

# More on IF

GEEN163 Introduction to  
Computer Programming

*“In any moment of decision, the best thing you can do is the right thing, the next best thing is the wrong thing, and the worst thing you can do is nothing.”*

Theodore Roosevelt

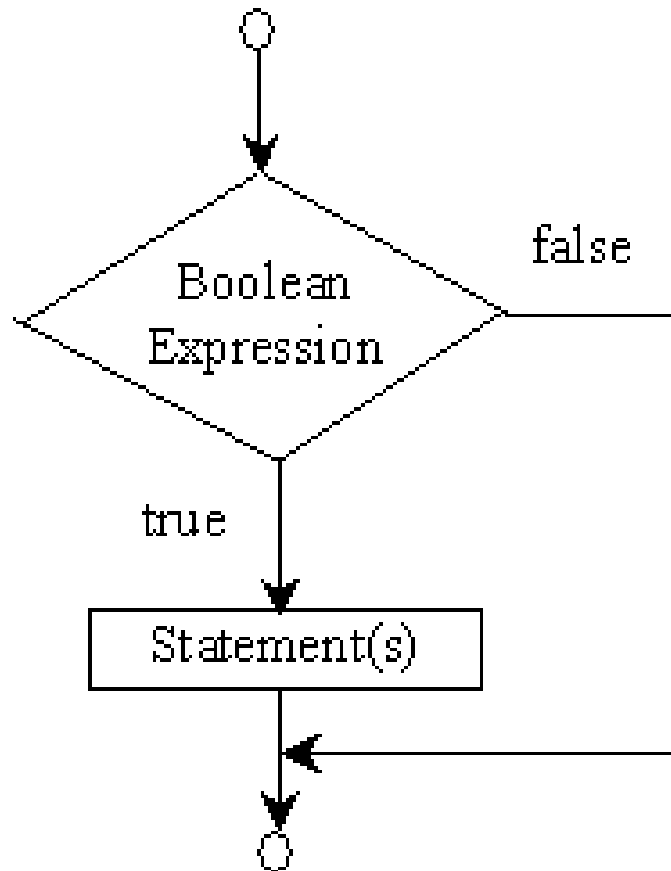
# TuringsCraft Assignment

- Answer any of the 62 questions in sections 5.1 – 5.5 of the TuringsCraft tutoring system
- You will earn **3** points for each correct answer up to a maximum of 100
- Due midnight on **Tuesday**, September 27, 2016

# Programming Assignment

- A programming assignment has been posted on Blackboard for this week
- You must write a program with a GUI that also makes a simple decision
- Due midnight on **Saturday**, September 24, 2016

# if Logic



# if Syntax

**if** ( *true or false decision* )  
*next statement; or {next block}*

- The program will execute the next statement if and only if the decision is true
- The next statement can be a single Java statement or a block
- Whitespace is optional

# Comparison Operators

*Operator Name*

< less than

<= less than or equal to

> greater than

>= greater than or equal to

== equal to

!= not equal to

# Watch Out for Double Equals

if (no = good) /\* Incorrect \*/

if (does == work) /\* Correct \*/



# Indenting

- Although the Java compiler does not care, it is traditional to indent the statements that are executed only when the if is true
- Indenting is required for all assignments

```
if ( dog > cat ) {  
    cow = 5;  
    bull = 37;  
}
```

# Blocks



- In Java, a block is a bunch of code surrounded by { curly brackets }
- Almost anyplace you might put a single statement, you can put a block of statements

# Brackets Recommended

- Your instructor recommends you always use {brackets} around statements following an **if**
- This makes it clear what will be skipped
- Without brackets, a careless update later can make your program not work

```
if (dog == cat )
```

```
    cat = 47;
```

```
dog += cat;
```

```
if (dog == cat )
```

```
    update = 1;
```

```
    cat = 47;
```

```
dog += cat;
```

# Compound Logical Statements

- You can combine relational expressions with logical or Boolean operators
- Expressions can be combined with AND, OR, XOR and NOT

# Boolean Operators

*Operator Name*

! not

&& and

|| or

^ exclusive or



George Boole  
19<sup>th</sup> century British mathematician  
inventor of Boolean logic

