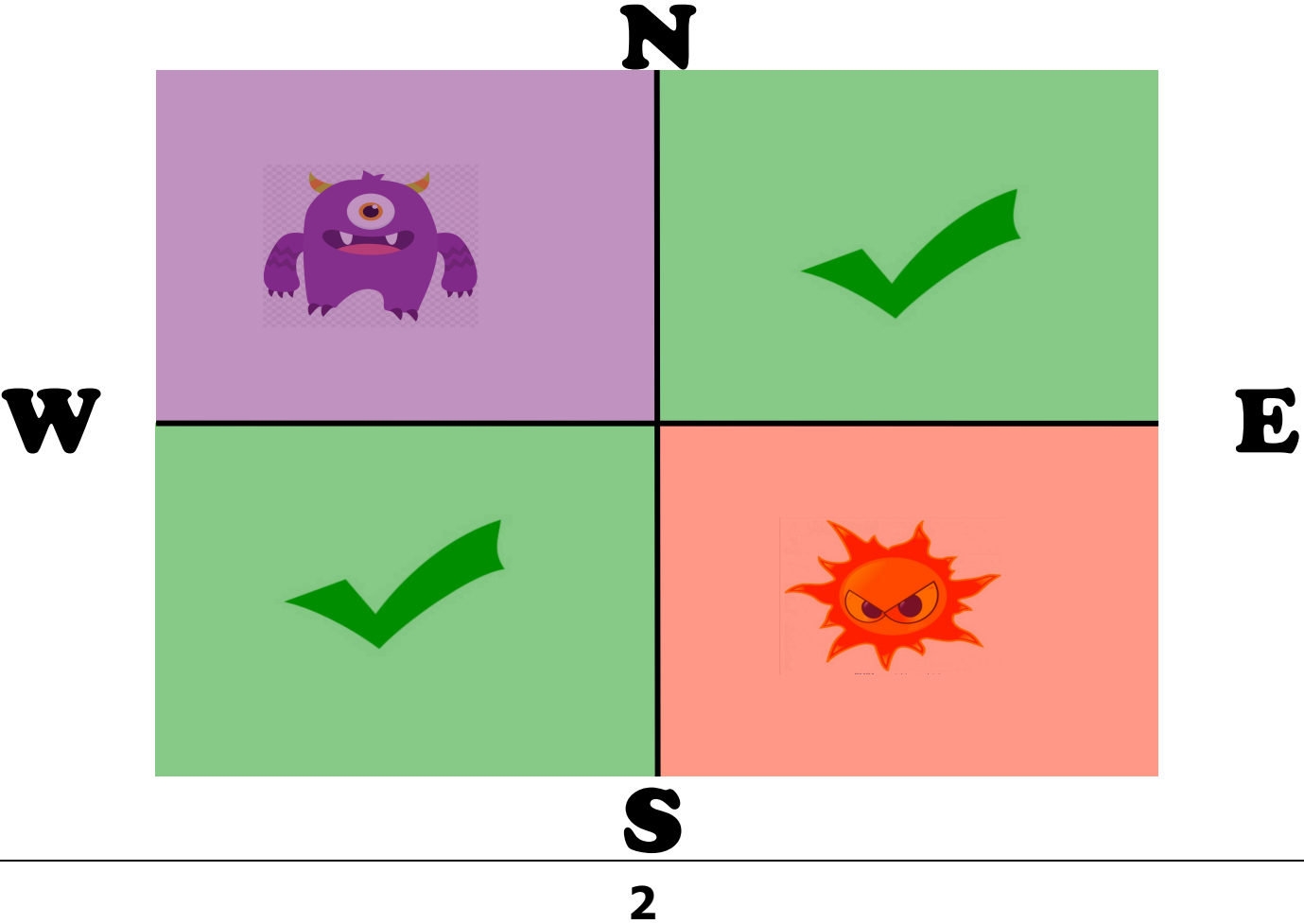
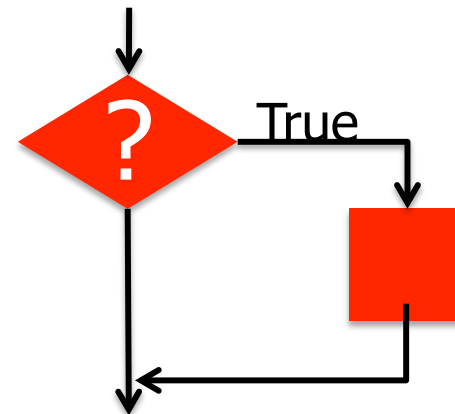
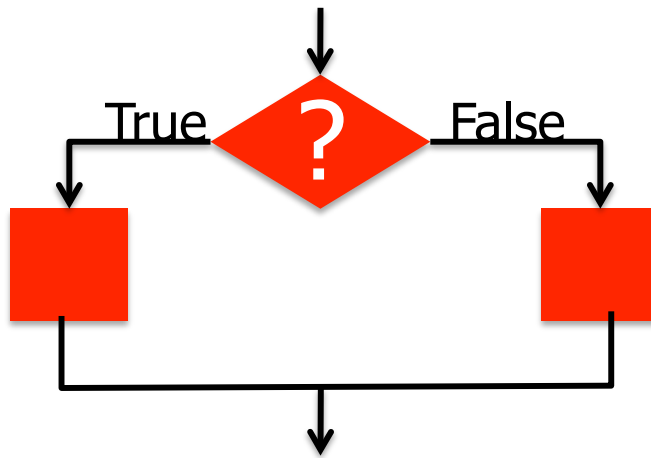


