Ch. 3 - Branches

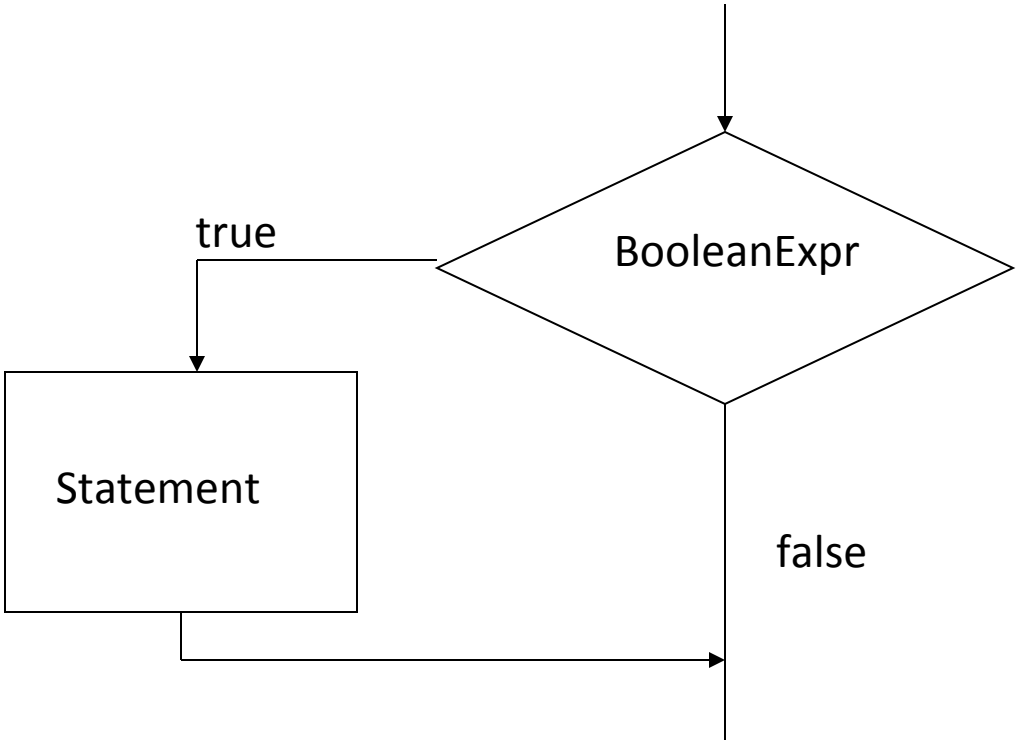
Recap of Ch. 1 & 2

- Comments and whitespace
- Primitive data types
- Variables
- Arithmetic Operators
- Boolean Operators
- Boolean Expressions
- Scanner class
- Math class
- Calling methods

The `if` statement

`if` (*BooleanExpression*)
Statement

Statement, here, could be several lines of statements.



Semicolons and the if statement

```
if (temp < 32)
   System.out.println("Warning: Below Freezing!");
   System.out.println("It's " + temp + ” degrees");

if (temp < 32) {
   System.out.println("Warning: Below Freezing!");
   System.out.println("You need a jacket!");
   System.out.println("Also some gloves.");
}
```

If there is only one statement in true branch, { } are optional.

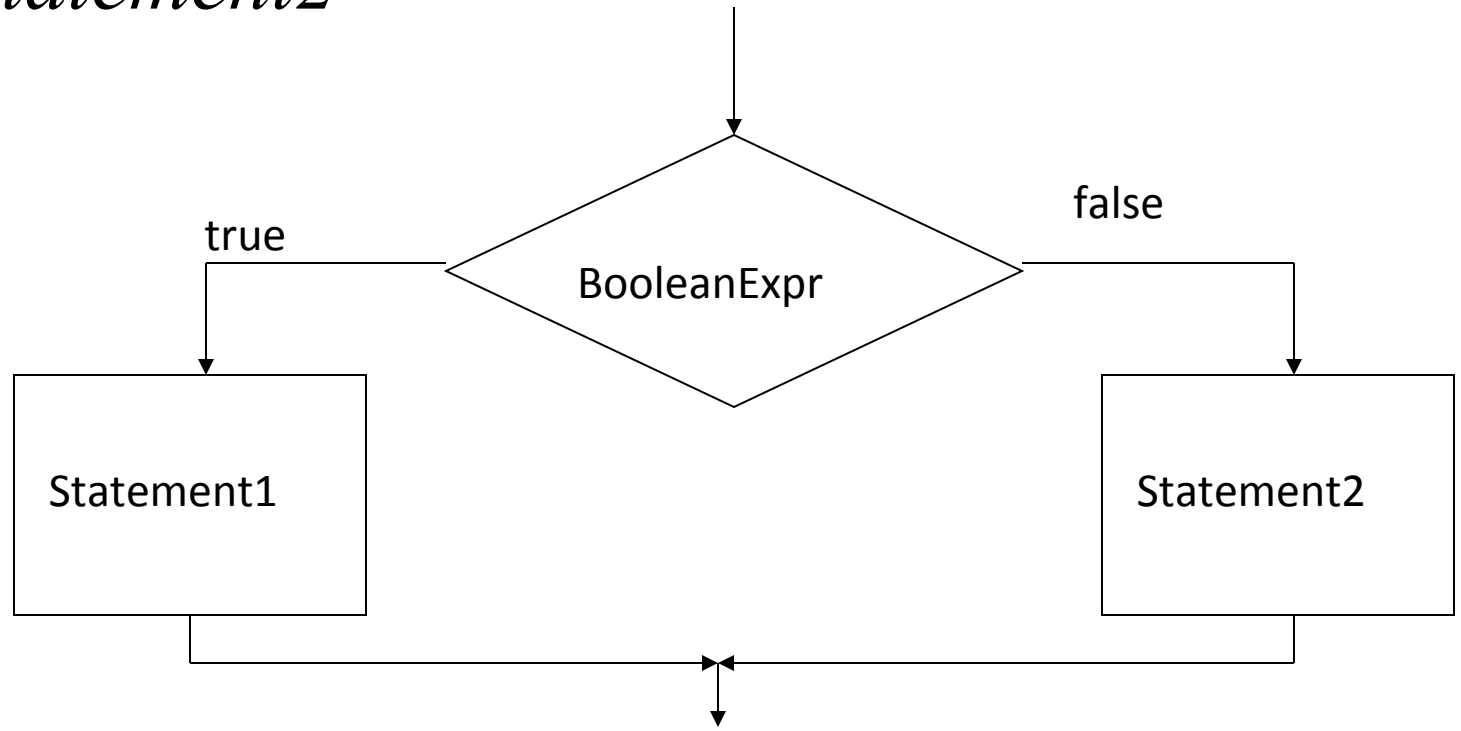
Common Error

```
if(temp <= 32);
    freezing = true;
```

