Branching

Colorado State University

Announcements

BELIEVE YOU CAN AND YOU'RE HALFWAY THERE.
- THEODORE ROOSEVELT

THE ART OF LIVING

- Reminder – readings are due **before** lecture

  – You don't have to do all of it - challenge problems can be challenging…

  – You can return to them.

  – We start off each lecture with a quiz from your reading!

- Labs – you need to go to lab to have your participation points

- NEXT WEEK – EXAM 1 Week – you need to work to have your module open for the Exam
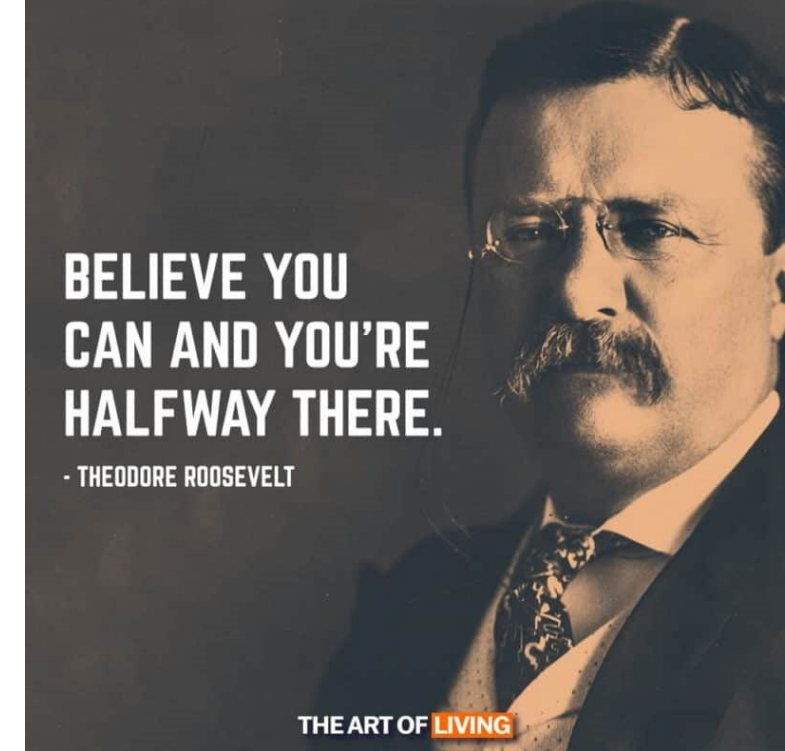
Recall Activity

Grab a paper, write your name as it is in Canvas, and your answers to the following questions. Turn this as your attendance for today's lecture.

List the conditional instructions that you can use in Java.

What is the difference between comparing primitive types, such as char, int, and float, with Strings for equality? Provide examples to support your answer.

Basic Conditionals

- Logic that evaluates as
    - Yes or No
    - **true** or **false**

- Essential in all programming languages
    - You do this all the time
    - 100 pennies greater than $1?

Conditional operators
- == Equals
- < Less than (is left less than right)
- > Greater than
- <= Less than OR equal
- >= Greater than OR equal
- != NOT equal (! is your NOT character)

Conditional Expressions

- Conditional Operators
  - Evaluated second to last (storing values or returns last)

- Always two sides / pairs
  - You can NOT have 10 < x < 20
  - You would need to check (10 < x) == (x < 20)

- Adding conditional operators
  - means it is **true** or **false**
  - boolean primitive can store it, and you can return boolean from a method

```java
public static void main(String[] args) {
    boolean myVal = 10 > 2;
    System.out.println(myVal); // prints true to the screen
}
```

```java
public static void main(String[] args) {
    boolean myVal = 10 <= (20 * 2) + 1 / 2 - 39;
    System.out.println(myVal); // prints false to the screen
}
```

```java
public static void main(String[] args) {
    System.out.println(myCoolMethod()); // prints true
}

public static boolean myCoolMethod() {
    return 10.5 < 20;
}
```