What if?

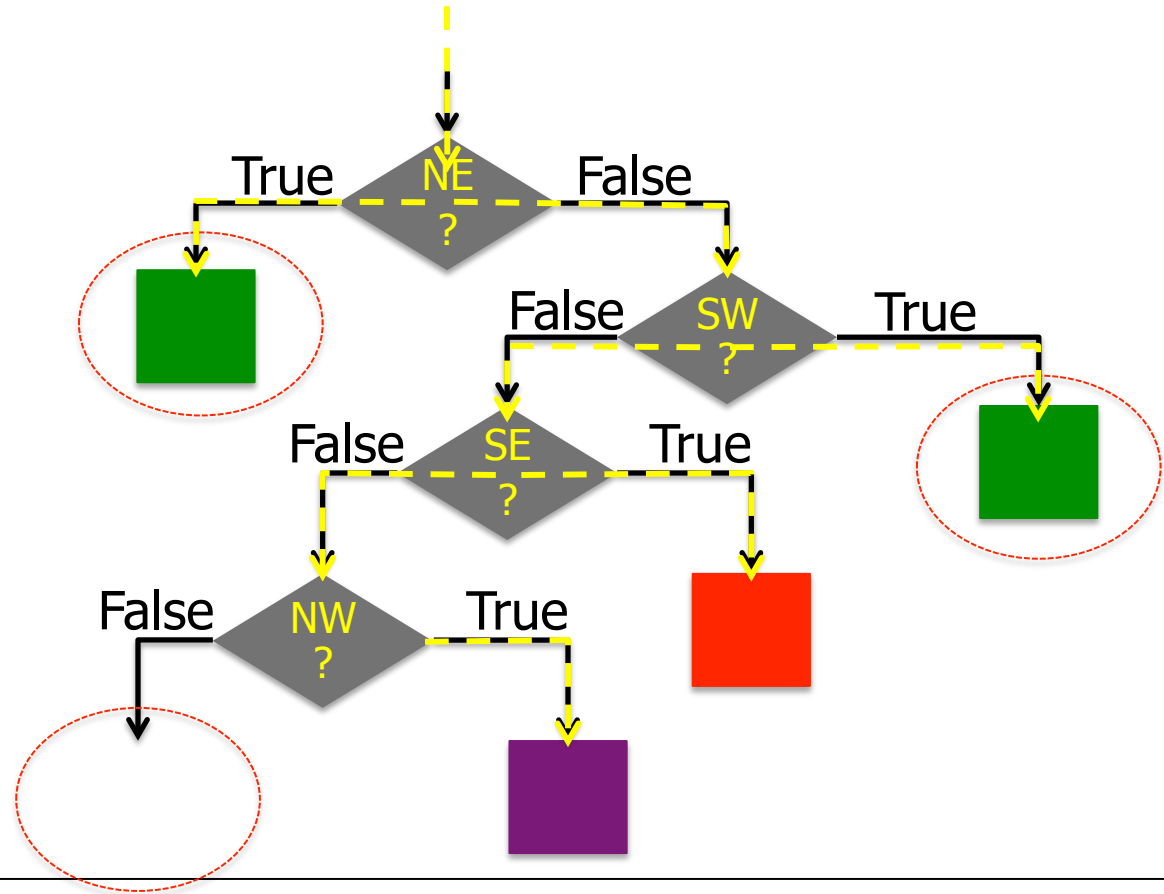
- **Set max to 10 sometimes, and sometimes I don't?**
- **Print out only the names of players with scores above 90?**



