Nested If Statements
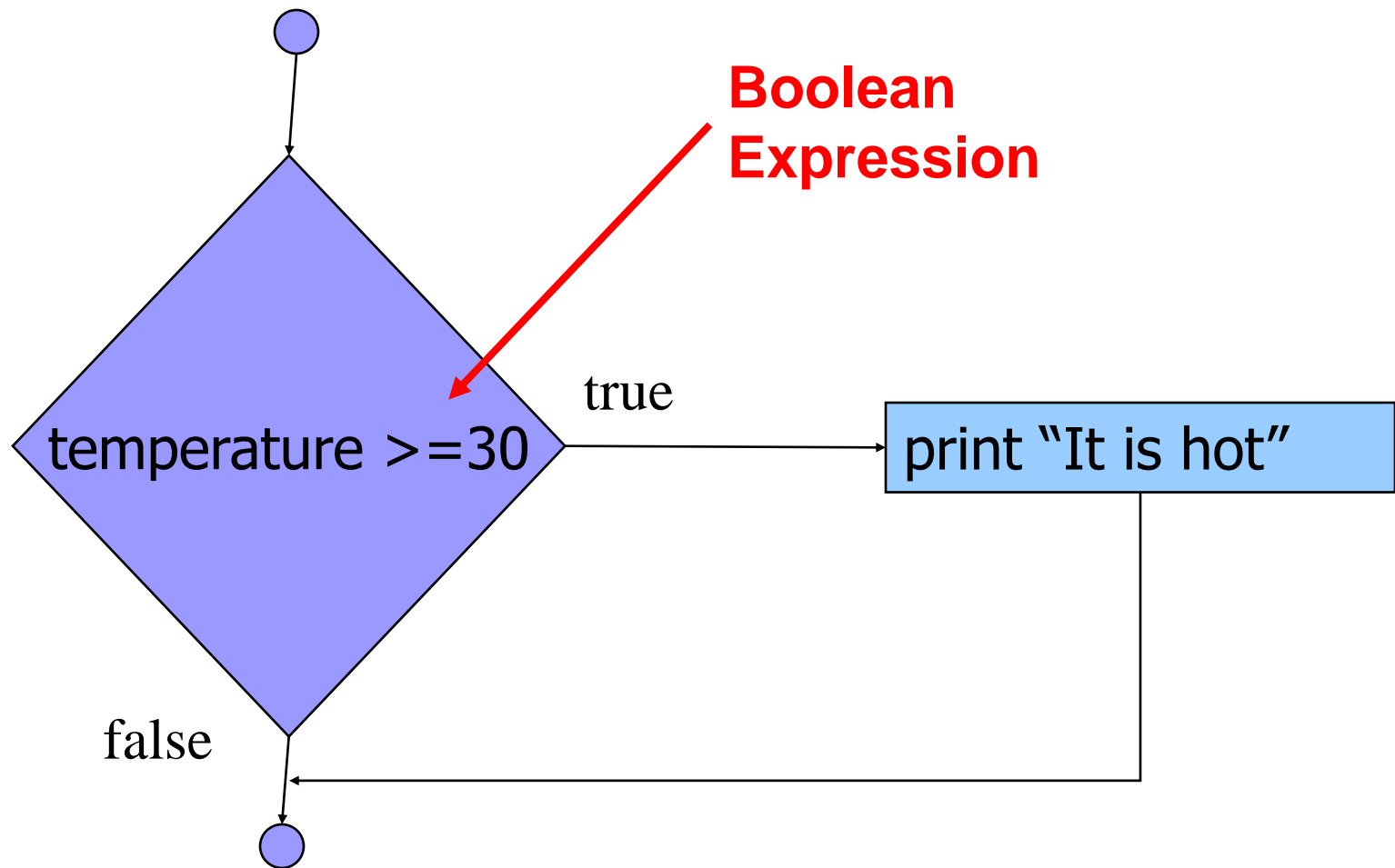
An simple if else statement checks to see if something is true or false. This can be illustrated in a flow chart diagram.

`if` Flow Chart

Boolean Expression

temperature >=30

true

print "It is hot"

false

Flow Chart

The diamond shape in the diagram is a boolean expression. Boolean expressions can be either true or false.

The rectangle in this flow chart represents an action or an outcome.

Flow charts are a good way to represent in pictures what is going on in a program.