If Statements

- if / else statements

  – Only run instructions based on true or false for else

  – Essentially, choose to run certain lines of code or skip them!

```java
if(puppies > 100) System.out.println("Yay puppies"); // one line, allowed but not common
```

```java
if(puppies >= 100) { //block of code!
   int happy  = puppies + 1;
   System.out.println("So happy");
}
```

```java
if(puppies >= 100) {
    int happy  = puppies + 1;
    System.out.println("happy level: " + happy);
} else {
   System.out.println("Need more puppies");
}
```

If Statements – Coding Along Practice 1

- In your table, write a method that receives two integer values and return the minimum value between the two parameters.

- Write a call to the method you created from the following main method and print the minimum value returned.

```java
import java.util.Scanner;
public class Conditionals {
    public static void main(String args[]){
        Scanner in = new Scanner(System.in);
        System.out.print("Enter fist value: ");
        int num1 = in.nextInt();
        System.out.print("Enter second value: ");
        int num2 = in.nextInt();
        System.out.printf("First number: %d, Second Number: %d", num1, num2);

    }
}
```

How can we use this method
to calculate the minimum between
three values?
Write the necessary changes in your code.

If Statements – Coding Along Practice 2

- In your table, write a method that receives two String objects and prints those strings in alphabetical order.

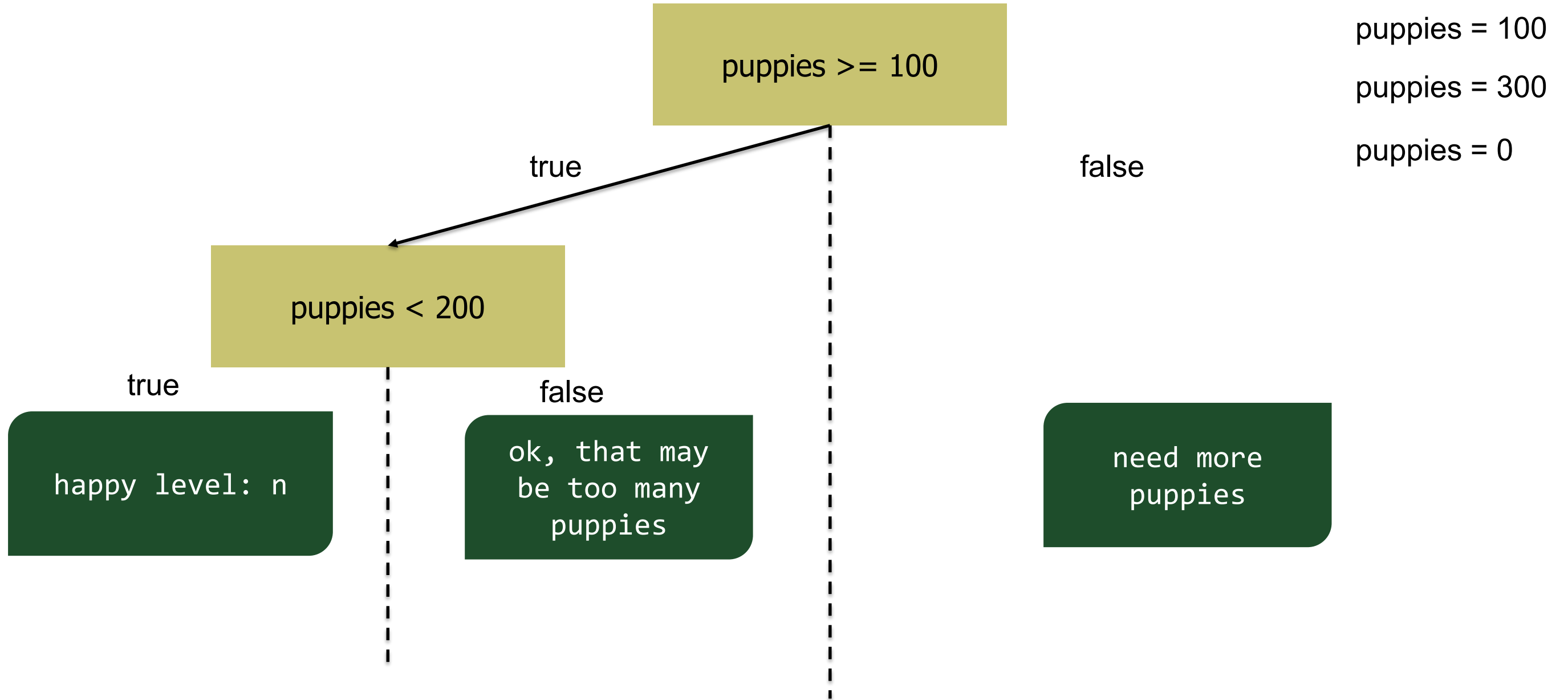- Write a call to the method you created from the following main method.

```java
import java.util.Scanner;
public class Conditionals {
    public static void main(String args[]){
        Scanner in = new Scanner(System.in);
        System.out.print("Enter first string: ");
        String str1 = in.nextLine();
        System.out.print("Enter second string: ");
        String str2 = in.nextLine();
        System.out.printf("First string entered: %s, Second string entered: %s", str1, str2);
    }
}
```

