Refactoring Practices

1. Consider these smells:
    a. Comments
    b. Large Class
    c. Long Method
    d. Long Parameter List

    For each refactoring in the following list, write the letter for the smell(s) it might help cure:

    ____ *Duplicate Observed Data*
    ____ *Extract Class*
    ____ *Extract Interface*
    ____ *Extract Method*
    ____ *Extract Subclass*
    ____ *Introduce Assertion*
    ____ *Introduce Parameter Object*
    ____ *Preserve Whole Object*
    ____ *Rename Method*
    ____ *Replace Parameter with Method*

2. In the following code,
    a. What duplication do you see?
    b. What would you do to remove the duplication? (You don't have to actually write code here, but be as specific as you can about the actions you would take.)

```
try {
    String template = new String(sourceTemplate);

    // Substitute for %CODE%
    int templateSplitBegin = template.indexOf("%CODE%");
    int templateSplitEnd = templateSplitBegin + 6;
    String templatePartOne = new String(
   template.substring(0, templateSplitBegin));
    String templatePartTwo = new String(
   template.substring(templateSplitEnd, template.length()));
    code = new String(reqId);
    template = new String(templatePartOne + code +
templatePartTwo);

    // Substitute for %ALTCODE%
    templateSplitBegin = template.indexOf("%ALTCODE%");
    templateSplitEnd = templateSplitBegin + 9;
    templatePartOne = new String(
        template.substring(0, templateSplitBegin));
    templatePartTwo = new String(
        template.substring(templateSplitEnd,
template.length()));
    altcode = code.substring(0,5) + "-" + code.substring(5,8);
    out.print(templatePartOne + altcode + templatePartTwo);
```

```
} catch (Exception e) {
     System.out.println("Error in substitute()");
}
```

3. Consider this code fragment:

```
if (!((score > 700) ||
     ((income >= 40000) && (income <= 100000)
  && authorized && (score > 500)) ||
    (income > 100000)))
      reject();
else
      accept();
```

   a. What code smells do you see in this code? Give examples of each smell you list.
   b. Apply DeMorgan's Law to simplify this as much as possible.
   c. Starting from the original, rewrite the condition by introducing explaining variables.
   d. Starting from the original, flip the if and else clauses, then break the original into several if clauses. (You'll call accept() in three different places.)
   e. Consolidate Conditional Expression by extracting a method to compute the condition.
   f. Which approach was the simplest? The clearest? Can you combine the techniques? Explain your answers.

4. a Refactor the following method:

```
double getPayAmount() {
        double result;
        if (_isDead) result = deadAmount();
        else {
                if (_isSeparated) result = separatedAmount();
                else {
                        if (_isRetired) result = retiredAmount();
                        else result = normalPayAmount();
                };
        }
        return result;
};
```

   b. What refactoring(s) did you use? Why did you choose the one(s) you did?