lewis university

**course:** Object Oriented Design

Effect of code bad smells on Modularity

**Team Members:**

Bhavyothkalik Reddy Vontair

Saikiran Sabavat

GitHub:

https://github.com/bhavyoth99/OODASSIGNMENT2.git

Effect of code bad smells on Modularity

Section1

GQM approach

Goal: In this assignment 2, our aim is to study the "Effect of code bad smells on Modularity" Question: Does the code smells really have an impact on the modularity of the software project? Metrics: For the purpose of evaluation, I would like to use the DIT, NOC, CBO metrics from ck metrics

Section2

I have downloaded the following subject programs from the git hub and analyzing the data sets from my GQM point of view

|  |  |
| --- | --- |
| SNO | ProjectName |
| 1 | DesignPatternsSimplifiedMaster |
| 2 | DesignPatternsJava |
| 3 | GreenhouseControlSystem |
| 4 | Drawing-Figures-using-Design-Patterns-master |
| 5 | Simple\_Java\_AdventureGame-master |
| 6 | Virtual-Coffee-Shop-master |
| 7 | HospitalManagementSystem-master |
| 8 | Computer-Store-DP-main |
| 9 | MultiThreading-master |
| 10 | Huston |

Section 3

Here we are using two tools

1. CK metrics as used in assignment1

I have downloaded this from git hub. This tool majorly provides 6 metrics: WMC (Weighted Methods per Class)

DIT (Depth of Inheritance Tree) NOC (Number of Children)

CBO (Coupling Between Objects) RFC (Response for Class)

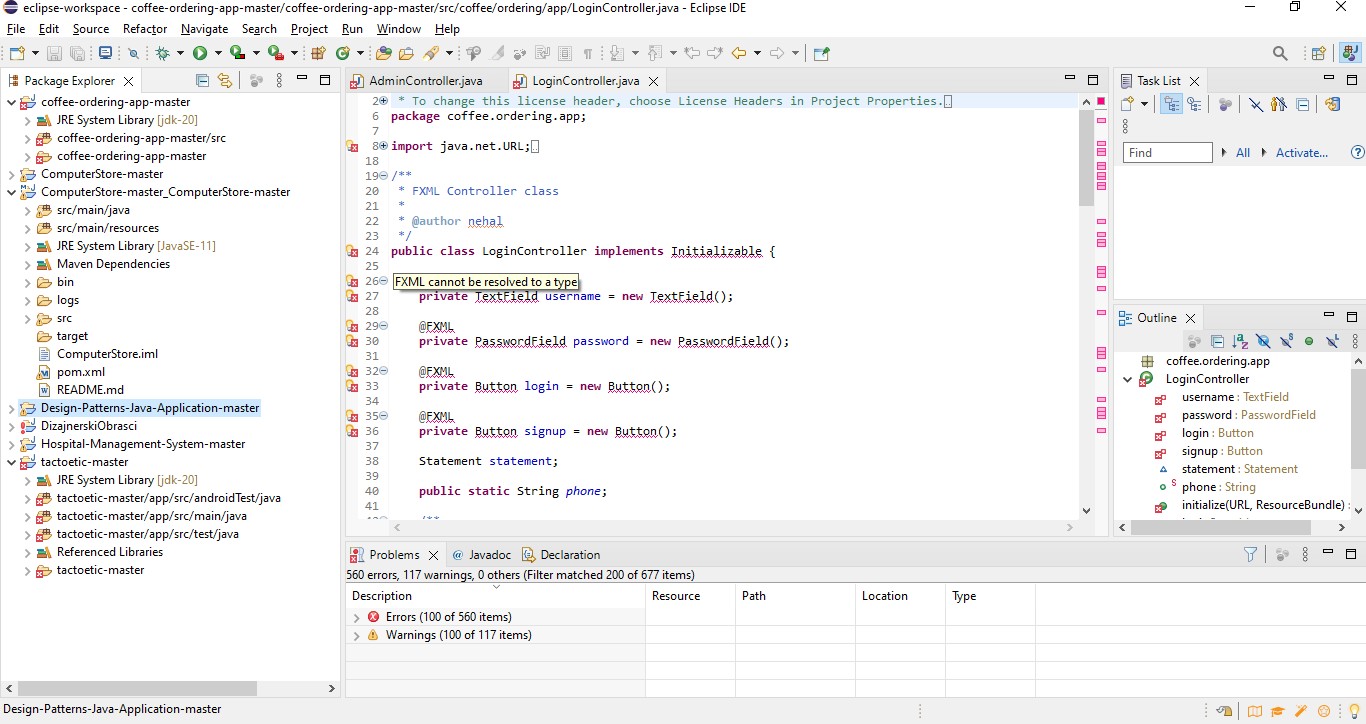
LCOM (Lack of Cohesion of Methods)

