Flow Control

Java Programming for Absolute Beginners Phase I

Booleans

- A Boolean is a data type whose value can be either true or false.
- There are several operators that can be performed on Booleans:
  - AND: returns true if both given Booleans are true
  - OR: returns true if at least one given Boolean is true
  - NOT: negates the given Boolean, if it is true, the operation returns false, and if it is false, the operation returns true.

Truth Tables

- The different results of different Boolean operations can be expressed in tables called Truth Tables.

A	B	A OR B
T	T	T
T	F	T
F	T	T
F	F	F

A	B	A AND B
T	T	T
T	F	F
F	T	F
F	F	F

A	NOT A
T	F
F	T

Relational Operators

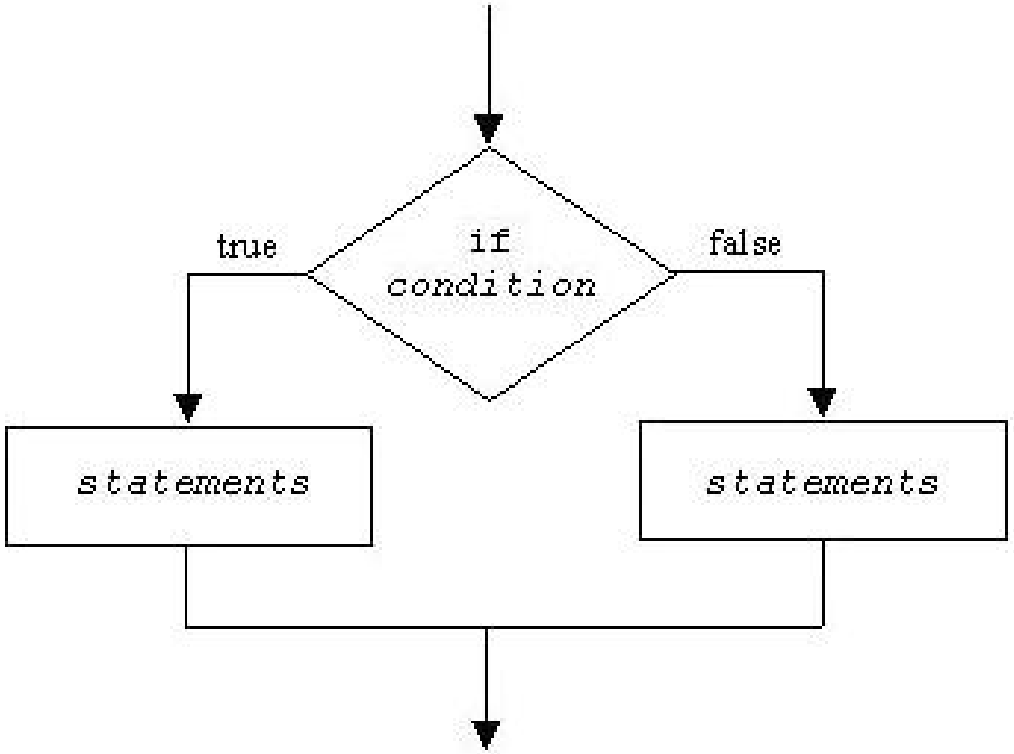
- These relational operators result in Boolean values.
- They can be put together for Boolean expressions.

Operator	Use	Description
>	op1 > op2	op1 is greater than op2
>=	op1 >= op2	op1 is greater than or equal to op2
<	op1 < op2	op1 is less than op2
<=	op1 <= op2	op1 is less than or equal to op2
==	op1 == op2	op1 and op2 are equal
!=	op1 != op2	op1 and op2 are not equal

Boolean Expressions

- A Boolean expression is an expression that results in a Boolean value.
- Here are some examples of mathematical expressions that translate to Boolean expressions:
  - $5 < x \leq 10$: `x > 5 && x <= 10`
  - $x \neq 20$: `x != 20`

So What are Booleans Used For?



- So far, all of our programs have executed from top to bottom, the same way every time, this can be useful for a variety of different problems, but the power of computer shines when the program can make decisions.
- Our programs could now make all sorts of decisions, and the flow of the program can consist of branching and looping.

