

Activity 4-1: Refactoring Review

For each problem below, read through the given code and explain what refactoring should be done to improve the code.

```
1. public class FindNumberProperties {  
    private int number;  
  
    public FindNumberProperties(int n) {  
        number = n;  
    }  
  
    public void getProperties() {  
        if ((number % 2) == 0)  
            System.out.println(number + "_is_even");  
        else  
            System.out.println(number + "_is_odd");  
  
        if (number < 0)  
            System.out.println(number + "_is_negative");  
        else if (number > 1)  
            System.out.println(number + "_is_positive");  
        else  
            System.out.println(number + "_is_neither_positive_or_negative");  
  
        if (number > 12 && number < 20)  
            System.out.println(number + "_is_a_teen_number");  
    }  
}
```

```

2. public class MiniCalculator {

    private String[] parseTokens(String s) {
        return s.trim().split("\\s+");
    }

    public double calculate(String strToParse) {
        String[] tokens = parseTokens(strToParse);
        double n1 = Double.parseDouble(tokens[0]);
        String op = tokens[1];
        double n2 = Double.parseDouble(tokens[2]);

        if (op.equals("+"))
            return add(n1, n2);
        else if (op.equals("-"))
            return subtract(n1, n2);
        else if (op.equals("*"))
            return multiply(n1, n2);
        else
            return divide(n1, n2);
    }

    private double add(double n1, double n2) {
        return n1 + n2;
    }

    private double subtract(double n1, double n2) {
        return n1 - n2;
    }

    private double multiply(double n1, double n2) {
        return n1 * n2;
    }

    private double divide(double n1, double n2) {
        return n1 / n2;
    }
}

```