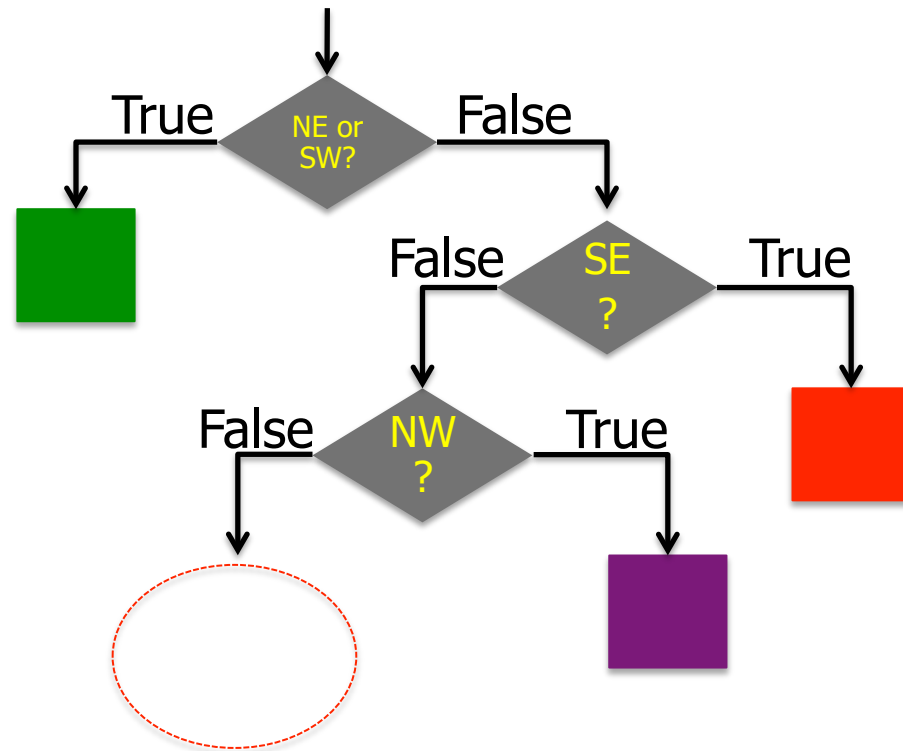
Use symbols like these



Did you see
the problem
this way?



Combine two
conditions
that result in
the same
outcome



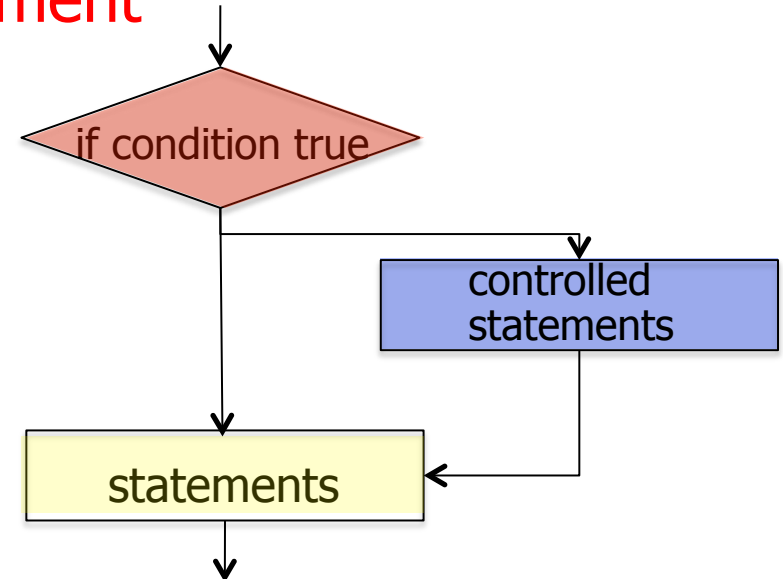
The `if` statement

Executes a block of statements only
if a test is true

```
if (test) {  
    statement(s);  
}  
statement(s);
```

Example:

```
int total = console.nextInt();  
if (total >= 10) {  
    Out.println("You have selected too many items.");  
    total = max;  
}  
Out.print(total);
```



How do we create these "tests"?

