Lecture 3

Conditions using if Statements

Boolean

- Booleans are variables that represent either true or false.
- You can think of them like a light switch (They are either on or off)
- true: Represents success or a valid condition
- false: Represents false or an invalid condition

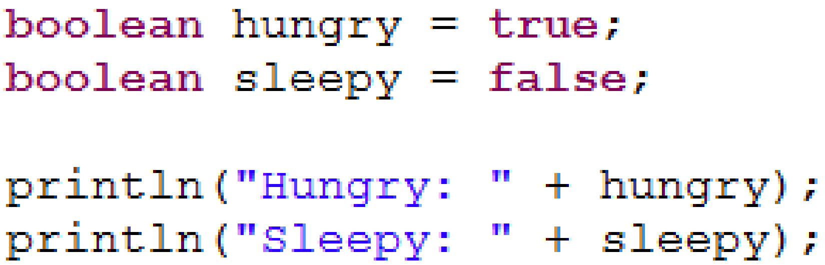
Value

- In Java and many programming languages
  - 0 means false
  - 1 means true

- This is NOT the case in C#

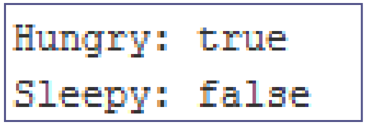
Creating Booleans

- You can declare booleans as follows:



```
boolean hungry = true;
boolean sleepy = false;

println("Hungry: " + hungry);
println("Sleepy: " + sleepy);
```



```
Hungry: true
Sleepy: false
```

- Notice how true and false are not in quotes because they are keywords in Java

if statement

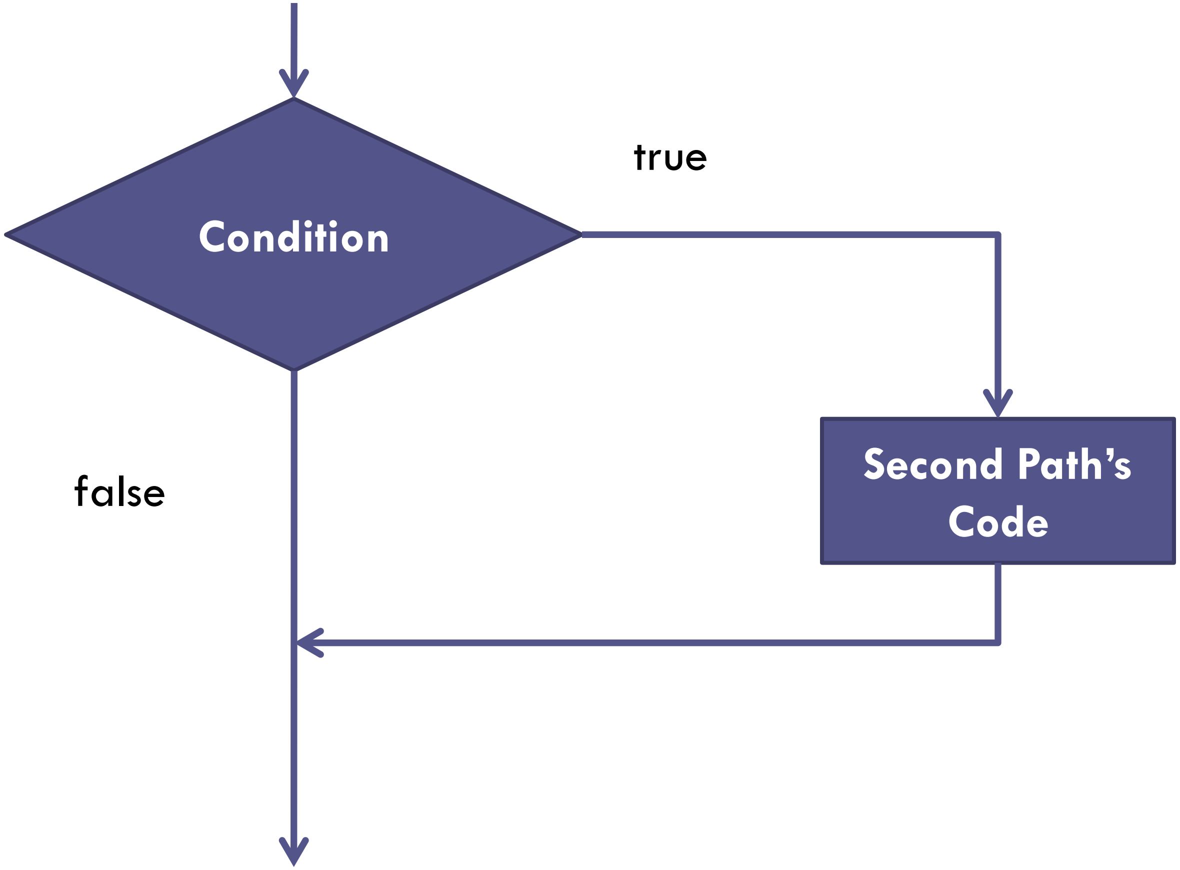
Like a fork in the road

Making Decisions: if Statement

- The if statement is like a fork in the road.

