Q: How could we improve this code?

```java
System.out.print("Enter a positive number (or any negative number to quit): ");
int num = Integer.parseInt(br.readLine());
int sum = 0, count = 0;
while (num >= 0) {
    count++;
    sum = sum + num;
    System.out.print("Enter a positive number (negative number to quit): ");
    num = Integer.parseInt(br.readLine());
}
double average = ((double)sum)/count;
System.out.println("The average of the numbers is "+average);
```

Count-controlled loops, revisited

- So far, we’ve used while loops and do-while loops for both count-controlled and sentinel-controlled loops
- It would be nice if there were an easier way to implement count-controlled loops than this:

```java
int count = 0, sum = 0;
while (count < 10) {
    sum = sum + count;
    count++;
}
```

for loop

- Used for count-controlled loops
- The counter is handled by the loop itself, and is part of the loop condition
- A control variable keeps track of how many times the loop has executed
- “For these values of i, do the following...”
### Examples

```java
int sum = 0;
for (int i=0; i<10; i++) {
    sum = sum + i;
}
for (int j=100; j>=0; j--) {
    System.out.print(j + " ");
}
int k;
for (k=1; k<200; k*2) {
    System.out.print(k + " ");
}
```

### Format

```
for (<initialization>; <boolean expression>; <increment>) {
    <statement>
}
```

- `<initialization>`: initializes the control variable
- `<boolean expression>`: some test of the control variable
  - loop ending condition
- `<increment>`: change the value of the control variable by this amount each time through

### Notes

- control variable can be declared inside or outside of `<initialization>`
  - for (int i=0; ...)
- Loop will execute as long as `<boolean expression>` is true
- `<increment>` can be any change to the variable
  - adding 1, adding 3, subtracting 1, subtracting 10, multiplying by 4, dividing by 7...

### Nested for loops

- Just as we nested `if` statements, we can nest `for` statements
- (we can nest `while` and `do-while` too)
- Example: looping through a two-dimensional structure
  - table
  - matrix
Example: Table

- Table contains student grades
- Grader made a mistake on everyone's grade on every assignment
- Need to raise everyone's grade by 10%
- Q: how to do this?

Grade change: pseudocode

for each row in the table,  
for each item in the row,  
multiply the item by 1.10;

Grade change: code

```java
public class Grades {
    int numStudents;
    int numHW;
    double[][] gradeList;
    public Grades() {
        this(0,0);
    }
    public Grades(int students, int hw) {
        this(students, hw);
        numStudents = students;
        numHW = hw;
        // create an array; rows correspond to students,
        // columns correspond to assignments
        gradeList = new double[numStudents][numHW];
    }
    public void populateGradeList() {
        // fill the table with the students' grades
    }
    public void changeGrades(int percent) {
        double factor = percent/100.0 + 1.0;
        for (int i=0; i<numStudents; i++) {
            for (int j=0; j<numHW; j++) {
                gradeList[i][j] = gradeList[i][j] * factor;
            }
        }
    }
}
```
Loop summary

- Use **while** loops when you want to test the condition before executing the loop (pre-test)
- Use **do-while** loops if you want the loop to execute at least once (post-test)
- Use **for** loops if you want to execute the loop a set number of times (count-controlled)
- Watch out for infinite loops!

Odds and ends

- Generating random numbers
  - Midterm evaluation

Random numbers in Java

- We did this in `Die.java` in the `roll()` method
- Two ways to generate random numbers
  - using `Math.random()` method
  - using `java.util.Random` class
- This class: we'll use `Math.random()`

Example: Drawing a card

- Need to generate two random numbers:
  - one for the suit
  - one for the value
- Problem: `Math.random()` returns a number between 0 and 1!
  - we want a number between 1 and 4 for the suit, and 1 and 13 for the value
- Solution: “scale up” and then round down
  - \( num = \text{floor}(x \times (\text{max} - \text{min} + 1)) + \text{min} \), where \( x \) is the randomly-generated number
public class CardGame {
...

public void drawCard() {
    int min = 1;
    int maxSuit = 4;
    int maxValue = 13;
    int suit = (int) (Math.floor(Math.random()*maxSuit) +
                min);
    int value = (int) (Math.floor(Math.random()*maxValue) +
                min);
    System.out.print("Card is ");
    switch(value) {
        case 1: // ace
            System.out.print("ace of ");
            break;
        // code continued on next page
        case 11:  // jack
            System.out.print("Jack of ");
            break;
        case 12:  // queen
            System.out.print("Queen of ");
            break;
        case 13:  // king
            System.out.print("King of ");
            break;
        default:  // number
            System.out.print(value+" of ");
    }
    switch (suit) {
        case 1:  // clubs
            System.out.println("clubs");
            break;
        // code continued on next page
        case 2:  // spades
            System.out.println("spades");
            break;
        case 3:  // hearts
            System.out.println("hearts");
            break;
        case 4:  // diamonds
            System.out.println("diamonds");
            break;
        default:  // something weird happened
            System.out.println("Error in generating suit");
    }
}

Notes

- Numbers generated are pseudorandom
  - not really random
  - numbers will eventually repeat if we generate enough of them
- Control the numbers that are generated using a seed (starting randomizer)
  - Math.random() does not allow us to set a seed, while java.util.Random does
- Generating “good” random numbers is an important problem in many areas of CS
  - computer simulation, cryptography, etc.
Midterm evaluation

- What 1-2 things are going well for your learning in the course so far (please be as specific as you can)?
- What 1-2 things are not going well for your learning?
- What can I, the teacher, do differently, and what can you, the students, do differently to improve the second half of this course?
- How often do you attend the prefector sessions? Are the times adequate or inadequate? (explain)
- How much do you ask the lab assistants for help on assignments?  
  {all the time, semi-regularly, once in a while, never}
- Other comments?