In the previous slide the flow chart represents a question being asked: *Is the temperature greater than or equal to 30?*

if the answer to this question is true(remember boolean statements can only be true or false), then the text "it is hot" is printed, otherwise the program skips this section of code and the program continues.

```java
public class SimpleIf
{
    public static void main(String[] args)
    {
        int temp =32;

        if(temp >= 30)
        {
            System.out.println("It is hot");
        }
    }
}
```
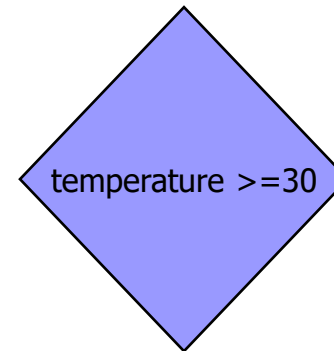
temperature >=30

print "It is hot"

The diamond represents a boolean expression,
in this case, is the temp >= 30

The rectangle represents an action or outcome

Consider a more complex example:
Imagine a program that took into account the temperature
and also whether or not it was sunny.

Before we do this, lets introduce a new variable type called
a boolean.

A boolean variable can only have one of two value, true or false.

```java
boolean sunny = true;

if(sunny == true)
{
    System.out.println("The sun is shining");
}
```

We use this expression in an if else statement as follows:

```java
boolean sunny = true;

if(sunny == true)
{
    System.out.println("The sun is shining");
}
else
{
    System.out.println("The day is cloudy");
}
```

Now imagine a scenario where the program takes into account the temperature and whether or not it is sunny.

First we check the temperature

```java
public class FirstNestedIf
{
    public static void main(String[] args)
        {
            int temp =32;

            if(temp >= 30)
            {
                System.out.println("It is hot");
            }
            else
            {
                System.out.println("It is not hot");
            }
        }
}
```

Now we ask a question within a question.

Once we establish whether the temp is >= 30, we can then ask if sunny == true or false.

outer if statement

nested
if statement

```java
public class FirstNestedIf
{
    public static void main(String[] args)
    {
        int temp =32;
        boolean sunny = true;
        if(temp >= 30)
        {
            System.out.println("It is hot");
            if(sunny == true)
            {
                System.out.println("It is sunny");
            }
        }
        else
        {
            System.out.println("It is not hot");
        }
    }
}
```

We create an if statement within an if statement,
what we call a nested if statement

With input values of 32 for temp
and true for the boolean variable sunny,
we get the following

```java
public class FirstNestedIf
{
    public static void main(String[] args)
    {
        int temp =32;
        boolean sunny = true;
        if(temp >= 30)
        {
            System.out.println("It is hot");
            if(sunny == true)
            {
                System.out.println("It is sunny");
            }
        }
        else
        {
            System.out.println("It is not hot");
        }
    }
}
```

Console ⊠

<terminated> FirstNestedIf [Java A

It is hot
It is sunny

We can add a nest if statement to the outer else as follows:

```java
public static void main(String[] args)
{
    int temp =28;
    boolean sunny = true;
    if(temp >= 30)
    {
        System.out.println("It is hot");
        if(sunny == true)
        {
            System.out.println("It is sunny");
        }
    }
    else
    {
        System.out.println("It is not hot");
    }
    if(sunny == true)
    {
        System.out.println("It is sunny");
    }
}
```

And if we modify the value of temp to 28….
we get:

Slight modification to nested if statements…can you spot them

Proper indentation makes it so much easier to see the structure of the program and spot errors-particularly those involving brackets.

```java
public static void main(String[] args)
    {
        int temp =28;
        boolean sunny = true;
        if(temp >= 30)
        {
            if(sunny == true)
            {
                System.out.println("It is hot and sunny");
            }
            else
            {
                System.out.println("It is hot and cloudy");
            }
        }
        else
        {
            if(sunny == true)
            {
                System.out.println("It is not hot and but it is sunny");
            }
            else
            {
                System.out.println("It is not hot and its cloudy");
            }
        }
    }
```

Challenge

Modify previous program so that it prompts user for inputs.

*Hint:
sunny = Boolean.*parseBoolean(strSunny);*

Answer on next slide: Try it on your own before peeking.

```java
import javax.swing.*;
public class FirstNestedIf
{
    public static void main(String[] args)
        {
            String strTemp, strSunny;
            int temp;
            boolean sunny;
            strTemp = JOptionPane.showInputDialog("Enter the Temperature");
            strSunny = JOptionPane.showInputDialog("Enter true/false for sunny");
            temp = Integer.parseInt(strTemp);
            sunny = Boolean.parseBoolean(strSunny);

            if(temp >= 30)
            {
                if(sunny == true)
                {
                    System.out.println("It is hot and sunny");
                }
                else
                {
                    System.out.println("It is hot and cloudy");
                }
            }
            else
            {
```

YouTube Tutorial Similar to previous slides

https://www.youtube.com/watch?v=x0eYqzUVPY8