DECISIONS – THE “IF” STATEMENT

CS302 – Introduction to Programming
University of Wisconsin – Madison
Lecture 5

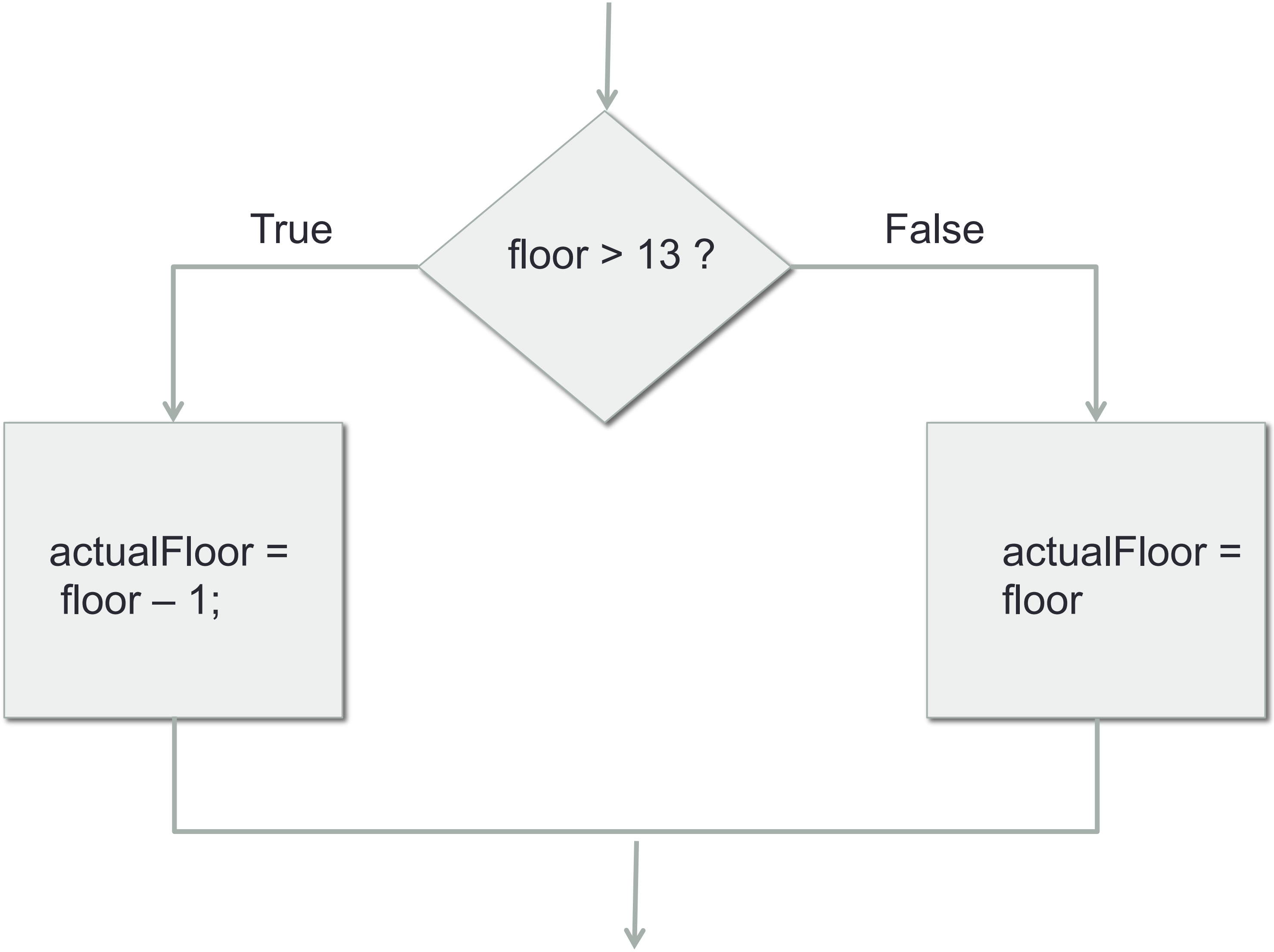
By Matthew Bernstein – matthewb@cs.wisc.edu

Decisions – The “if” Statement

- The if statement is used to implement a a decision when a condition is fulfilled.
- If the condition within an “if” statement evaluates to true, execute a set of statements
- Example:

```
int actualFloor ;

if (floor > 13)
{
        actualFloor = floor – 1;
}
else
{
        actualFloor = floor;
}
```

Flowchart Representation



True

floor > 13 ?