- if checks a condition:
  - If it is true it takes one path
  - if it is false it takes another

if Statement Flow



if Statement Syntax

```
if(condition)
{
  conditional code
}
```

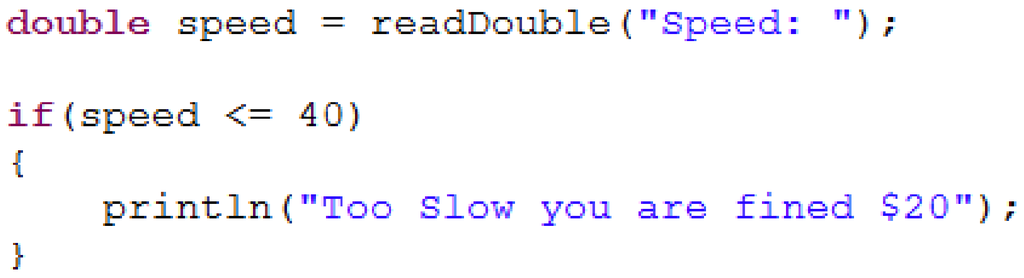
Note: You should indent the code inside the curly brackets

Comparison Operators

Operator	Title	Use	Description
==	Equal	a == b	true if a and b have the same value
!=	Not Equal	a != b	true if a and b don’t have the same value
>	Greater Than	a > b	true if a is greater than b
<	Less Than	a < b	true if a is less than b
>=	Greater Than or Equal to	a >= b	true if a is greater than or equal to b
<=	Less Than or Equal	a <= b	true if a is less than or equal to b

if Statement Example

- □ Here if the user enters a speed less than or equal to 40 they will get the angry message otherwise they'll see nothing.

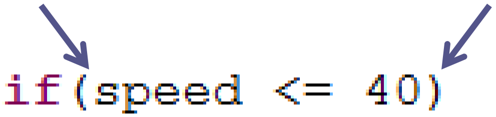


```
double speed = readDouble("Speed: ");

if(speed <= 40)
{
    println("Too Slow you are fined $20");
}
```

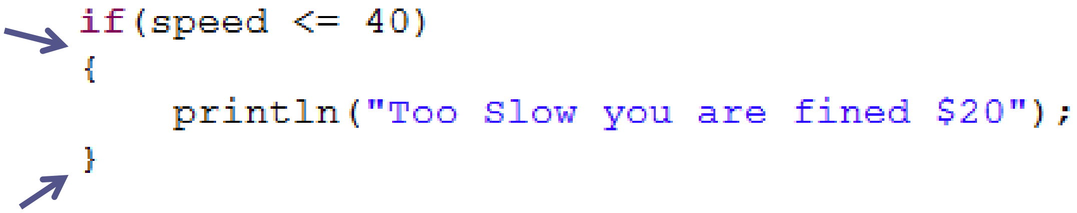
Bracket Breakdown

- Around the condition we have parethensis



```
if(speed <= 40)
```

- The curly braces scope the code that will be executed if the condition is true.
- You put multiple lines in the curly braces



```
if(speed <= 40)
{
    println("Too Slow you are fined $20");
}
```

What if they are going above 40?

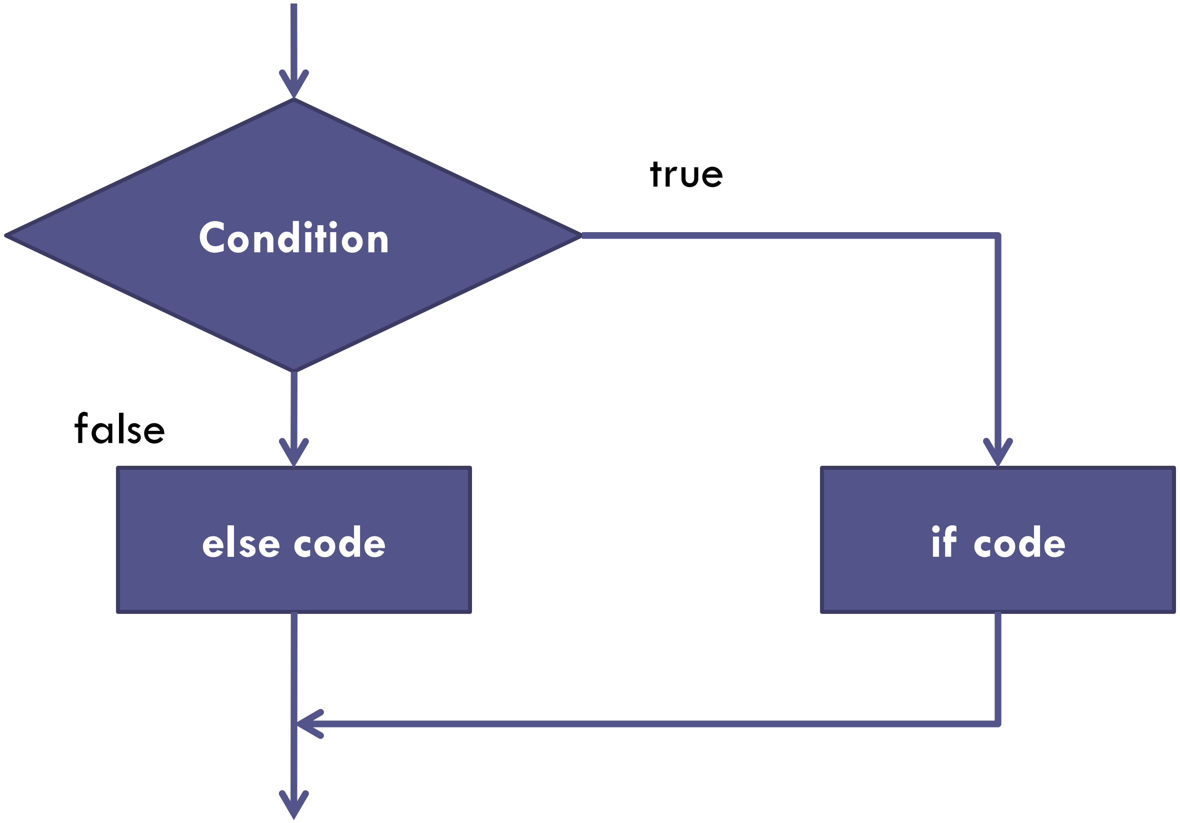
- But what if we want to give a message to all of the people going above 40 and say “You are not driving too slow”
- We need to extend our structure to account for that