If Statements

- In Java, if statements are used for flow control.
- These are the four types: if, if-else, if-else if, if-else if-else

```
<statements>
if(<booleanExpr>) {
     <statements>
}
<statemets>
```

```
<statements>
if(<booleanExpr>) {
     <statements>
} else {
     <statements>
}
<statements>
```

```
<statements>
if(<booleanExpr>) {
     <statements>
} else if(<booleanExpr>) {
     <statements>
}
<statements>
```

```
<statements>
if(<booleanExpr>) {
     <statements>
} else if(<booleanExpr>) {
     <statements>
} else {
     <statements>
}
<statements>
```

If Statement

```
<statements_1>
if(<booleanExpr>) {
    <statements_2>
}
<statements_3>
```

1. Execute <statements_1>
2. Check <booleanExpr>, if it is true, go to 3, otherwise go to 4.
3. Execute <statements_2>
4. Execute <statements_3>

If-Else Statement

```
<statements_1>
if(<booleanExpr>) {
    <statements_2>
} else {
    <statements_3>
}
<statements_4>
```

1. Execute <statements_1>
2. Check <booleanExpr> if it is true, go to three, if it is false, go to 5.
3. Execute <statements_2>
4. Go to 6
5. Execute <statements_3>
6. Execute <statements_4>

If-Else If Statement

```
<statements_1>
if(<boolExpr_1>) {
    <statements_2>
} else if (<boolExpr_2>) {
    <statements_3>
}
<statements_4>
```

1. Execute <statements_1>
2. Check <boolExpr_1>, if true, go to 3, if false, go to 5
3. Execute <statements_2>
4. Go to 7
5. Check <boolExpr_2>, if true, go to 6, if false, go to 7
6. Execute <statements_3>
7. Execute <statements_4>

Note: You can have as many else if blocks after an if statement as you want.

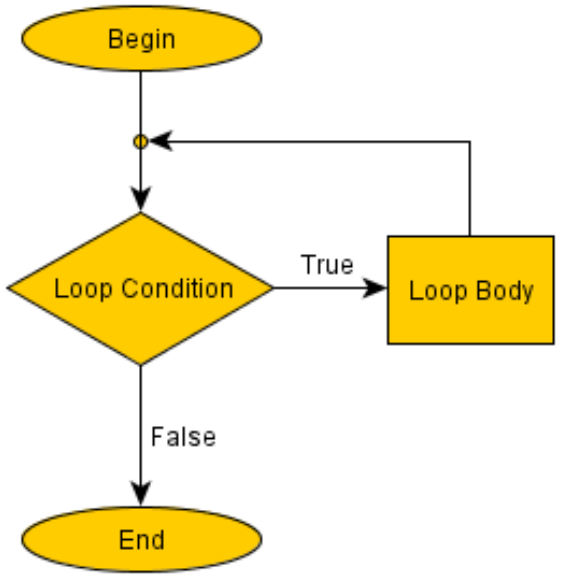
If-Else If-Else Statement

```
<statements_1>
if(<boolExpr_1>) {
    <statements_2>
} else if (<boolExpr_2>) {
    <statements_3>
} else {
    <statements_4>
}
<statements_5>
```

1. Execute <statements_1>
2. Check <boolExpr_1>, if it is true, go to 3, if it is false, go to 5
3. Execute <statements_2>
4. Go to 9
5. Check <boolExpr_2>, if it is true, go to 6, if it is false, go to 8
6. Execute <statements_3>
7. Got to 9
8. Execute <statements_4>
9. Execute <statements_5>

Note: You can have as many else if blocks between the if and the else, but only one else is allowed.

Loops



- Computers are very good at executing the same task over and over again.
- They can even perform the same task millions of times, with only a slight variation between each **iteration**.
- Programmers accomplish this using a **loop**, which is a block of code that will be repeated, rather than writing the same code millions of times.
- There are several types of loops, today we will be looking at the while loop and the for loop.

While Loop

```
<statements_1>
while (<boolExpr>) {
    <statements_2>
}
<statements_3>
```

1. Execute <statements_1>
2. Check <boolExpr>, if it is true, go to 3, if it is false, go to 5.
3. Execute <statements_2>
4. Go to 2
5. Execute <statements_3>

For Loop

```
<statements_1>
for (<initializer>; <boolExpr>; <incrementor>) {
    <statements_2>
}
<statements_3>
```

1. Execute <statements_1>
2. Execute <initializer>
3. Check <boolExpr>, if it is true, go to 4, if it is false, go to 7
4. Execute <statements_2>
5. Execute <incrementor>
6. Go to 3
7. Execute <statements_3>

Any Questions?