`for` loop

```
for (int i = 0; i < count; i++) {
   …
}
```

```
private void turnRight() {
   for (int i = 0; i < 3; i++) {
       turnLeft();
   }
}
```

while loop

```
while ( condition ) {
   …
}
```

```
private void moveToWall() {
   while (frontIsClear()) {
       move();
   }
}
```

Conditions Karel can check for

Test	*Opposite*	*What it checks*
`frontIsClear()`	`frontIsBlocked()`	Is there a wall in front of Karel?
`leftIsClear()`	`leftIsBlocked()`	Is there a wall to Karel's left?
`rightIsClear()`	`rightIsBlocked()`	Is there a wall to Karel's right?
`beepersPresent()`	`noBeepersPresent()`	Are there beepers on this corner?
`beepersInBag()`	`noBeepersInBag()`	Any there beepers in Karel's bag?
`facingNorth()`	`notFacingNorth()`	Is Karel facing north?
`facingEast()`	`notFacingEast()`	Is Karel facing east?
`facingSouth()`	`notFacingSouth()`	Is Karel facing south?
`facingWest()`	`notFacingWest()`	Is Karel facing west?

This is **Table 1** on page 18 of Karel course reader

`if` statement

```
if ( condition ) {
   …
}
```

```
private void safePickUp() {
   if (beepersPresent()) {
       pickBeeper();
   }
}
```

`if-else` statement

```
if ( condition ) {
   …
} else {
   …
}
```

```
if (frontIsClear()) {
   move();
} else {
   turnLeft();
}
```