if…else Statement

- else is the coding term for otherwise.

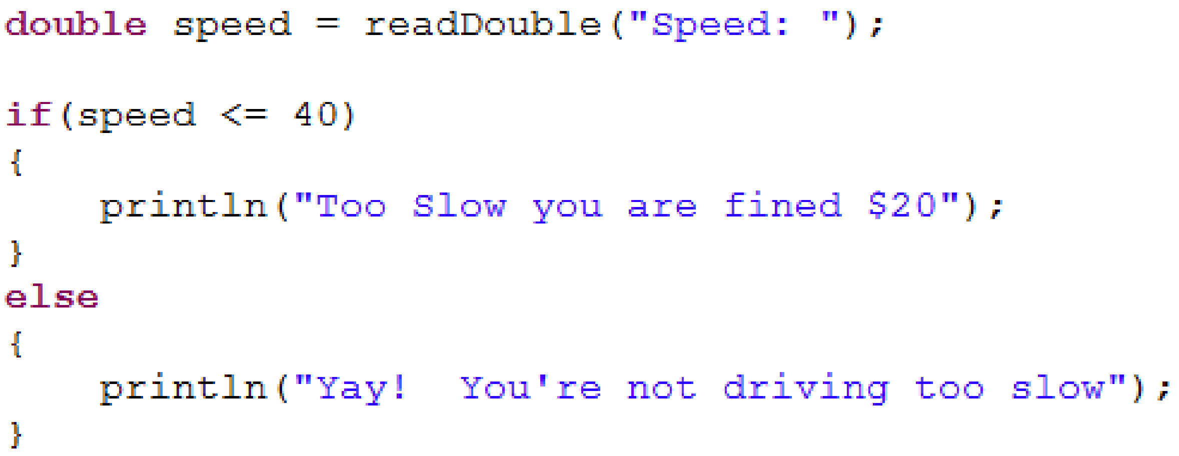
```
if(condition)
{
  code we run if condition is true
}
else
{
  code we run if condition is false
}
```

if…else Statement Flow



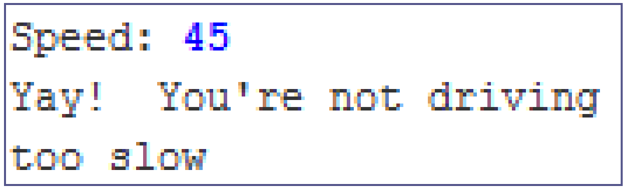
if…else Example

- Now there’s a different message depending on what speed the user goes at



```
double speed = readDouble("Speed: ");

if(speed <= 40)
{
    println("Too Slow you are fined $20");
}
else
{
    println("Yay!  You're not driving too slow");
}
```



```
Speed: 45
Yay!  You're not driving
too slow
```

What if they have more conditions

- Maybe we want to give a variety of messages based on the users speed.
  - less than 40 -> Too slow you have a ticket
  - between 40 and 60 -> Perfect Speed
  - between 60 and 70 -> Too fast you have a warning
  - greater than 70 -> fine time