# Boolean Logic

- A logical **AND** is true if both sides are true

```
if (cow > bull && bull == 17)
```

- A logical **OR** is true if at least one side is true

```
if (cow > bull || bull == 17)
```

# What is displayed?

```
int bull = 3, cow = 5, cat = 7;  
int dog = 9;  
if((bull != cow) && (cat < cow)) {  
    dog = 10;  
}  
System.out.println(dog);
```

- A. 3
- B. 5
- C. 7
- D. 9
- E. 10

# Now what is displayed?

```
int bull = 3, cow = 5, cat = 7;  
int dog = 9;  
if((bull != cow) || (cat < cow)) {  
    dog = 10;  
}  
System.out.println(dog);
```

- A. 3
- B. 5
- C. 7
- D. 9
- E. 10



# Caution

- Adding a semicolon at the end of an `if` clause is a common mistake

```
if (radius >= 0) ; ← Wrong
{
    area = radius*radius*Math.PI;
    System.out.println(area);
}
```

- This mistake is hard to find, because it is not a compilation error or a runtime error, it is a logic error
- This error often occurs when you use the next-line indenting style

## Try It

- Write an **if** statement that sets `mongoose` to 5 if `weasel` is bigger than `mink` or if `mink` is negative

```
int mongoose = 0, weasel, mink;
```

```
// assume values are given to weasel and mink
```

# Possible Solution

- Write an **if** statement that sets mongoose to 5 if weasel is bigger than mink or if mink is negative

```
int mongoose = 0, weasel, mink;
```

```
// assume values are given to weasel and mink
```

```
if ( weasel > mink || mink < 0) {
```

```
    mongoose = 5;
```

