Side effects

A side effect occurs when evaluation of an expression changes a value. For example, this assignment:

```
b[i++]= 5;
```

has the side effect of adding 1 to `i`. Here's another one. A side effect occurs if evaluation of the following loop condition, which is a call on function `f`, changes some variable:

```
while (f(...)) { ... }
```

We rarely use such side effects. They can be confusing, and a side effect could could be disastrous unless it is very well documented and the programmer is aware of what is happening. In fact, the theory of proving programs correct tells us that formal proofs of correctness when side effects may occur are far more complex than when side effects are not allowed.

Standard paradigms

One does see some standard paradigms that use side effects. For example, below is a loop that reads and prints lines of a `BufferedReader br`; the loop condition has the side effect of assigning to variable `line`.

```
String lin;
while ((lin= br.readLine()) != null) {
  System.out.println("lin is " + line);
}
```

We never use this paradigm, preferring to write the loop as follows, in a way that has no side effects, even though it is longer that the one above.

```
String lin= br.readLine();
// invariant: all lines that have been read in, except lin, have been printed
while (lin != null) {
  System.out.println("lin is " + line);
  lin= br.readLine();
}
```

Benevolent side effects

There is one class of side effect that is OK. A *benevolent side effect* is one that changes the state but not in a way that the user can observe. A benevolent side effect can be used to make a program more efficient. Here is an example.

> Consider implementing a class that maintains a set of questions-and-answers. When someone asks a question, the corresponding answer is given. The questions-and-answers are implemented in an array, and one has to search the list in linear fashion to find a question.
>
> It is better if more frequently asked questions are at the front of the list, so they are more efficiently found. This can be achieved by the following: Whenever a question is asked, move it up one position in the array. This changes the representation of the set of questions-and-answers, but not the set itself; the user cannot notice the change. This is a benevolent side effect.