Ninja Coding Exercises

1. **Ternary Operators**
   The following code snippet is boring, clean, and clear. Anyone at your entire company can follow the flow of this code and can work with it. Obviously, this cannot stand! Please ninjify the following code. It should, at minimum, involve a thrice nested ternary operator and poor variable names:

```
public static String getStatus(int age, boolean registered, String gender) {
    if (age > 60) {
        if (registered) {
            if (gender.equals("male")) {
                return "Senior Registered Male";
            } else {
                return "Senior Registered Female";
            }
        } else {
            if (gender.equals("male")) {
                return "Senior Unregistered Male";
            } else {
                return "Senior Unregistered Female";
            }
        }
    } else {
        if (registered) {
            if (gender.equals("male")) {
                return "Junior Registered Male";
            } else {
                return "Junior Registered Female";
            }
        } else {
            if (gender.equals("male")) {
                return "Junior Unregistered Male";
            } else {
                return "Junior Unregistered Female";
            }
        }
    }
}
```

```
public static String getStatus(int a, boolean r, String g) {
        return a > 60 ?
                r ?
                        g.equals("male") ? "Senior Registered Male" : "Senior Registered Female"
                :
                        g.equals("male") ? "Senior Unregistered Male" : "Senior Unregistered Female"
        :
                r ?
                        g.equals("male") ? "Junior Registered Male" : "Junior Registered Female"
                :
                        g.equals("male") ? "Junior Unregistered Male" : "Junior Unregistered Female";
}
```

2. **Single Letter Variables**
   You are provided with a Java snippet containing a series of functions with clear, self-explanatory variable names. Your job is to replace all the variable names with single or double-letter variable names, ensuring the logic remains unchanged. The functionality should remain the same, but the readability should suffer greatly! Here's your starting point:

```java
public double calculateDiscountedPrice(double originalPrice, double discountPercentage) {
    double discountAmount = originalPrice * (discountPercentage / 100);
    return subtractDiscount(originalPrice, discountAmount);
}

public double subtractDiscount(double priceBeforeDiscount, double discountValue) {
    return priceBeforeDiscount - discountValue;
}

// elsewhere in the code base, this is called with:
double finalPrice = calculateDiscountedPrice(100.0, 15.0);
```

Requirements:

1. Replace all variable names with single-letter variable names without changing the core logic.
2. Ensure the logic remains the same.

```java
public double a(double b, double c) {
        double d = b * (c / 100);
        return e(b, d);
}

public double e(double f, double g) {
        return f - g;
}
double h = a(100.0, 15.0);
```

3. Full Ninja Challenge
   You get the idea! Here is your final challenge: create a clean bit of code that does something straightforward, and then apply as many bad practices as you can think of to make it fairly incomprehensible. Then, next class, we'll try to solve each other's challenges.

   Please note that if you take this too far, it stops being fun for everyone. The most fun code of this form is solvable, but *just*. Going past the line is frustrating. (Of course, all of this stuff happens in real code, and a lot of that code is absolutely no fun.)

I will create function that calculates the average of an array of integers in Java, and returns a double of their average. I will then make this function hard to understand and unnecessarily complicated.

Normal version:

```
public double calculateAverage(int[] numbers) {
        int sum = 0;
        for (int number : numbers) {
                sum += number;
        }
return (double) sum / numbers.length;
}
```

Ninja/Difficult Version:

```
public double c(int[] n) {int s = 0, l = n.length; for (int i = 0; i < l; i++) s += n[i%l];return (double) s / l;}
```