Another loop: Do-While

do {
   statement(s)
} while (condition);

- Do the statement/block at least once
- Evaluate the condition. If it is
  - true: re-execute statement(s); repeat step 2
  - false: we’re done with the loop

```java
int x = 0;
do{
   x = x + 1;
} while (x < 3);
```

Break and Continue Statements

- `break` and `continue` are Java statements
- Are also “flow control” statements
  - if, while, do-while, for, return
- A break “breaks you out” of the closest enclosing loop
- A continue is a shortcut to the next iteration of the loop
- A loop may have
  - Zero or more break statements
  - Zero or more continue statements

while-loop with break, continue

```java
while (condition1){
   if (condition2)
      continue; // go up and re-evaluate condition1
   if (condition3)
      break; // exit the loop
   .
   .
   .
}
// after a break statement, execution resumes here
```
while-loop with break, continue example

```java
int x = 1;
while (x <= 10){
    if (x % 2 == 0){
        System.out.println(x); break;
    } else {
        System.out.println(x); continue;
    }
    x = x + 1;
}
```

What will happens with break vs. continue?

---

for-loop with break, continue

```java
for (expr1; condition1; expr2){
    if (condition2)
        continue; // evaluate expr2, then condition1
    if (condition3)
        break; // exit the loop
    ...
    ...
} // after a break statement, execution resumes here
```

---

Labels

- Optionally you can also provide a label that will cause break/continue statement to exit to an outer level of nested loop

```java
outside: // label
for (x=0;x<10;x++) {
    for (y=0;y<10;y++) {
        f(x,y); // do something with x and y
        if (CPU.readPin(CPU.pin1)) break; // skip to next X early
        if (CPU.readPin(CPU.pin3)) break outside; // stop both loops
    }
}
```

---

Example with label & continue

```java
outside: // label
for (x=0;x<10;x++) {
    for (y=0;y<10;y++) {
        if (CPU.readPin(CPU.pin2)) continue; // skip this value of Y
        if (CPU.readPin(CPU.pin3)) continue outside; // skip to next X
        f(x,y); // do something with x and y
    }
}
```