Relational expressions

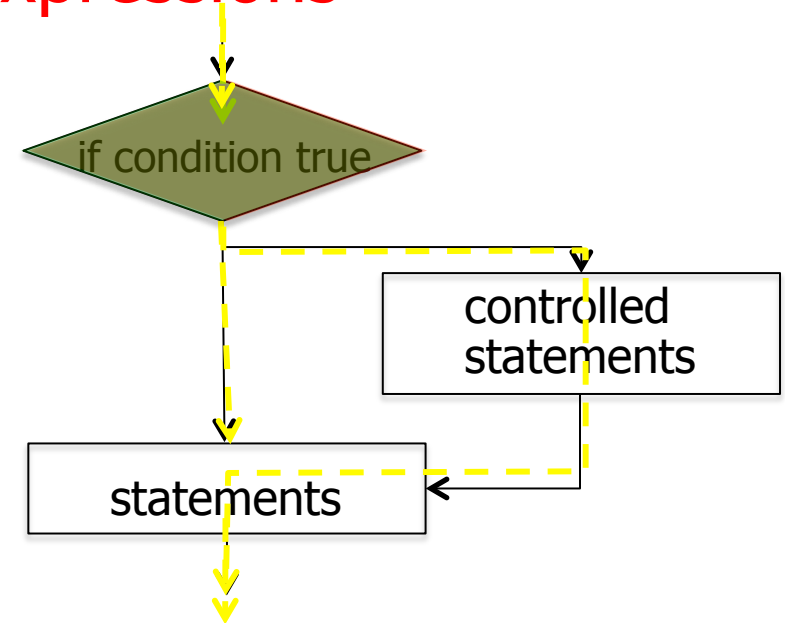
- These tests are called `boolean` expressions.
- Tests use *relational operators*:

operator		example	result
<code>==</code>	equals	<code>3 * 2 == 6</code>	true
<code>!=</code>	not equal	<code>3 * 2 != 7</code>	true
<code>< , ></code>	less than, greater than	<code>8 < 4</code>	false
<code><= , >=</code>	less than or equal to, greater than or equal to	<code>9 <= 9</code>	true

Example Boolean Expressions

```
x >= 10  
max < cur  
sum != 0  
age == 65
```

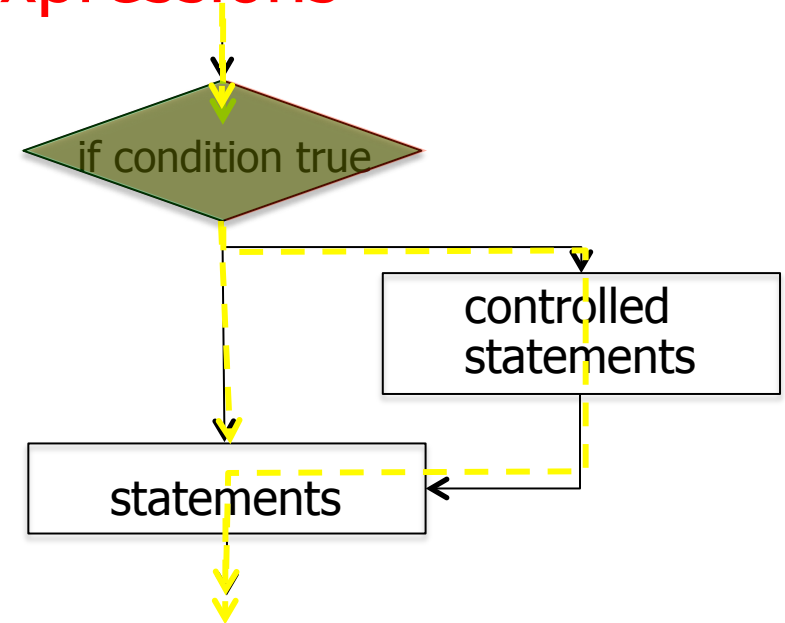
```
if ( x >= 10 ) {  
    controlled statements  
}  
statements
```



