] args)
    {
        int beerNum = 99;
        String word = "bottles";

        boolean keepingGoing = true;
        String answer;
        Scanner in = new Scanner(System.in);

        while ((beerNum > 0) && keepingGoing)
        {
            if (beerNum == 1)
            {
                word = "bottle";
            }

            System.out.println(beerNum + " " + word + " of beer on the wall.");
            System.out.println(beerNum + " " + word + " of beer.");
            System.out.println("If one of the bottles");
            System.out.println("should happen to fall...");
            beerNum = beerNum - 1;
        }
    }
}
```

35

Fun With Loops

```
System.out.println("Continue? (y/n): ");
answer = in.nextLine();
if (answer.equals("n"))
{
    keepingGoing = false;
}

if (beerNum < 1)
{
    System.out.println("No more bottles of beer on the wall.");
}
}
```

36

Other Loop Statements

```
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");
    }
}
```

- Equivalent to...

37

Other Loop Statements

```
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");
    }
}
```

- ...this loop using for statement

38

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");
    }
}
```

- for statement

39

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

40

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");
    }
}
```

- Initialization: first part
 - executed only one time, at beginning

41

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");
    }
}
```

- boolean expression: second part
 - evaluated just before loop body, like in while

42

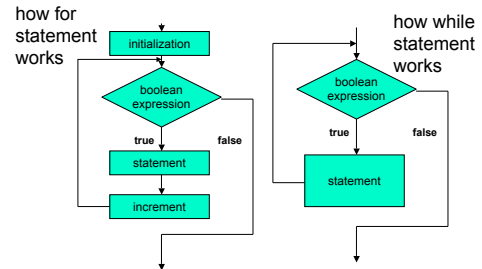
For Statement

```
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");
    }
}
```

- Increment: third part
 - executed at end of loop body
- Despite name, arbitrary calculation allowed
 - could decrement, for example!

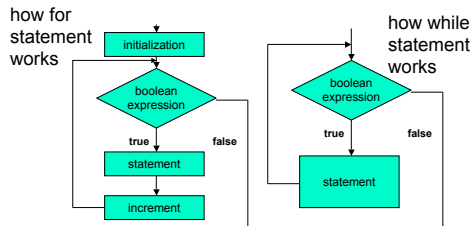
43

For Versus while Statement



44

For Versus while Statement



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

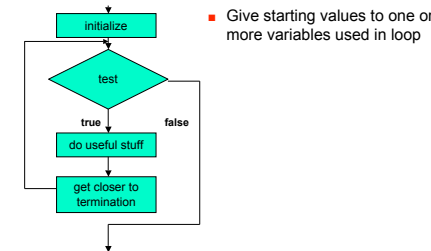
45

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

46

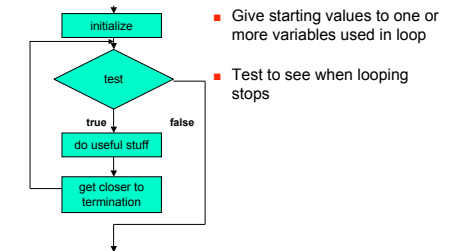
Four Things Needed In Any Loop



how loops work in general

47

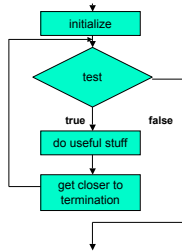
Four Things Needed In Any Loop



how loops work in general

48

Four Things Needed In Any Loop

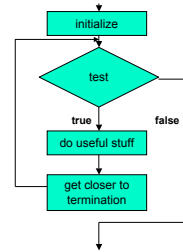


- Give starting values to one or more variables used in loop
- Test to see when looping stops
- One or more useful operations here

how loops work in general

49

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

50

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

51

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

52

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

53

Do Statement

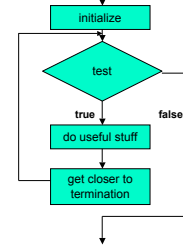
```

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

54

Four Things Needed In Any Loop

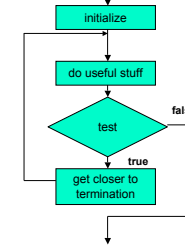


- Give starting values to one or more variables used in loop
- Test to see when looping stops
- One or more useful operations here
- Change something to move process closer termination

how loops work in general

55

Do Statement



- Body always executed at least once

order of four things can change, but need them all

56