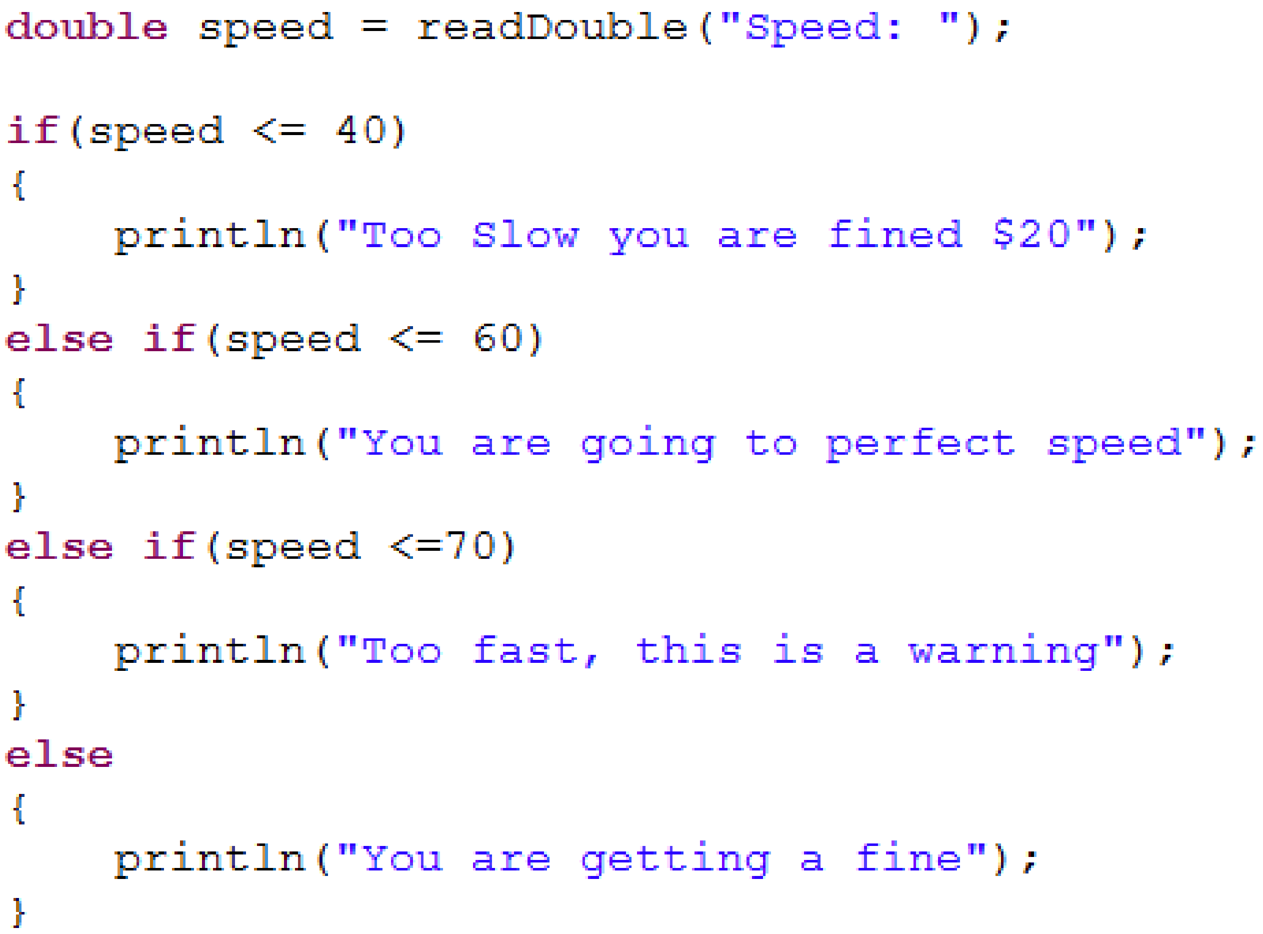
- Now we need a way to make multiple conditions

if… else if … else statement

else if, is the coding term for otherwise check this

```
if(condition1)
{
     code we run if condition is true
}
else if(condition 2)
{
     code if condition 2 is true
}
else
{
     code if both conditiosn are false
}
```

if … else if … else example

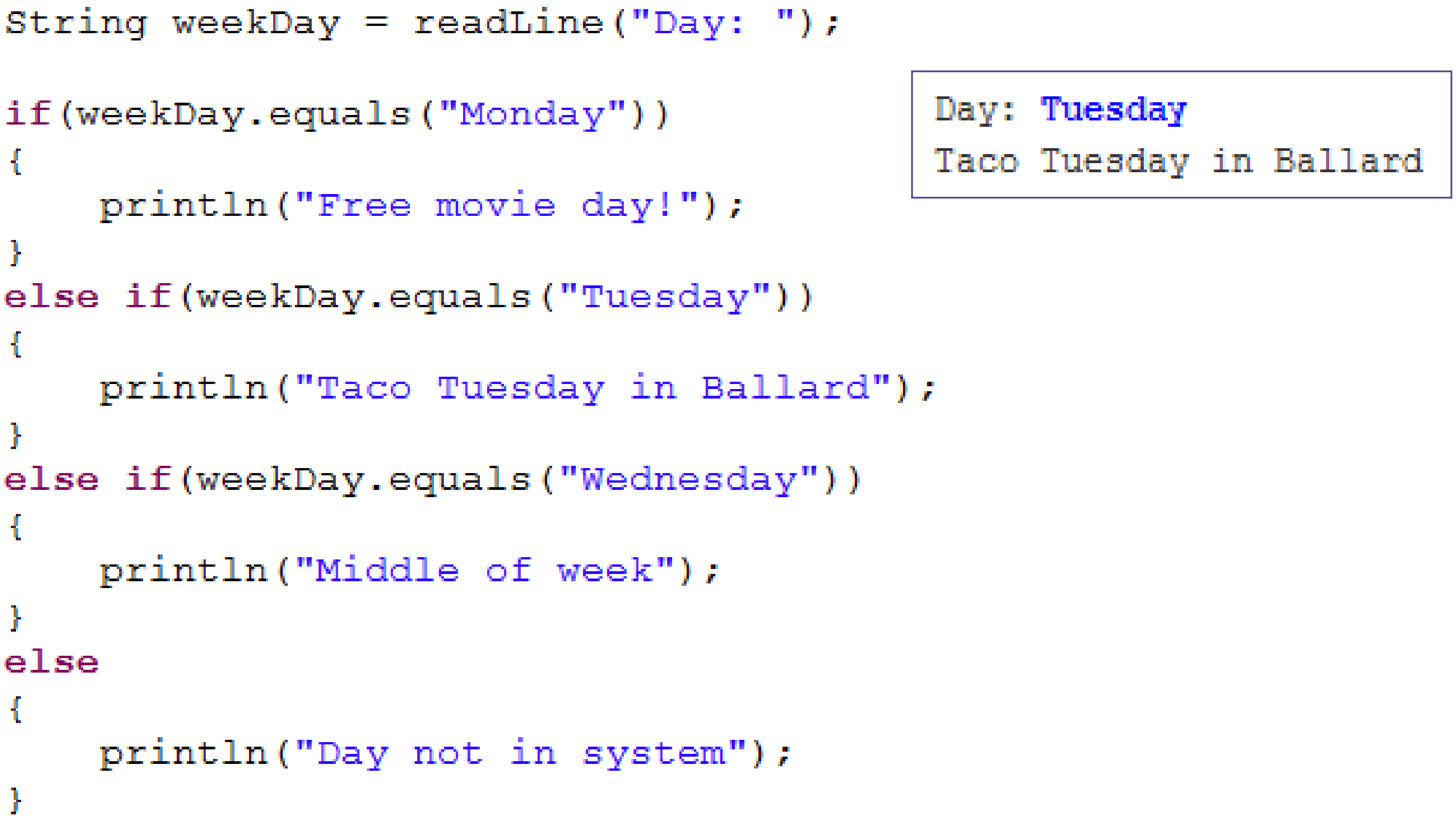


```
double speed = readDouble("Speed: ");

if(speed <= 40)
{
    println("Too Slow you are fined $20");
}
else if(speed <= 60)
{
    println("You are going to perfect speed");
}
else if(speed <=70)
{
    println("Too fast, this is a warning");
}
else
{
    println("You are getting a fine");
}
```