False

actualFloor = floor – 1;

actualFloor = floor

Notes on Flowcharts

- Conditionals (i.e. the “if” statement) are written in a diamond shape
- Blocks of statements are written in a rectangular shape
- Arrows show flow of the program

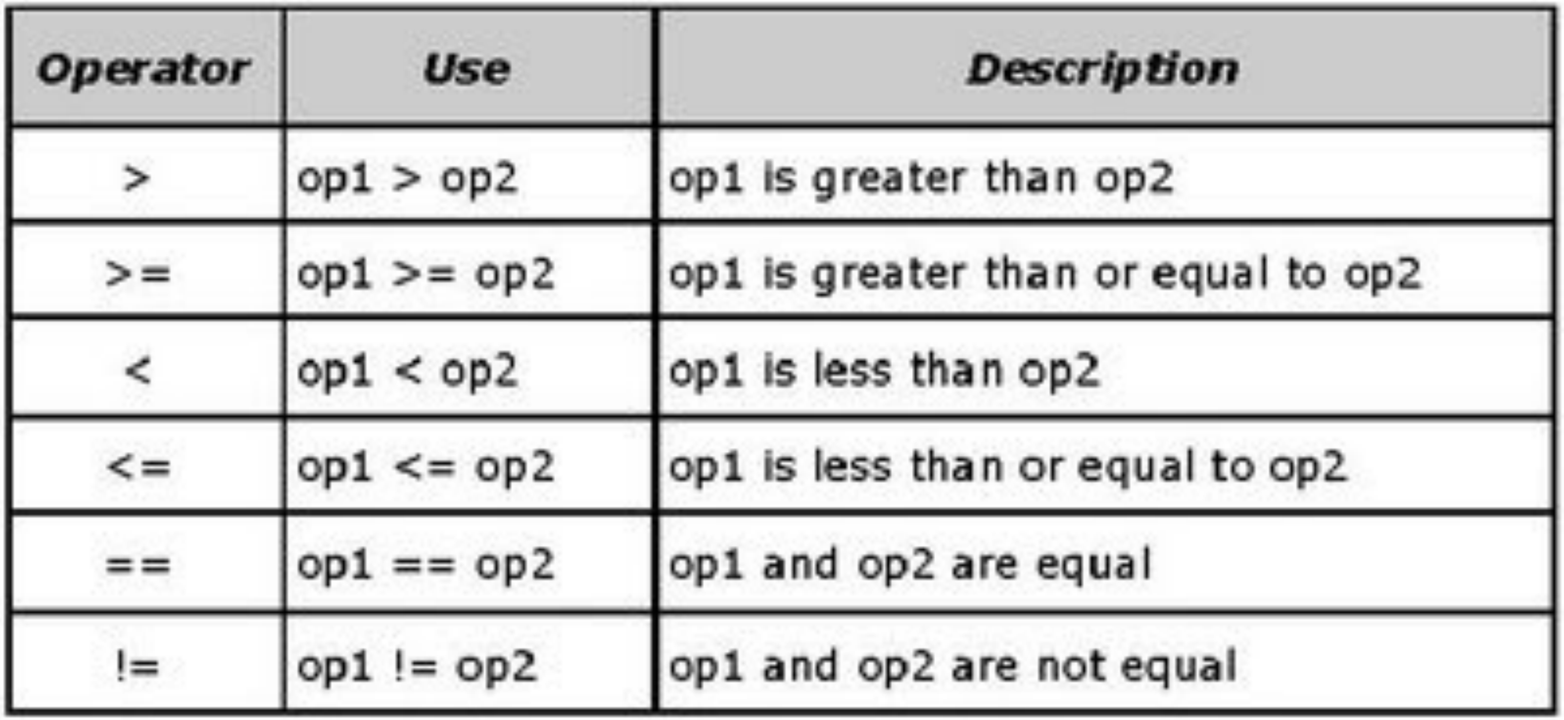
More Details About “if” Statements

- You do not need an “else” statement following an “if”.
- If the condition in the “if” statement does not evaluate to true, the block of code is simply not executed
- What will the count variable hold after the following code executes?

```
int count = 2;
int numPounds = 20;

count++;

if (numPounds > 30)
{
        count++;
}

count++;
```

Relational Operators

- **Relational operators** compare numbers.
- You can use any of the following relational operators in an if statement:



Operator	*Use*	*Description*
>	op1 > op2	op1 is greater than op2
>=	op1 >= op2	op1 is greater than or equal to op2
<	op1 < op2	op1 is less than op2
<=	op1 <= op2	op1 is less than or equal to op2
==	op1 == op2	op1 and op2 are equal
!=	op1 != op2	op1 and op2 are not equal

Example

```
int x = 40;

if (x >= 40)
{
        if (x > 30)
        {
                x = 10;
        }
        else
        {
                x  = 0;
        }
}
```

What is wrong with this code?

```
if (scoreA = scoreB)
{
        System.out.println(“Tie”);
}
```

The Assignment Operator vs. The “Equal to” Relational Operator

- The “=“ symbol is an assignment operator. It does not compare two numbers for equality.
  - It is only used for assigning a value to a variable
  - Example:

```
int x = 4;
```

- In order to compare two numbers for equality, you must use the “Equal to” relational operator: “==“

Comparing Strings

- You must use Strings’ equals() method for checking equality between two strings
- Example:

```
String myName = “Matt”;
String yourName = “Matt”;
if (myName.equals( yourName ) )
{
        System.out.println(“Same name.”);
}
else
{
        System.out.println(“Different names.”);
}
```

Comparing Strings (continued)

- DO NOT USE THE “==“ OPERATOR WHEN COMPARING STRINGS!
- The “==“ operator does not test whether two strings have the same sequence of characters, but rather, it tests whether the two strings are stored in the same memory address.
- You will never need to know whether two strings occupy the same location in memory

Programming Exercise

- Add to the temperature converter program from the previous exercise such that the program first prompts the user for the temperature units the user will input and the units the program will output (either “F” or “C”). If the input units and output units are the same, then the program will simply output the same number.
- Example:

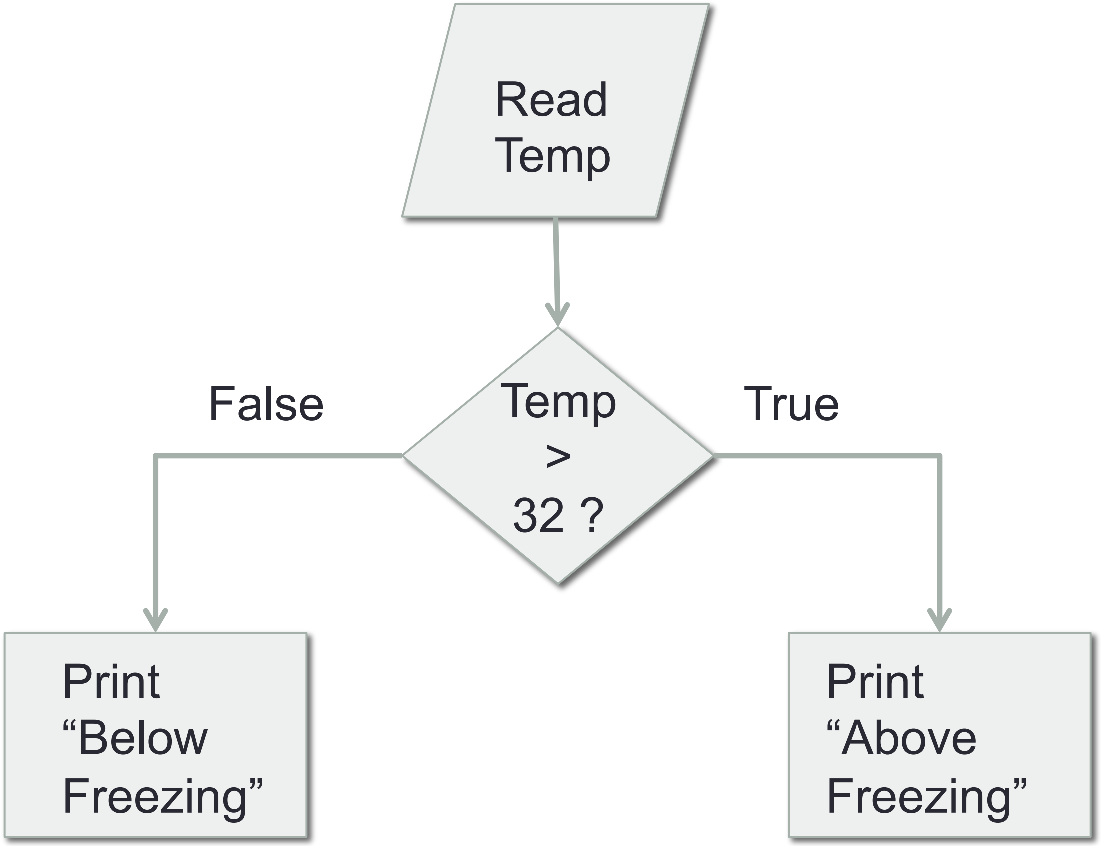
```
Input Units: F
Output Units: C
Input Temperature: 32.0
```

