# Loop - break and continue

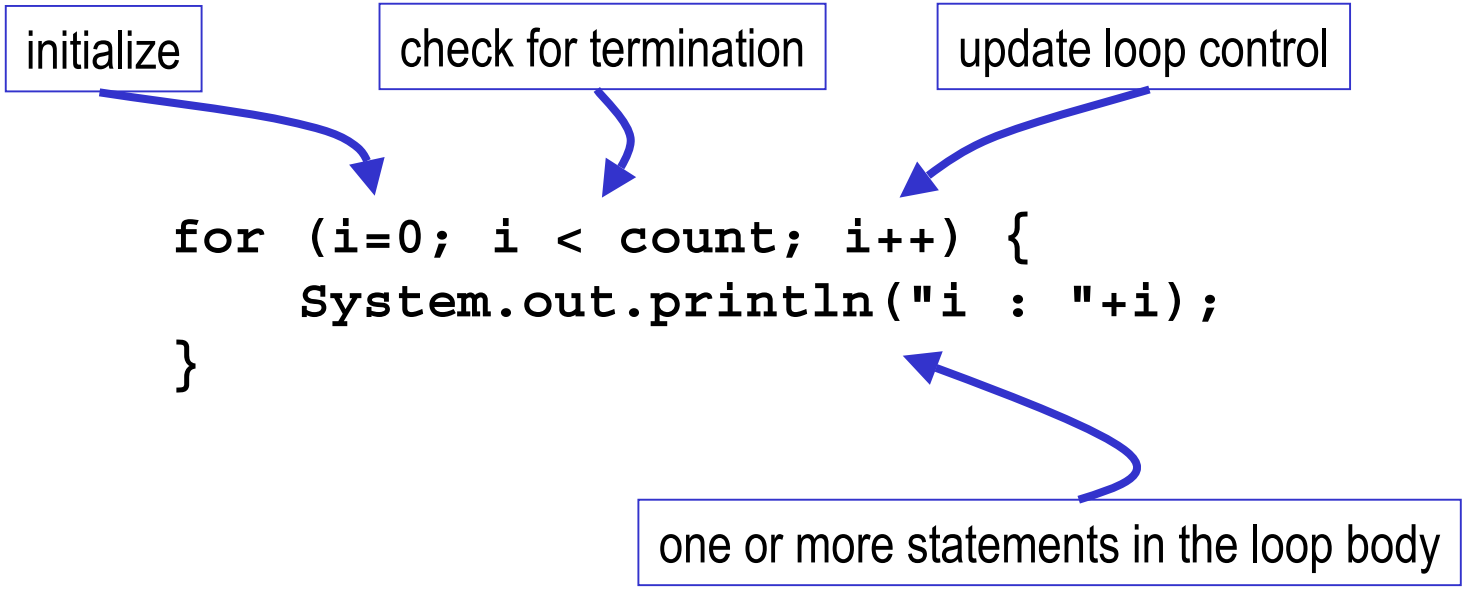
CSE 142, Summer 2002

Computer Programming 1

http://www.cs.washington.edu/education/courses/142/02su/

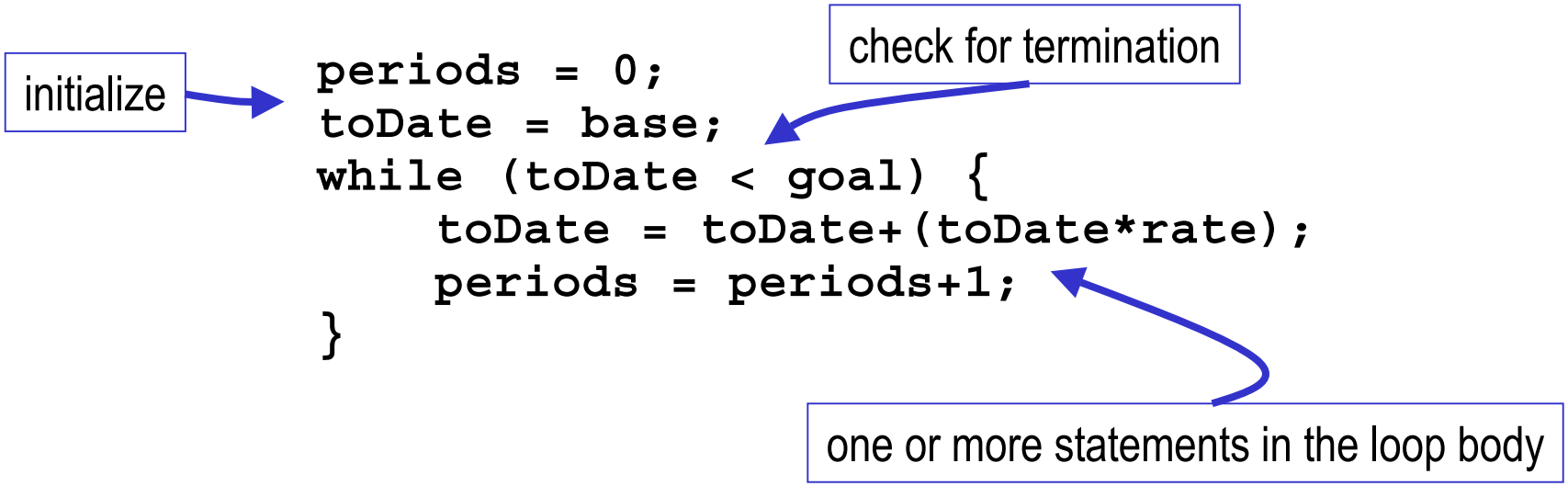
# The `for` loop

- A counting loop is usually implemented with `for`
  - » The `for` statement is defined in section 14.13 of the Java Language Specification



# The **`while`** loop

- condition loop is usually implemented with **`while`**
  - » The **`while`** statement is defined in section 14.11 of the Java Language Specification




Note: reaching a limit by counting is satisfying a condition.
**`for`** loops can be rewritten as **`while`** loops, and vice versa

# Early terminaton of the loop statement

- A loop is often used to look at all the elements of a list one after another
  - » all the Animals in a PetSet
  - » all the Shapes in a Car
- Sometimes we want to
  - » exit the loop statement early if we find some particular element or condition while we are looping
  - » ie, get out of the loop statement (for, while) entirely

# break - jump to loop exit

```
public void snack() {
  for (int i=0; i<theBunch.size(); i++) {
    if (remainingFood <= 0) {
      System.out.println("No food left, so no more snacks.");
      break;
    }
    Animal pet = (Animal)theBunch.get(i);
    double s = Math.min(remainingFood,pet.getMealSize());
    pet.eat(s);