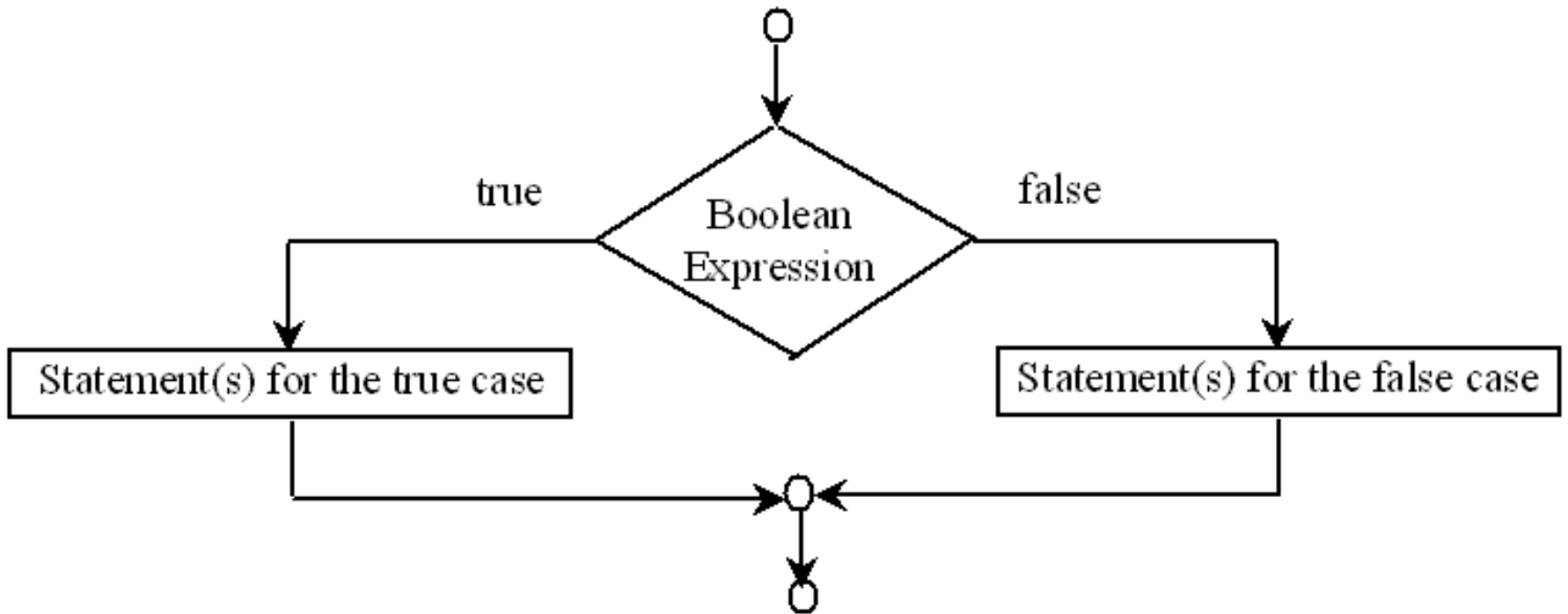
```
}
```

# else

- An if statement can have an else clause that is executed only when the if condition is false

```
if ( cat > 4 ) {  
    System.out.println("cat is big");  
} else {  
    System.out.println("cat is small");  
}
```

# if else Logic

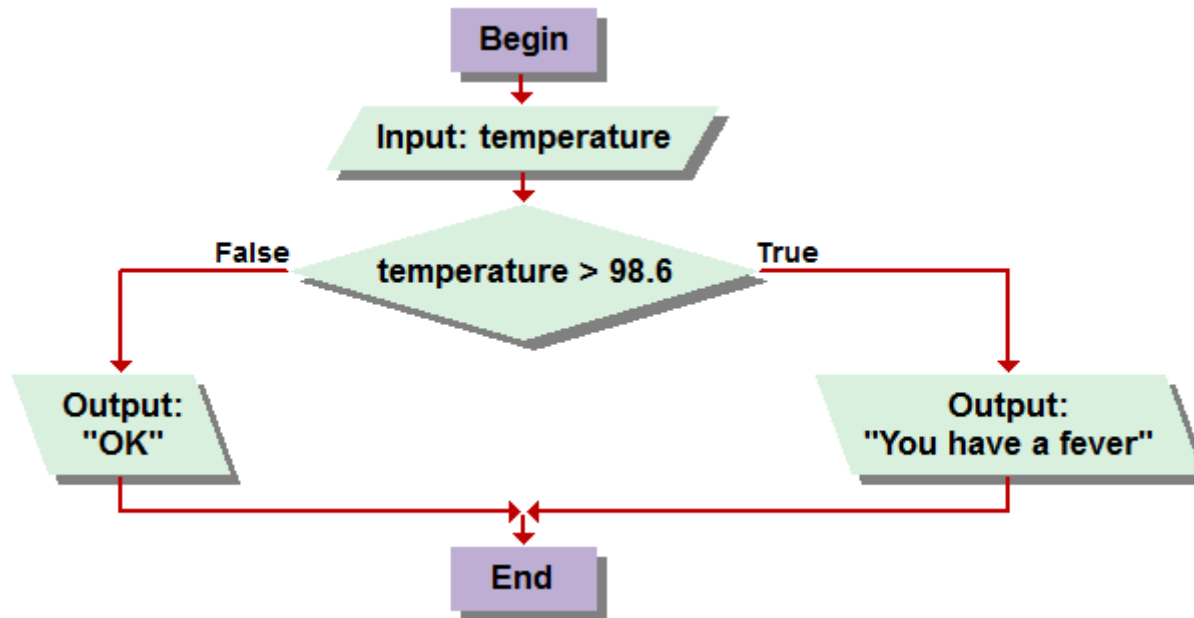


# One or the Other

- With an if – else statement, either the if part or the else part are executed, but never both

```
if ( logical expression ) {  
    Executed only if true  
} else {  
    Executed only if false  
}
```