Example Boolean Expressions

```
x >= 10  
max < cur  
sum != 0  
age == 65
```

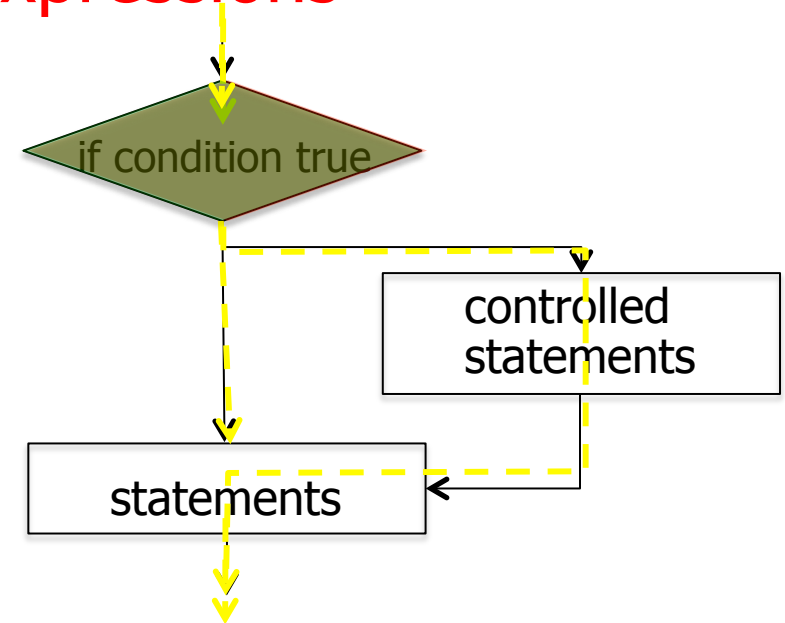
```
if ( max < cur ) {  
    controlled statements  
}  
statements
```



Example Boolean Expressions

```
x >= 10  
max < cur  
sum != 0  
age == 65
```

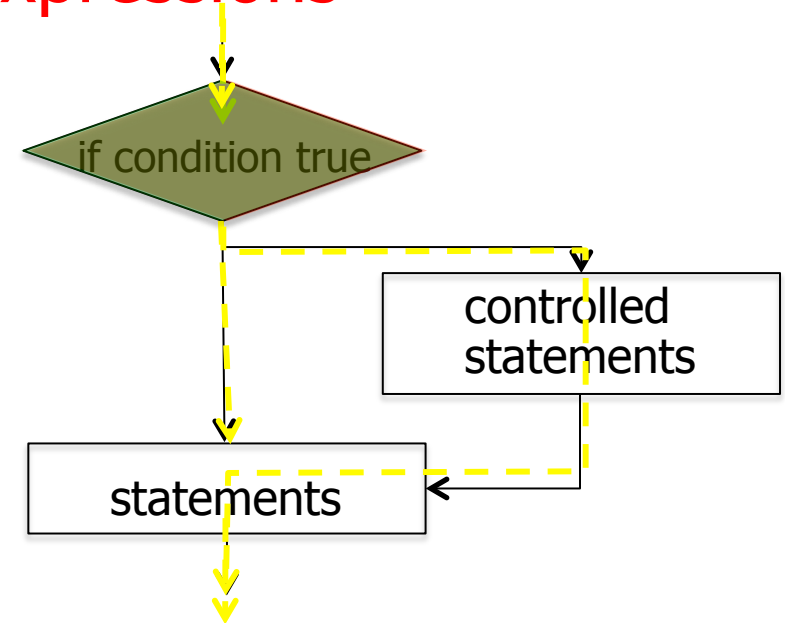
```
if ( sum != 0 ) {  
    controlled statements  
}  
statements
```