This is what the output should look like →

```
32.0 F = 0.0 C
```

More on Flowcharts

- The parallelogram represents user input. Example:



Solution to Programming Exercise

-- See Class Demo --

Multiple Alternatives

- Multiple alternatives are structures of “if”, “else if”, “else” clauses that allow you to program as many alternative as you need

Multiple Alternatives Example

```
if (altitude < 12) {
		System.out.println(“Troposphere”);
}
else if (altitude < 31) {
		Sysem.out.println(“Stratosphere”);
}
else if (altitude < 53) {
		System.out.println(“Mesosphere”);
}
else if (altitude < 375) {
		System.out.println(“Thermosphere”);
}
else if (altitude < 6200) {
		System.out.println(“Exosphere”);
}
else {
		System.out.println(“Outer Space”);
}
```

More details on multiple alternatives

- As soon as one of the “if” statements succeeds, its code is executed and no further “if” statements in this code structure are tested.
- If no “if” or “else if” conditional tests evaluate to true, the “else” statement’s code executes
- If no “else” statement exists, no blocks are executed and the program resumes execution after the multiple alternative structure

More details

- You CANNOT have any code between clauses of a multiple alternatives structure
- The following code is WRONG:

```
if (x > 10)
{
        System.out.println(“Greater than10”);
}
double y = 20.0;
else
{
        System.out.println(“Equal to or less than 10”);
}
```

Wrong Example (Missing “else”)

```
if (altitude < 12) {
        System.out.println(“Troposphere”);
}
if (altitude < 31) {
        Sysem.out.println(“Stratosphere”);
}
if (altitude < 53) {
        System.out.println(“Mesosphere”);
}
if (altitude < 375) {
        System.out.println(“Thermosphere”);
}
if (altitude < 6200) {
        System.out.println(“Exosphere”);
}
else {
        System.out.println(“Outer Space”);
}
```

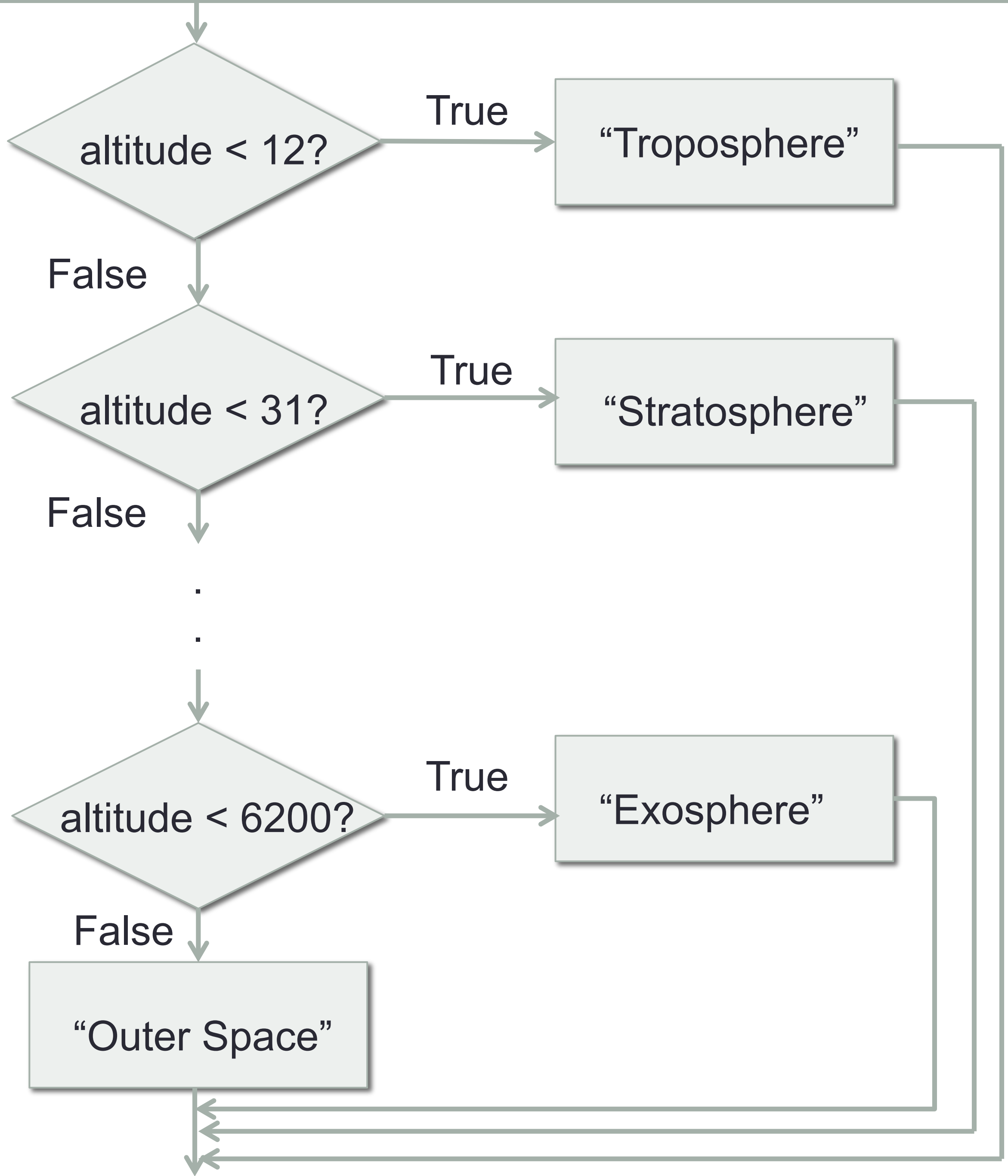
Another Wrong Example (Wrong order)

```
if (altitude < 6200) {
        System.out.println(“Exosphere”);
}
else if (altitude < 375) {
        System.out.println(“Thermosphere”);
}
else if (altitude < 53) {
        System.out.println(“Mesosphere”);
}
else if (altitude < 31) {
        Sysem.out.println(“Stratosphere”);
}
else if (altitude < 12) {
        System.out.println(“Troposphere”);
}
else {
        System.out.println(“Outer Space”);
}
```