# Flowchart to tell if a fever



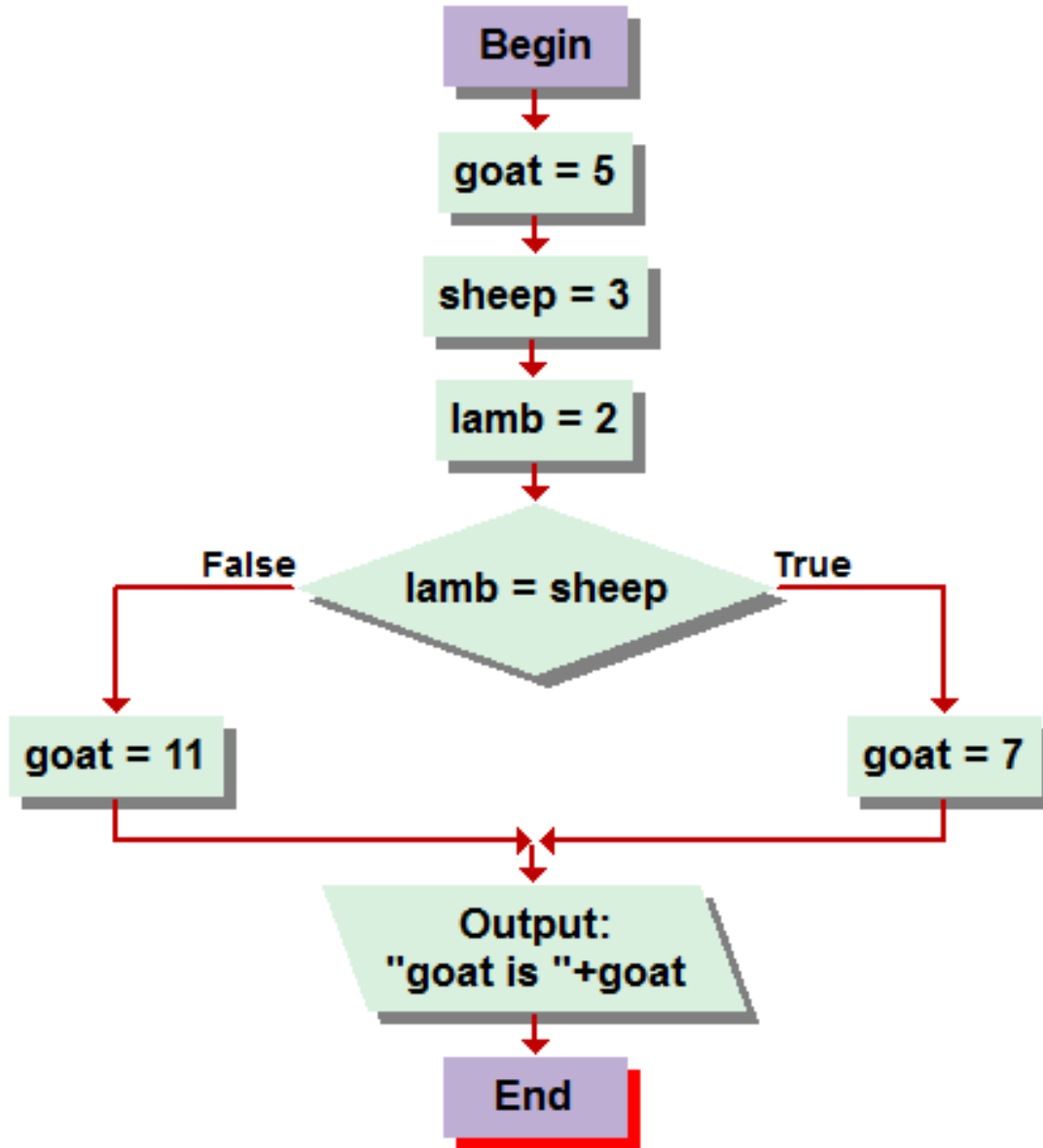
# Java Version

```
public class Fever {
    public static void main(String[] unused) {
        double temperature;
        java.util.Scanner keyboard = new
            java.util.Scanner(System.in);
        System.out.println("Enter temperature");
        temperature = keyboard.nextDouble();

        if (temperature > 98.6) {
            System.out.println("You have a fever");
        } else {
            System.out.println("OK");
        }
    }
}
```