Example Boolean Expressions

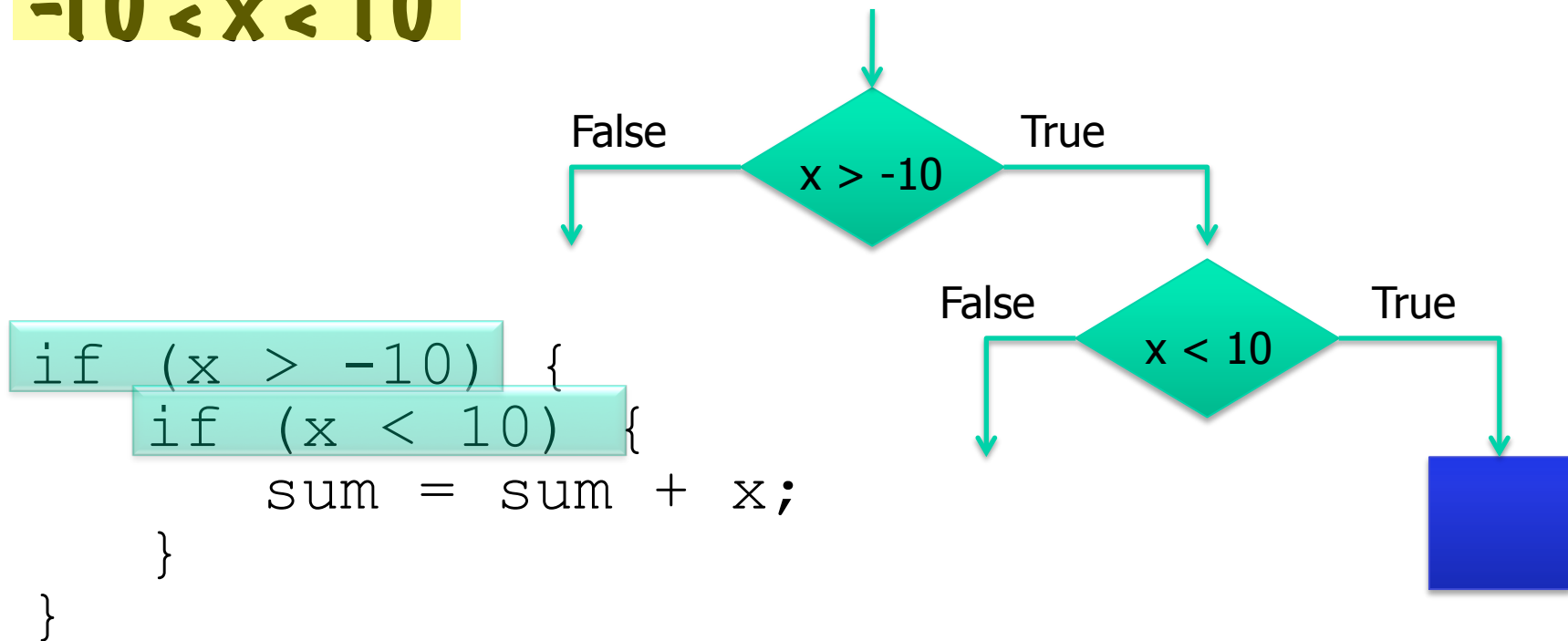
```
x >= 10  
max < cur  
sum != 0  
age == 65
```