if example comparing Strings

- To compare Strings we use the .equals



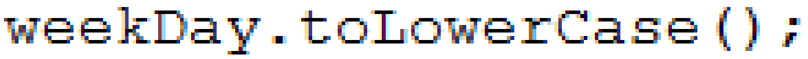
```
String weekDay = readLine("Day: ");

if(weekDay.equals("Monday"))
{
     println("Free movie day!");
}
else if(weekDay.equals("Tuesday"))
{
     println("Taco Tuesday in Ballard");
}
else if(weekDay.equals("Wednesday"))
{
     println("Middle of week");
}
else
{
     println("Day not in system");
}
```

```
Day: Tuesday
Taco Tuesday in Ballard
```

Case Sensitive

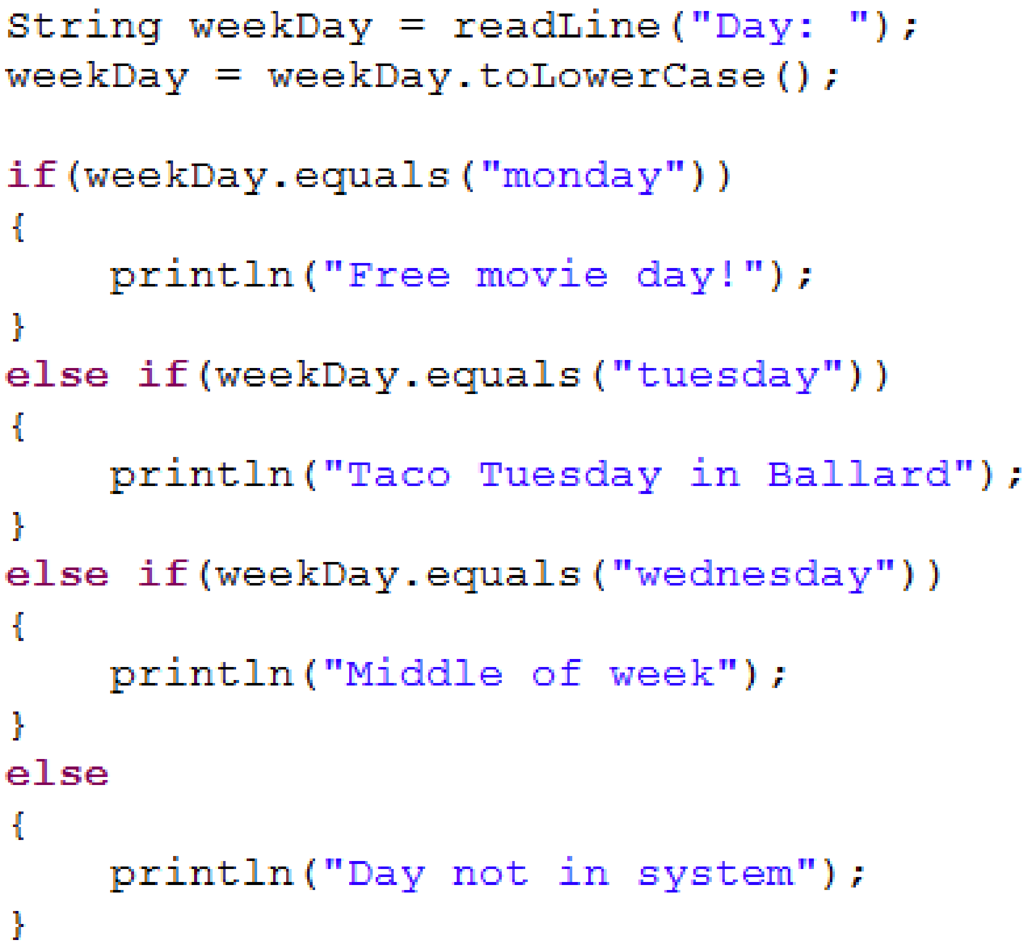
- □ Notice that if you put ‘tuesday’ or “TUESDAY” it won’t work because it’s case sensitive.
- □ To avoid case sensitivity you will convert the input to lower case using:



```
weekDay.toLowerCase();
```