    remainingFood -= s;
  }
  // the break statement takes us here, out of the loop entirely
}
```

# Early cycling of the loop

- Sometimes we want to
  - » Stop processing the item we are looking at right now and go on to the next one
- The loop statement (for, while) is still the controlling structure, but we just want to go to the next iteration of the loop

# continue - jump to loop end

```
public void dine() {
  for (int i=0; i<theBunch.size(); i++) {
    Animal pet = (Animal)theBunch.get(i);
    double s = 2*pet.getMealSize();
    if (remainingFood < s) {
      System.out.println("Not enough food for "+pet+
        "'s dinner, so we'll skip to next animal.");
      continue;
    }
    pet.eat(s);
    remainingFood -= s;
    // continue takes us here, the end of this loop
  }
}
```

# Example

```
C:\home\finson\cse142\dev\ex142\lect13>java -classpath . PetList
Created Fred
Created Smoky
Created Libby
Fred : Woof! Woof!
Smoky : Meow! Meow!
Libby : Chirp! Trill!
Fred : Ate 1.0 pounds of 1.0 provided.  Weight is now 20.5 pounds.
No food left, so no more snacks.
Not enough food for Fred's dinner, so we'll skip to next animal.
Not enough food for Smoky's dinner, so we'll skip to next animal.
Not enough food for Libby's dinner, so we'll skip to next animal.
Fred : Snrf ... mutter ... snrf ...
Smoky : snrfle ... snore ... meow ... snrfle
Libby : <not a peep>
```