Nested If Statements

- You can nest as much as you want

```java
if (puppies >= 100) {
    if (puppies < 200) {
        int happy  = puppies + 1;
        System.out.println("happy level: " + happy);
    } else { // puppies is over 200!!
        System.out.println("ok, that may be too many puppies");
} else {
    System.out.println("Need more puppies");
}
```

How to track?

puppies >= 100

true

false

puppies = 100

puppies = 300

puppies = 0

puppies < 200

true

false

happy level: n

ok, that may be too many puppies

need more puppies

Colorado State University

In Class Activity: Group Practice 1

- Complete the following code with a conditional expression (available on canvas / in zyBooks)

```java
public static boolean quickPractice1(int value) {
    // just write the line of code that goes here!
}

public static void main(String[] args) {
    System.out.println(quickPractice1(100));  // prints true
    System.out.println(quickPractice1(101));  // prints true
    System.out.println(quickPractice1(99));   // prints false
}
```

In Class Activity: Group Practice 2

- Complete the following code with a conditional expression. In zyBooks / canvas – same one.

```java
public static String quickPractice2(int value) {
    // just write the lines of code that go here!
}

public static void main(String[] args) {
    System.out.println(quickPractice2(100));  // prints YES, at min
    System.out.println(quickPractice2(101));  // prints YES, over
    System.out.println(quickPractice2(99));   // prints NO, under
}
```

Group Activity – Draw the Tree

```java
public String branchingCheck(int valOne, int valTwo, int valThree) {

    double pi = 3.14;
    String hasPie = "I like pie";
    String morePie = "I need more pie";
    String actualPie = "π";

    if (valOne > valTwo) {
        if (valOne > pi) {
            return hasPie;
        }
    } else if (valTwo >= valThree) {
        return morePie;
    } else {
        if (valThree < 3.14) {
            return actualPie;
        }
        return "The value of pi is: " + pi;
    }
    return "no pie";
}
```

Answer the following

1. `branchingCheck(10, 20, 30)`

2. `branchingCheck(-1, -1, -1)`

3. `branchingCheck(3, 0, 2)`

4. `branchingCheck(-1, 0, 1)`

5. `branchingCheck(20, 19, 18)`

1. The value of pi is: 3.14

2. I need more

3. no pie

4. π

5. I like pie

Use the tree you drew !

Review

- Conditional operators compare **primitives** to evaluate to **true** or **false**

    - == Equals

    - < Less than (is left less than right)

    - > Greater than

    - <= Less than OR equal

    - >= Greater than OR equal

    - != NOT equal (! is your NOT character)

- if statements execute a block of code, if **true** in the condition

- else statements execute a block of code, if **false** in an if condition (else is optional)

- Practice **drawing** the tree!!

    - Really, makes them a lot easier, especially when nested