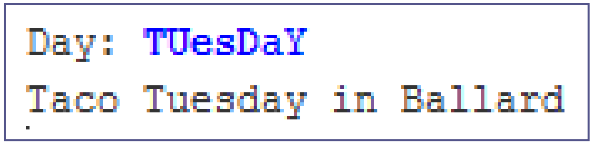
- □ Then compare it against all lower case data

Case InSensitive Program



```
String weekDay = readLine("Day: ");
weekDay = weekDay.toLowerCase();

if(weekDay.equals("monday"))
{
    println("Free movie day!");
}
else if(weekDay.equals("tuesday"))
{
    println("Taco Tuesday in Ballard");
}
else if(weekDay.equals("wednesday"))
{
    println("Middle of week");
}
else
{
    println("Day not in system");
}
```



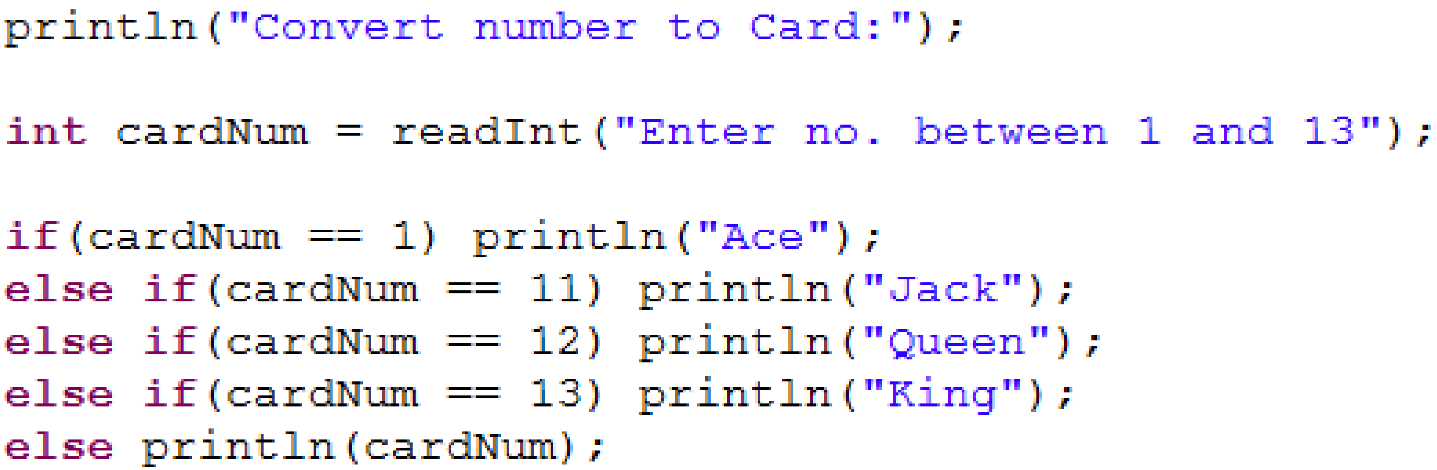
```
Day: TUesDaY
Taco Tuesday in Ballard
```

switch…case

Switch statements are useful when you are continuously comparing one variable to a variety of other ones.

When to use switch…case

- □ Here we are continuously comparing cardNum to other numbers this is the perfect scenario for a switch…case.



```
println("Convert number to Card:");

int cardNum = readInt("Enter no. between 1 and 13");

if(cardNum == 1) println("Ace");
else if(cardNum == 11) println("Jack");
else if(cardNum == 12) println("Queen");
else if(cardNum == 13) println("King");
else println(cardNum);
```

switch … case Syntax

The syntax for the switch statement is:

```
switch (expression)
{
  case label:
      statement(s);
      break;
  case label:
      statement(s);
      break;
  ...
  default:
      statement(s);
}
```

Case Line

- We test the expression against the label.
- The lines that follow the case lines are the ones that execute if the case is met.

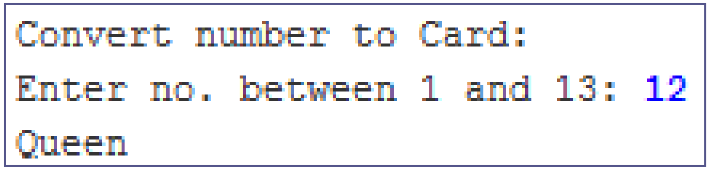
```
switch(expression) {
  case label:
```

- The break is required to get out of the structure when you've finished executing the codition