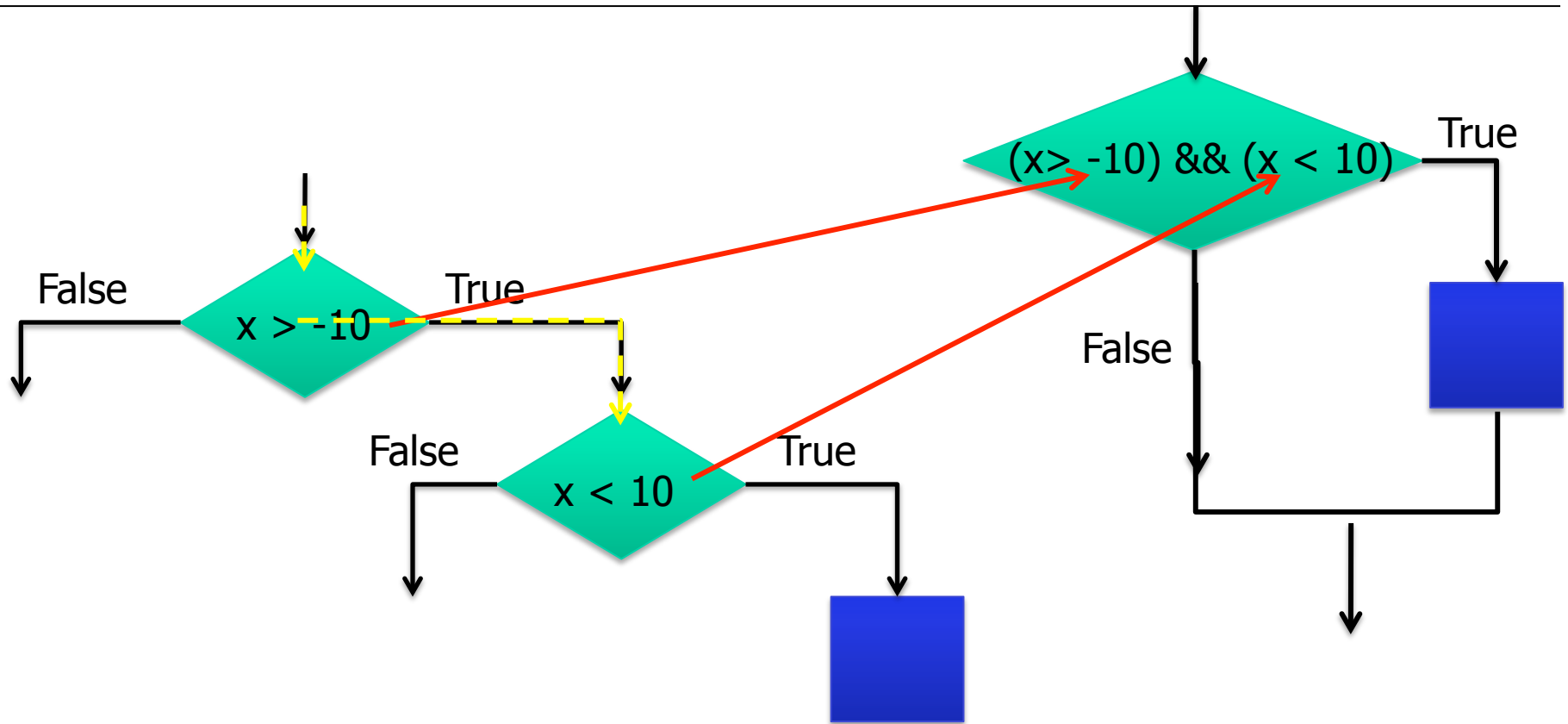
```
if ( age == 65 ) {  
    controlled statements  
}  
statements
```



How do we test a value to see if it is within a certain range?

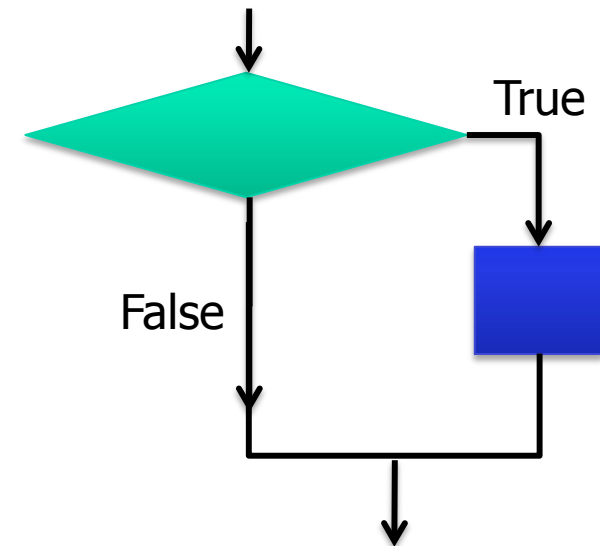
$-10 < x < 10$





```
if ((x > -10) && (x < 10))  
{  
    sum = sum + x;  
}
```

```
if (x > -10) {  
    if (x < 10) {  
        sum = sum + x;  
    }  
}
```



Logical operators: `&&`, `||`, `!`

Conditions can be combined using *logical operators*:

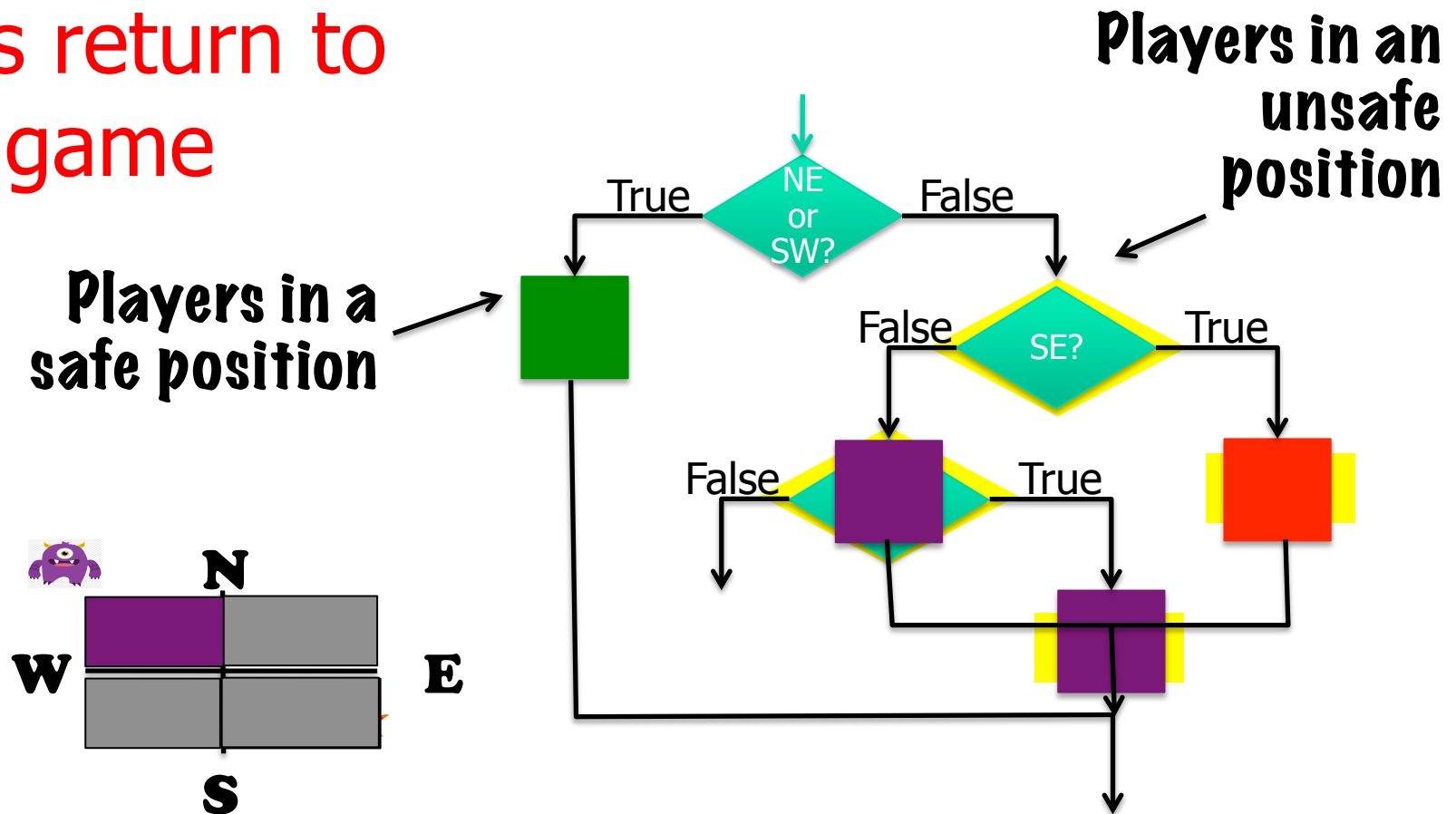
Operator		Example	Result
<code>&&</code>	and	<code>(2 == 3) && (-1 < 5)</code>	false
<code> </code>	or	<code>(2 == 3) (-1 < 5)</code>	true
<code>!</code>	not	<code>!(2 == 3)</code>	true

“Truth tables” for each, used with logical values p and q :

value of p	value of q	p && q	p q
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

value of p	value of !p
true	false
false	true

Let's return to
our game



In our next lesson...