In Class Exercise

- Draw a flowchart that represents the “if” clause structure that calculates the layer of the atmosphere
- Assume the altitude was input by a user (include this in the diagram)

Solution



altitude < 12?
True
"Troposphere"
False
altitude < 31?
True
"Stratosphere"
False
.
.
altitude < 6200?
True
"Exosphere"
False
"Outer Space"

The Switch Statement

```
int digit = 2;
switch(digit)
{
        case 1:
                System.out.println(“one”);
                break;
        case 2:
                System.out.println(“two”);
                break;
        default:
                System.out.println(“Neither one nor two”);
                break;
}
```

How it works

- It tests a value against several alternatives (the “cases”)
- If the value matches a case, the code following that case is executed until it reaches a break statement
- If no break statement is encountered, the program will keep executing statements
- If the value does not match any cases, then the code following the “default” case is executed

This is a shortcut for:

```
if (digit == 1)
{
        System.out.println(“one”);
}
else if (digit == 2)
{
        System.out.println(“two”);
}
else
{
        System.out.println(“Neither one nor two”);
}
```

Wrong Example (Missing Breaks)

```
int digit = 2;
switch (digit)
{
        case 1:
                System.out.println(“one”);

        case 2:
                System.out.println(“two”);
        default:
                System.out.println(“Neither one nor two”);
}
```

- Why do we need the break statements?
    - See Class Example

Important thing to note

- The values in the case clauses MUST be constant variables or **literals**

Why Switch Statements?

- The Java compiler optimizes the bytecode produced by switch statements (less bytecode is produced than “if” clauses)
- Some programmers feel that its logic is easier to follow
- You don’t necessarily *have* to use it, but you need to know what it is if you ever encounter it in other code

Random Number Generator

- How do we generate a random number in programming?
- Use a psuedorandom number generator
- These are implemented in **java.util.random**
- Example:

```
// Create a random number generator object
Random rand = new Random();

int randomInt = rand.nextInt() % 5; // Produces a random integer
```

Programming Exercise: Rock, Paper, Scissors

- Program one round of rock paper scissors such that the user enters a string (either “rock”, “paper”, “scissors”)
- After the user enters her shape, use a random number generator to have the computer throw a shape
- Output the results
- Example:

```
input shape (“rock”, “paper”, “scissors”): scissors
computer threw paper
You Win!
```

Cool CS Link

- A Look Into Google Glass:
- http://www.youtube.com/watch?v=v1uyQZNg2vE