

FRC Java

Lesson 2

Making Decisions

If Statements, Comparison Operators, Switch

Gus Michel III

3946 Tiger Robotics – slidell-robotics.com

IF statement

```
if(<condition>) {  
    <code>;  
}
```

- Checks the condition.
 - If it is true, it runs the code inside
 - If it is false, the code inside is skipped over

IF ELSE Statement

```
if(<condition>) {  
    <code>;  
} else {  
    <code>;  
}
```

- If condition is false
 - Run code in else
- If condition is true
 - Run code in if (NOT the code in else)

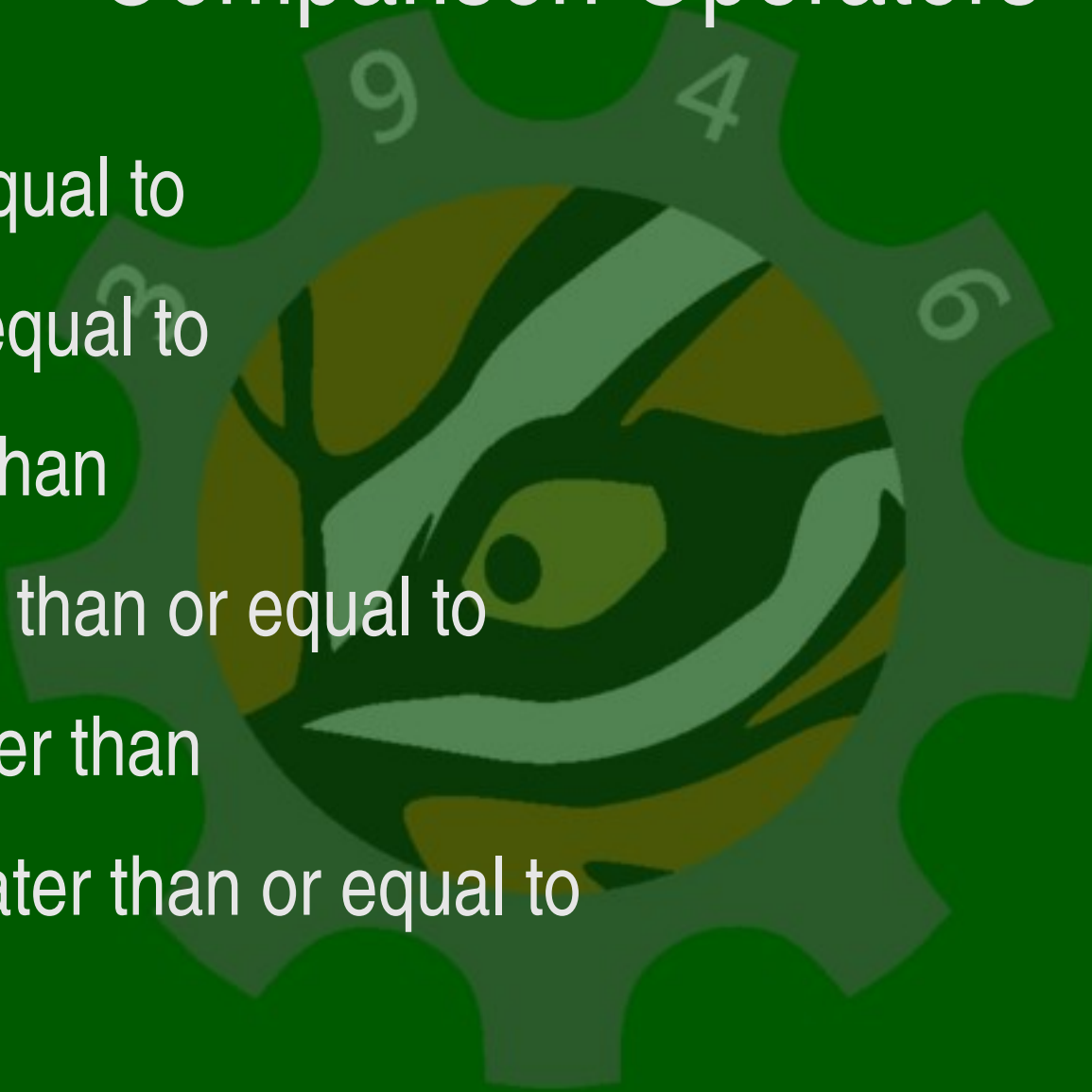
IF, ELSE IF, ELSE Statements

```
if(<condition>) { <code>; }  
else if(<condition>) {  
    <code>;  
} else { <code>; }
```

- When If is false, check else if, if that is false run else.
- Can have without else as well.
- You can have more than one else if()

Comparison Operators

- `==` - is equal to
- `!=` - not equal to
- `<` - less than
- `<=` - less than or equal to
- `>` - greater than
- `>=` - greater than or equal to



Using Comparison Operators

- `If(43 == 43) {}` - true
- `If(24 != 24) {}` - false
- `If(43 < 50) {}` - true
- `If(50 <= 49) {}` - false
- `If(38 > 2) {}` - true
- `If(32 >= 70) {}` - false
- Comparing Strings uses the `equals()` method
 - `<String or Variable>.equals(<String or Variable>);`