But here for my study I will be using the below 3 metrics:

DIT, NOC, CBO

For the analyses of testability, modularity

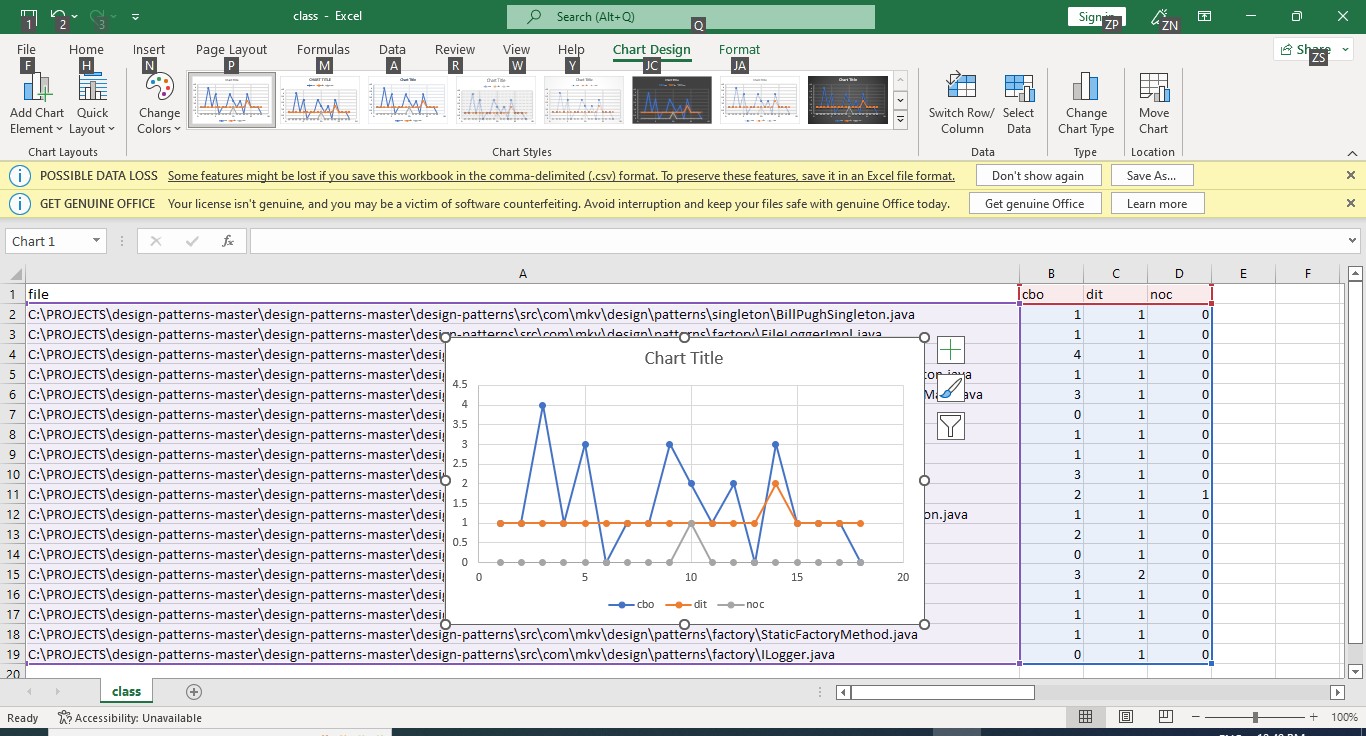
1. The other tool for detecting code bad smells is “check style”. I have opened the 10 projects in my eclipse. I have installed check style latest version from eclipse market place and activated it on the subject programs so that it gives details about the code smells in all the class, projects.



