

Review Questions

24.

Java's break has the ability to jump to labels. For example the code

```
label1:
    for(int i = 0; i < 3; i++)
        for(int j = 0; j < 0; j++)
            break label1;
```

is valid. This allows for user-located loop control mechanisms. It pretty much serves as a restricted goto statement. C does not have this feature.

Problem Set

9.

The major advantage of only being able to use boolean expressions is that it'll much more difficult to make a mistake. At compile time Java will tell you whether or not you there is a boolean expression inside of the if clause. Take the following code for example:

```
int x = 5;
if(x = 5)
    doSomething();
```

Let's pretend you really meant for the statement `x == 5` to be in the if clause. In Java this will produce a compilation error letting you know that you have an error. In C, however, this will evaluate to true and execute the if statement.

The downside to this is that it limits your flexibility. For example, say you have a variable `students` that holds the number of students in a class. In C, it is possible to say `if(students){statements()}`. If you're desired outcome was to do something only if the number of students were greater than 0, then this would achieve that. This is because in C, non 0 values will act as true, and 0 as false.

12.

I can imagine a situation like the following. Say for example you're checking to see if a variable is in a list. You could iterate through the list using a for statement, and if the statement is found, you break. In the else clause, you could raise an error stating that the value wasn't found. This is because the else is always executed unless there is a break.