Boolean Operators



- && - and
- || - or
- Using Them
 - If(4>3 && 5>3) {} - true
 - If(4>3 && 3>6) {} - false
 - If(7>3 || 4 > 8) {} - true
 - If(4 == 5 || 5 < 99) {} - true
 - If(4 != 4 || 6 > 2) {} - false
 - If(4 != 7 || 4 > 9) {} - true

Examples

- ```
if(numLeds % 2 == 0) {
 System.out.println("Even Number of LEDs");
}
```
- I used this for checking how many LEDs I had


```
if(interval <= 100 && interval >=500) {
 System.out.println("Input is in range");
} else {
 System.out.println("Input out of range");
}
```

- I used this for checking my delay timer was ok



# Switch Statement

```
switch(<variable>) {
 case <value>:
 <code>
 break;
 case <value>:
 <code>
 break;
 default:
 <code>
 break;
}
```

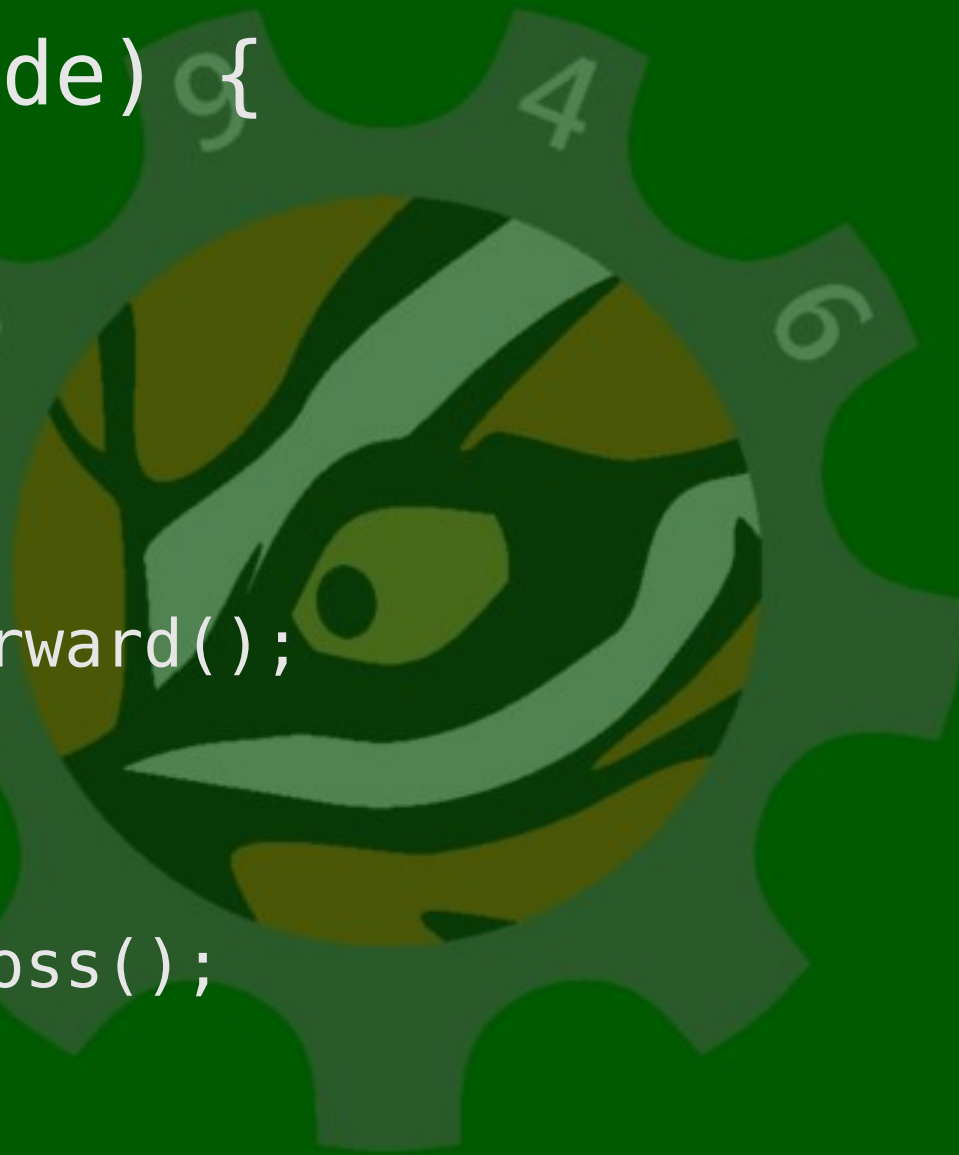


# Switch Statement

- Check bunch of inputs and perform code based on it
- `break;` is used to exit the Switch Statement
  - If left out it would also perform all options after the first.
- Can have as many case: as you want.
- `default:` can be used if you want an “else” on the end.

# Switch Statement Example

```
switch(mode) {
 case 0:
 off();
 break;
 case 1:
 pulseForward();
 break;
 case 3:
 crissCross();
 break;
}
```



# Assignment

- Use an if statement to check for positive, negative, or 0
  - If Positive print “POS” to screen, if Negative print “NEG”, 0 print ZERO.
- Use a switch statement to check if a given number is the same as your house number, age, or other possible number.
  - Print output accordingly.