The semicolon by itself is known as the empty statement.

The `if-else` statement

`if` (*BooleanExpression*)
  Statement1
`else`
  Statement2



```
int userAge = 0; // Years

System.out.print("Enter age: ");
userAge = keyboard.nextInt();

if (userAge < 18) {
   System.out.println("You are not allowed to vote.");
}
else {
   System.out.println("You are allowed to vote.");
}
```

Finding the Minimum

```
if(x < y)
   min = x;
else
   min = y;


// What if x and y are equal?
```

Nested if-statements

```
if(temp > 32) {
    if(temp < 212) {
        state = “liquid”;
    }
    else {
        state = “gas”;
    }
}
else {
    state = “solid”;
}
```

Nested if-statements

```
if(temp >= 212) {
    state = “gas”;
}
else {
    if(temp >=32) {
        state = “liquid”;
    }
    else {
        state = “solid”;
    }
}
```

If-else-if

```
if(temp >= 212) {
    state = “gas”;
}
else if(temp >= 32) {
       state = “liquid”;
}
else {
    state = “solid”;
}
```

If-else-if Rule

```
// The indentation here is deliberately confusing
if(temp < 212)
   if(temp <= 32)
      state = “freezing”;
else
   state = “liquid”;
else
   state = “gas”;
```

Rule: when there’s a choice of which else goes with which if, the else goes to the most recent if.

Shortcut

- Ternary Condition Operator

```
cond ? exp1 : exp2
```

```
int x, y = 6;
x = (y > 5 ? 10 : 20); // x is assigned 10

// shortcut for
if(y > 5)
   x = 10;
else
   x = 20;
```

Shortcut

```
x = (y < 5) ? 10 : (y < 10) ? 20 : (y < 15) ? 30 : 40;

// Convert this into if-else-if statement
```

```
int x = 5;
if (expression1){
  if (expression2)
    x = 10;
}
System.out.println(x);
```

```
int x = 5;
if (expression1 && expression2)
    x = 10;
System.out.println(x);
```

A. The above two always print the same thing.

B. The above two sometimes print the same thing, depending upon the values of expression1 and expression2.

C. The above two are totally different.

```
int x = 5;
if (expression1){
  if (expression2)
    x = 10;
}
else x = 20;
System.out.println(x);
```

```
int x = 5;
if (expression1 && expression2)
    x = 10;
else x = 20;
System.out.println(x);
```

A. The above two always print the same thing.

B. The above two sometimes print the same thing, depending upon the values of expression1 and expression2.

C. The above two are totally different.

Switch Statement

```
int data = 3;
switch(data) {
   case 0:
      // statements
      break;
   case 1:
      // statements
      break;
   case 2:
      // statements
      break;
   default:
      // these statements execute if all the cases failed
      break;
}
```

Switch Statement

```
String data = “fall”;
switch(data) {
    case “fall”:
        // statements
        break;
    case “winter”:
        // statements
        break;
    case “spring”:
        // statements
        break;
    default:
        // these statements execute if all the cases failed
        break;
}
```

Switch Statement

- Supported data types:
  - byte
  - short
  - char
  - int
  - String
  - enum types
  - wrapper classes

String equality

- How can we tell that 2 strings are equal?

```
String str1 = “hello”;
String str2 = “hello”;
if (str1 == str2) // WRONG!!!
   …
```

String equality

```
String str1 = “hello”;
String str2 = “hello”;
if (str1.equals(str2)) // Correct ☺
    …
```