# What is output by this program?



- A. goat is 2
- B. goat is 3
- C. goat is 5
- D. goat is 7
- E. goat is 11

# A Recommendation

- Always put the clause following an **if** and **else** in curly brackets
- This avoids nested if errors
- This avoids errors when inserting an extra line in an if clause
- The brackets should line up with proper indenting

# Try It

- Print an error message if the variable loan or the variable years are less than zero

```
double loan, years;
```

```
// input the value of loan and year
```

```
if( ? ) {
```

```
    System.out.println("Bad stuff");
```

```
}
```

# Possible Solution

- Print an error message if the variable loan or the variable years are less than zero

```
double loan, years;  
// input the value of loan and year  
  
if( loan < 0.0 || years < 0.0 ) {  
    System.out.println("Bad stuff");  
}
```

# Incorrect Solution

- Print an error message if the variable loan or the variable years are less than zero

```
double loan, years;
```

```
// input the value of loan and year
```

```
if( loan || years < 0.0 ) {    // Does NOT work
```

```
    System.out.println("Bad stuff");
```

```
}
```

# What is the final value of cow?

```
int dog=2, cat=3, cow=5;  
if (dog > cat) cow=7; else cow=9;
```

- A. 5
- B. 7
- C. 9
- D. Compiler Error

# Comparing Strings

- The comparison operators (such as `>`, `<`, `==`) do **not** work with Strings. They only work on primitive data types (`int`, `float`, `double` and `char`)
- You need to use the `equals` or `compareTo` methods to compare Strings

# equals Method

- **equals** is a boolean method of String that is true if the parameter has exactly the same characters

```
String cat = "frog", dog = "toad", cow = "frog";
```

```
if ( cat.equals(dog) ) is false
```

```
if ( cat.equals(cow) ) is true
```



# Comparison Examples

```
String dog = "dog", bull = "bull", bulldog="bulldog", cat;
```

```
cat = bull + dog;           // cat = "bulldog"
```

```
if (cat == bulldog) not true
```

```
if (cat.equals(bulldog)) true
```

```
cat = bulldog;
```

```
if (cat == bulldog) true
```

# compareTo method

```
String dog, cat;  
dog.compareTo( cat )
```

- if dog is alphabetically before cat, then the compareTo method returns a positive number
- if dog is alphabetically after cat, then the compareTo method returns a negative number
- if dog is the same as cat, then the compareTo method returns zero

# compareTo example

```
String dog = "bull dog";
```

```
String cat = "puma";
```

```
if (dog.compareTo(cat) > 0) { // if positive  
    System.out.println("dog comes before cat");  
}
```

- The result of compareTo is zero if the Strings are equal, positive if dog < cat and negative if dog > cat

# Upper and Lower Case

- When comparing Strings, upper and lower case characters are different

“Dog” is not equal to “dog”

- The **equalsIgnoreCase( anotherString )** method is true if both strings have the same letters regardless of case
- **compareToIgnoreCase( anotherString )** works like `compareTo` but ignores case

# Complete the Program

```
public class Tax{
    public static void main(String[] unused) {
        double rate;
        String name;
        java.util.Scanner keyboard = new
            java.util.Scanner(System.in) ;
        System.out.println("Enter name");
        name = keyboard.next ();

        /* Set rate to 0.01 if the name is Fred, otherwise set
           rate to 25.0 */
    }
}
```

# Possible Solution

```
public class Tax{
    public static void main(String[] unused) {
        double rate;
        String name;
        java.util.Scanner keyboard = new
            java.util.Scanner(System.in);
        System.out.println("Enter name");
        name = keyboard.next ();

        if( name.equalsIgnoreCase("Fred")) {
            rate = 0.01;
        } else {
            rate = 25.0;
        }
    }
}
```

# TuringsCraft Assignment

- Answer any of the 62 questions in sections 5.1 – 5.5 of the TuringsCraft tutoring system
- You will earn **3** points for each correct answer up to a maximum of 100
- Due midnight on **Tuesday**, September 27, 2016

# Programming Assignment

- A programming assignment has been posted on Blackboard for this week
- You must write a program with a GUI that also makes a simple decision
- Due midnight on **Saturday**, September 24, 2016