Switch … Case Example

```
int cardNum = readInt("Enter no. between 1 and 13: ");

switch(cardNum)
{
    case 1:
        println("Ace");
        break;
    case 11:
        println("Jack");
        break;
    case 12:
        println("Queen");
        break;
    case 13:
        println("King");
        break;
    default:
        println(cardNum);
}
```



```
Convert number to Card:
Enter no. between 1 and 13: 12
Queen
```

default is the else case

Logical Operators

Operators that produce boolean results

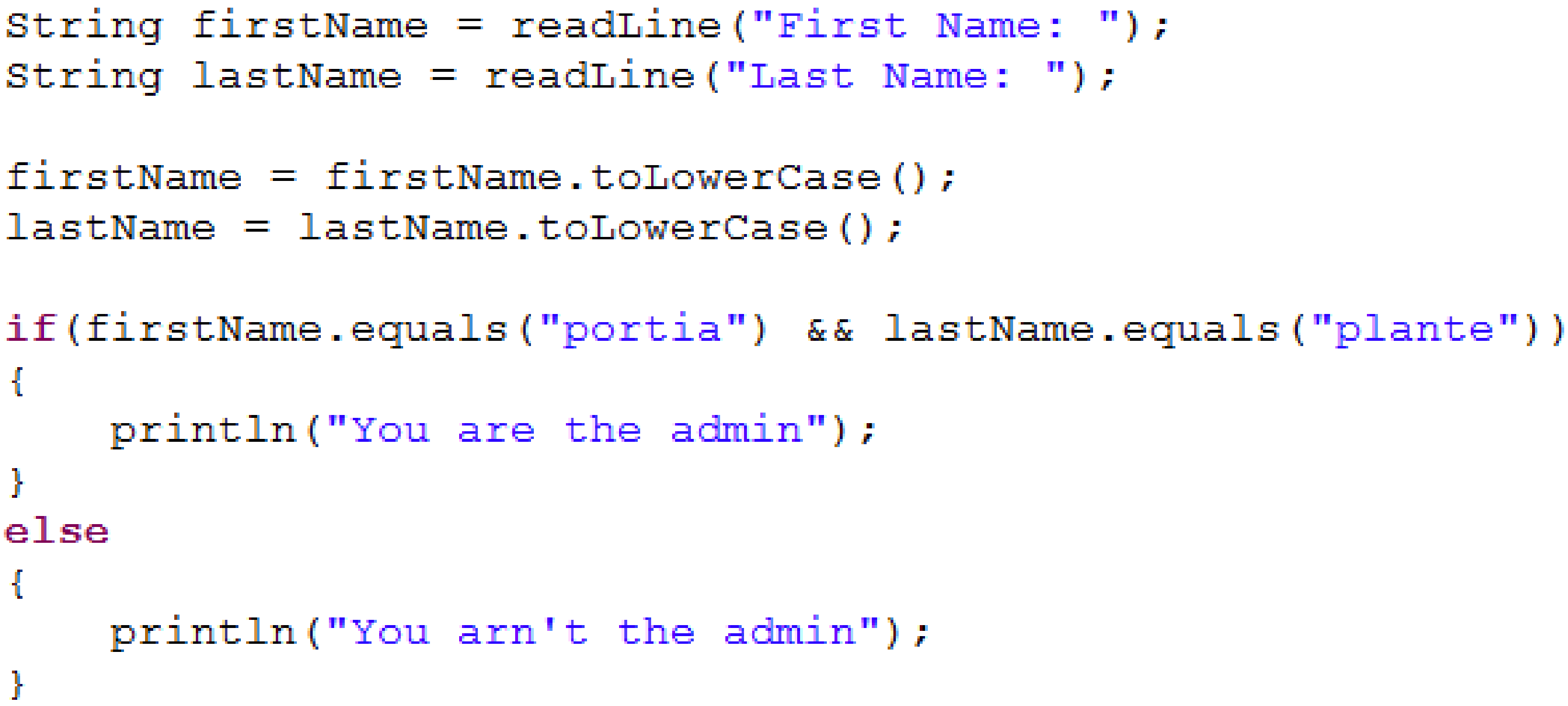
Logical Operators

Operator	Title	Example
&&	AND	true if both expressions are true
||	Short circuit OR	true if either expression is true
!	NOT	true if the expression is false

When to use AND

- I want to specify if someone is an admin of my program.
- They are an admin if:
  - There firstName is Portia, and
  - if their lastName is Plante

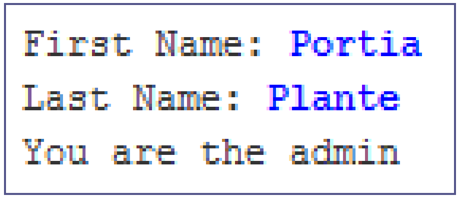
AND Example



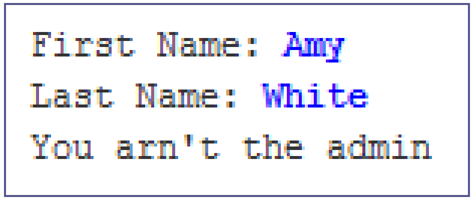
```
String firstName = readLine("First Name: ");
String lastName = readLine("Last Name: ");

firstName = firstName.toLowerCase();
lastName = lastName.toLowerCase();

if(firstName.equals("portia") && lastName.equals("plante"))
{
    println("You are the admin");
}
else
{
    println("You arn't the admin");
}
```



```
First Name: Portia
Last Name: Plante
You are the admin
```

