Algorithms

- actions
- order
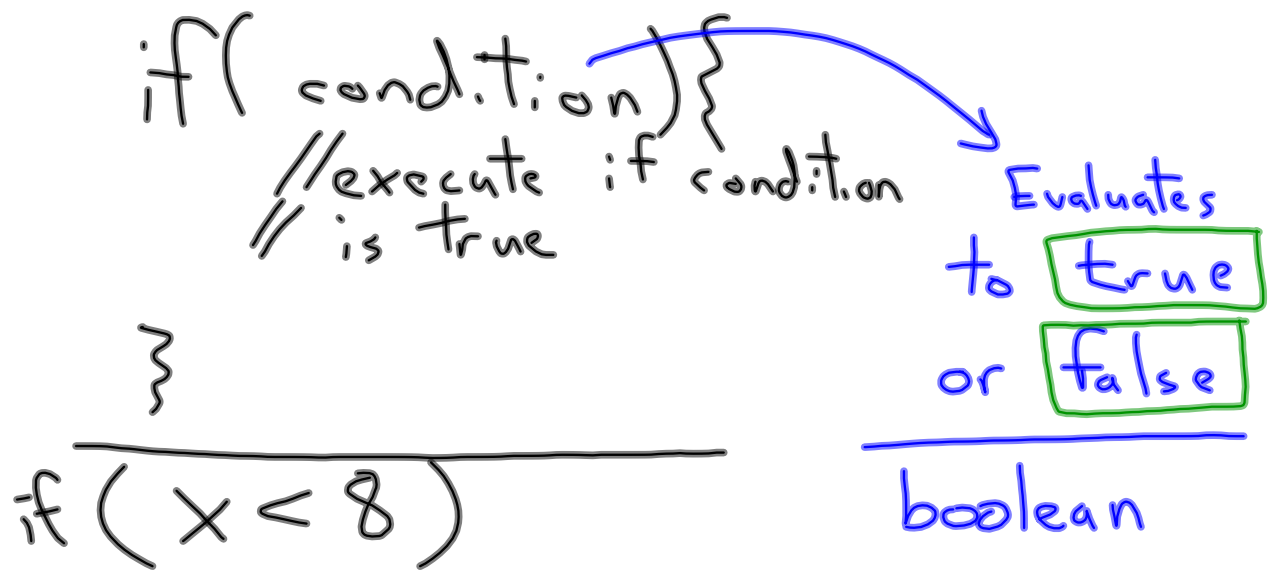
Control Structures

sequential action

selection/decision – if

subroutines – method calls

repetition – loops

```
if( condition ){
    //execute if condition
    // is true

}
```

if (x < 8)

Evaluates to **true** or **false**

boolean

```
if (x < 8)
    S.o.pln("x < 8");
S.o.pln("Done");
```

x: 3   X: 19
――――――   ――――――
"x < 8"   "Done"

"Done"

```
if ( x < 8 ){
    S.o.pln("x < 8");
    S.o.pln("Done");
}
;
```

x : 3   x : 19
――――――   ――――――
"x < 8"
"Done"
:
.   :
.

;

$x = 3$;;;;;;

WARNING!!!

```
if( x < 8 ) {
    S.o.pln("x < 8");
}
S.o.pln("Done");
```

Blocks

```
{
    int x = 3;
}
System.out.println(x);
            ?? 
```

```
int x = 0;
if (           )
{
    x = 3;
}
S.o.pln(x);
```

```
public double setBalance(double newB)
{
  .
balance=newB;  double value = balance;
               if(newB < 0){
                    S.o.pln("balance < 0!");
                    balance = value;
               }
}
```

```
int  x = ...

if( x < 0 ){
    x = -1 * x;
}
S.o.pf.("| x | = %d", x);
```

```
if (         )
    ↓
  <        <=        } int
  >        >=          double
  !=       ==
─────────────────────────────────
      String  line = in.nextLine();

   if ( line == "quit" ) {

   }

   if (line.equals("quit")) {
                          ↖ returns true or
                             false

   }
```

```
if( x != 8){    //run if cond is true
    S.o.pln(" x is not 8 ");
}
else {          //run if cond is false
    S.o.pln(" x is 8 ");
}
S.o.pln("Done");
```

```
if( x != 8)
    S.o.p(" not 8 ");
else
    S.o.p(" 8 ");
```

UML

```
      ●
      │
      ▼
 ┌─────────────┐
 │  print   3  │
 └─────────────┘
      │
      ▼
 ┌─────────────┐
 │  print   4  │
 └─────────────┘
      │
      ▼
      ●
```

```
         ●
         │
         ▼
       ◇ t < 8 ◇ ──── true ──→ ┌──────────┐
         │                     │ print < 8│
       false                   └──────────┘
         │                          │
         ▼         ┌────────────────┘
    ┌──────────────┐
    │  Print Done  │
    └──────────────┘
         │
         ▼
```