# Homework 4, ECE 590

Code smells are certain structures in the code that indicate a violation of fundamental design principles and negatively impact software quality, corresponding to a deeper problem in the software system. Please refer to the references below for more information.

In the assignment, you need to identify at least four different kinds of code smells in one or more pieces of code. You can use your own codes, however, you are highly recommended to use long codes (> 100 lines) to find smells. If you do not have such a project, you can use the example codes uploaded on the Canvas/Files/Code Samples (codes are mainly in *src/main/java/edu/duke/xz295/ticTacToe* and the python code *Calculator-Example-1.py*)

Guidelines:
1. Provide the codes you are working on.
2. Identify at least four different kinds of code smells. Highlight the lines which have the smells and explain why they have code smells.
3. Explain in your own words how the code smells negatively impact your system. The explanation should be specific and detailed.
4. Relate at least three develop/design principles to avoiding code smells. Give a short explanation in your own words about how it improves software reliability. You are recommended to use the design principles learnt in class, but you can also introduce other design principles as long as you explain them well.
E.g. There is a design principle “DRY”: Don’t Repeat Yourself. It aims to avoid the code smell “Duplicate Code”. Duplicate codes make maintenance more difficult because if the developer wants to make one change, (s)he needs to change in several places. Faults easily arise if the developer forgets to change in one place. Therefore, avoiding the code smell “Duplicate Code” by the “DRY” principle improves software reliability.
References you may find helpful:


1. https://en.wikipedia.org/wiki/Code_smell
2. https://refactoring.guru/refactoring/smells
3. https://apiumhub.com/tech-blog-barcelona/code-smells