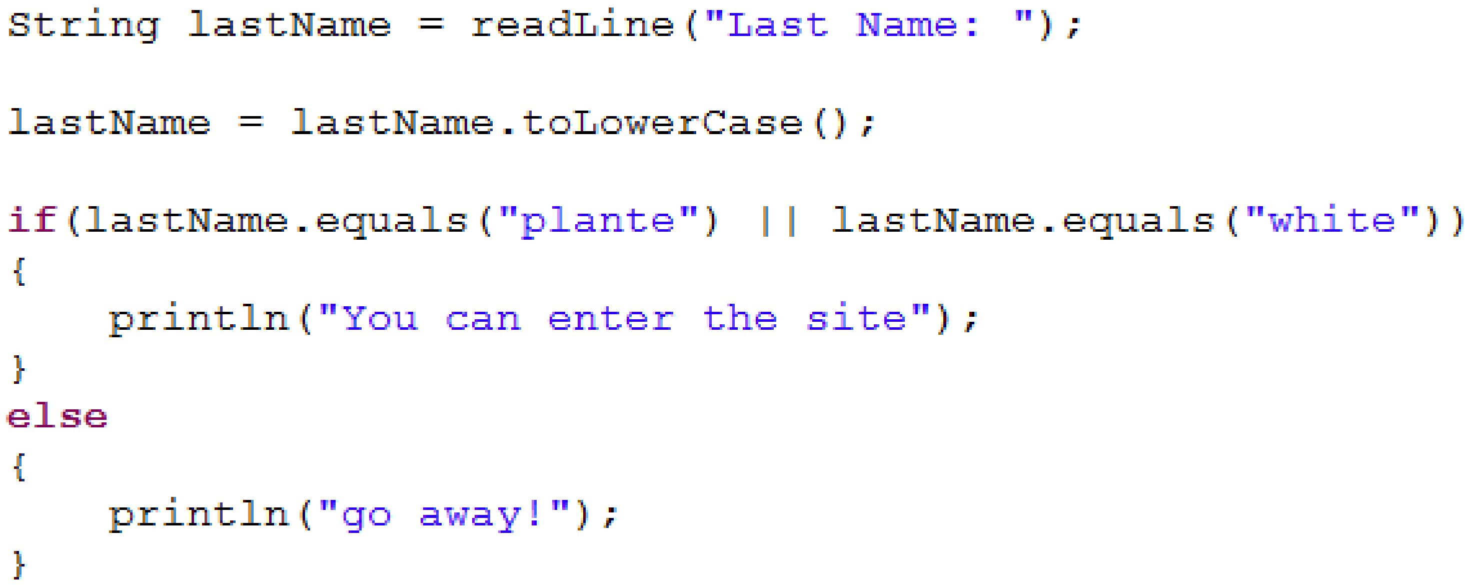


```
First Name: Amy
Last Name: White
You arn't the admin
```

When to use OR

- If I want to say multiple families can sign into my program.
- If your lastName is Plante or your lastName is White you can enter the site

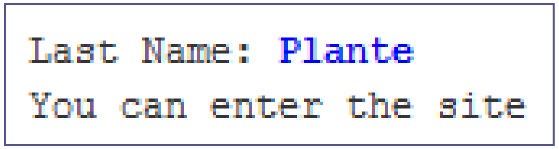
OR Example



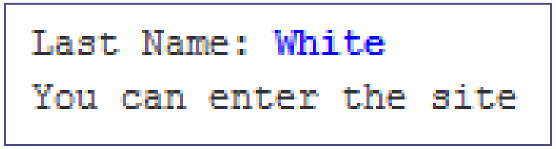
```
String lastName = readLine("Last Name: ");

lastName = lastName.toLowerCase();

if(lastName.equals("plante") || lastName.equals("white"))
{
    println("You can enter the site");
}
else
{
    println("go away!");
}
```

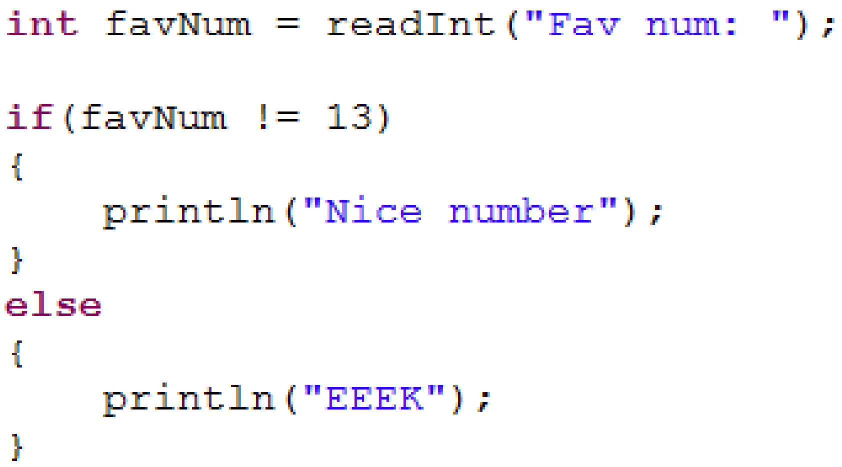


```
Last Name: Plante
You can enter the site
```



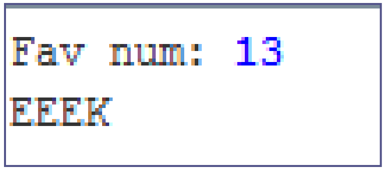
```
Last Name: White
You can enter the site
```

! (Not) Example

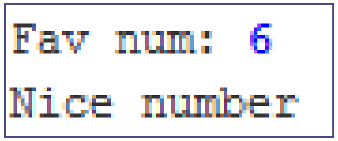


```
int favNum = readInt("Fav num: ");

if(favNum != 13)
{
    println("Nice number");
}
else
{
    println("EEEK");
}
```



```
Fav num: 13
EEEK
```



```
Fav num: 6
Nice number
```