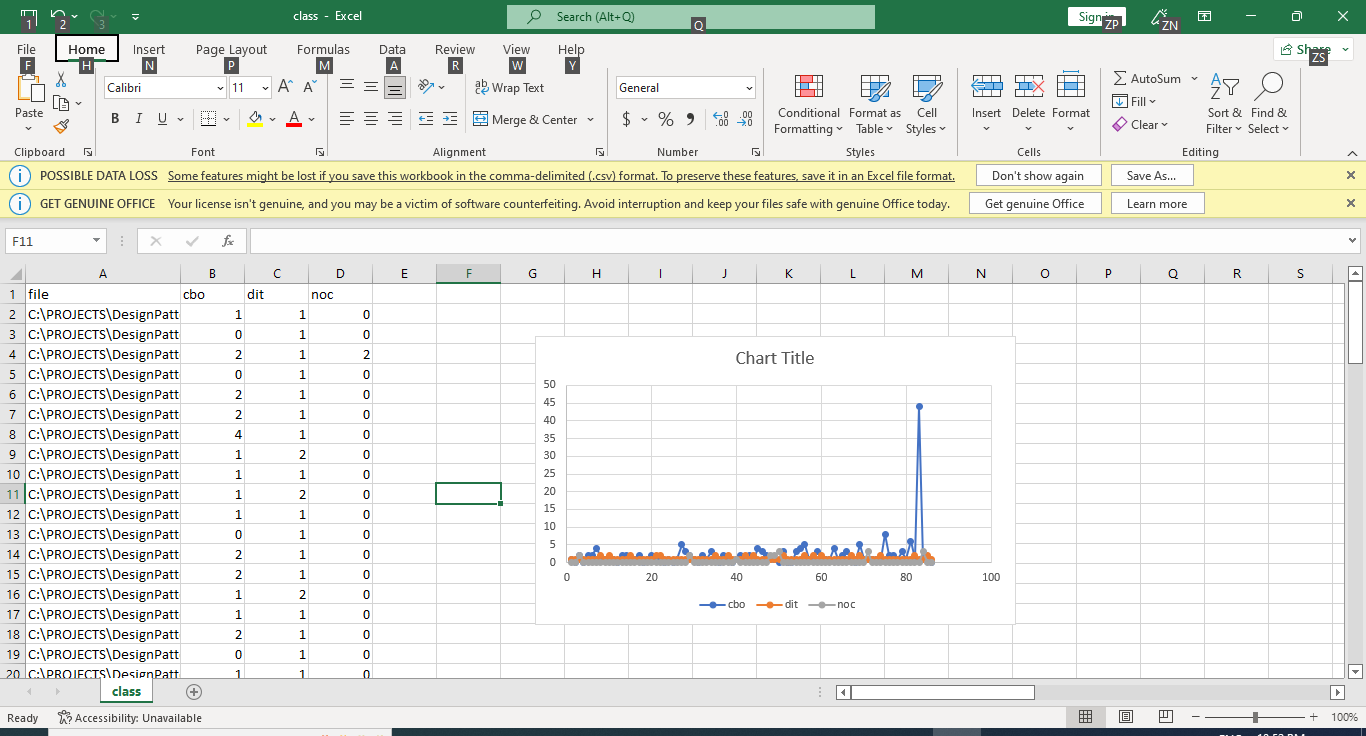
I have analyzed the ck metrics for the same projects to study the modularity.

Section 4

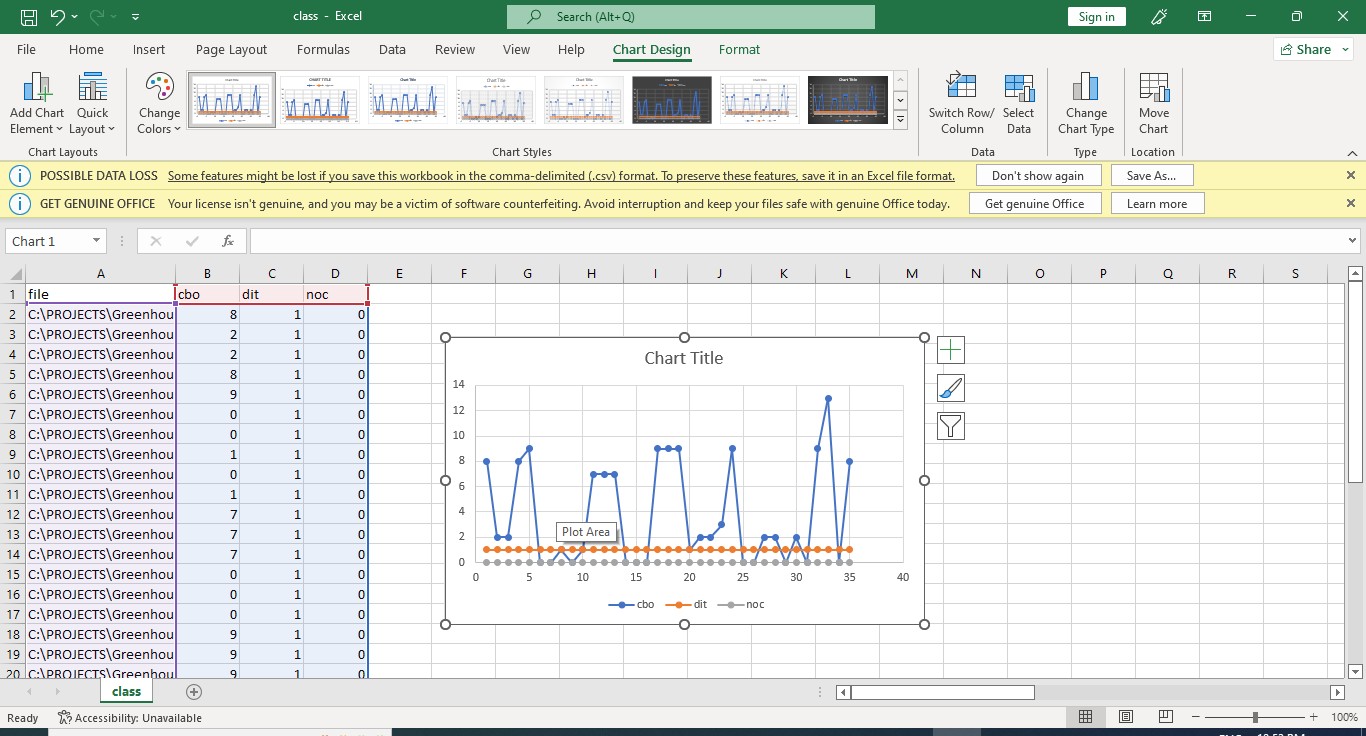
Project1:



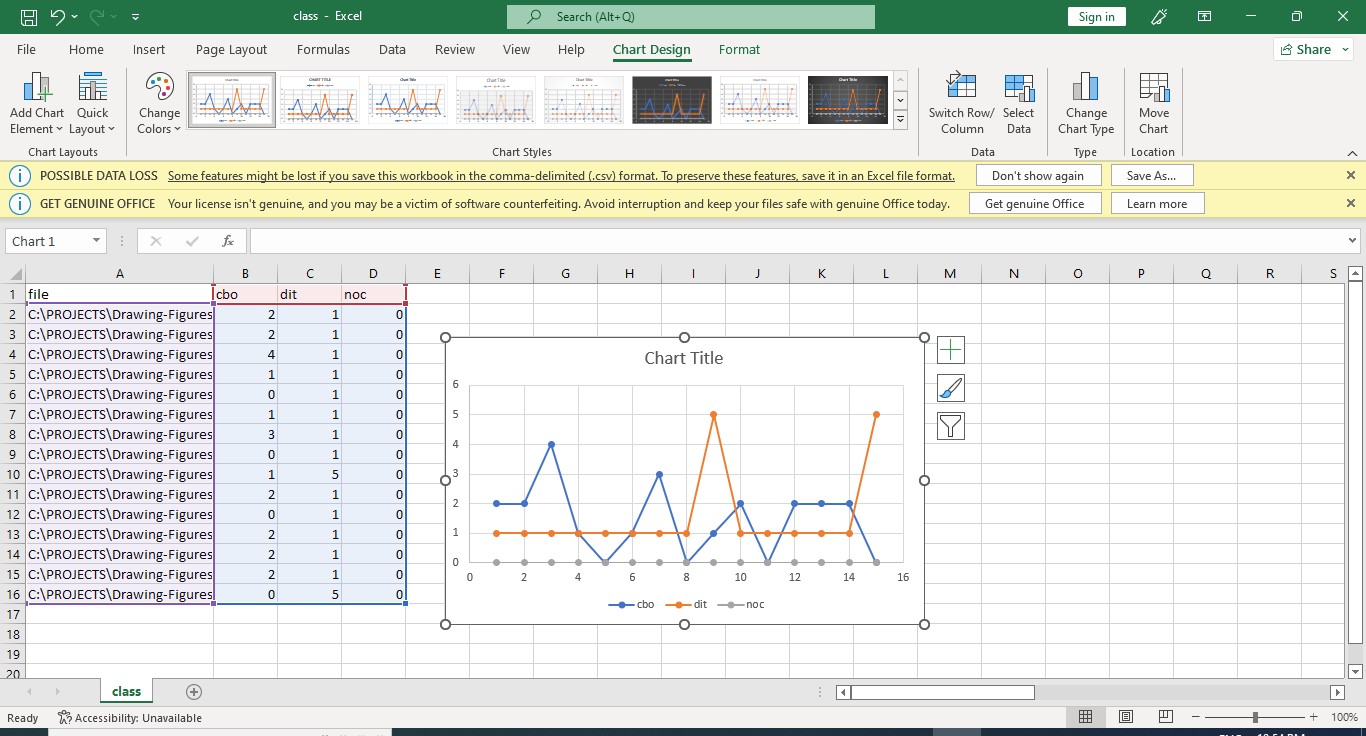
Project2:



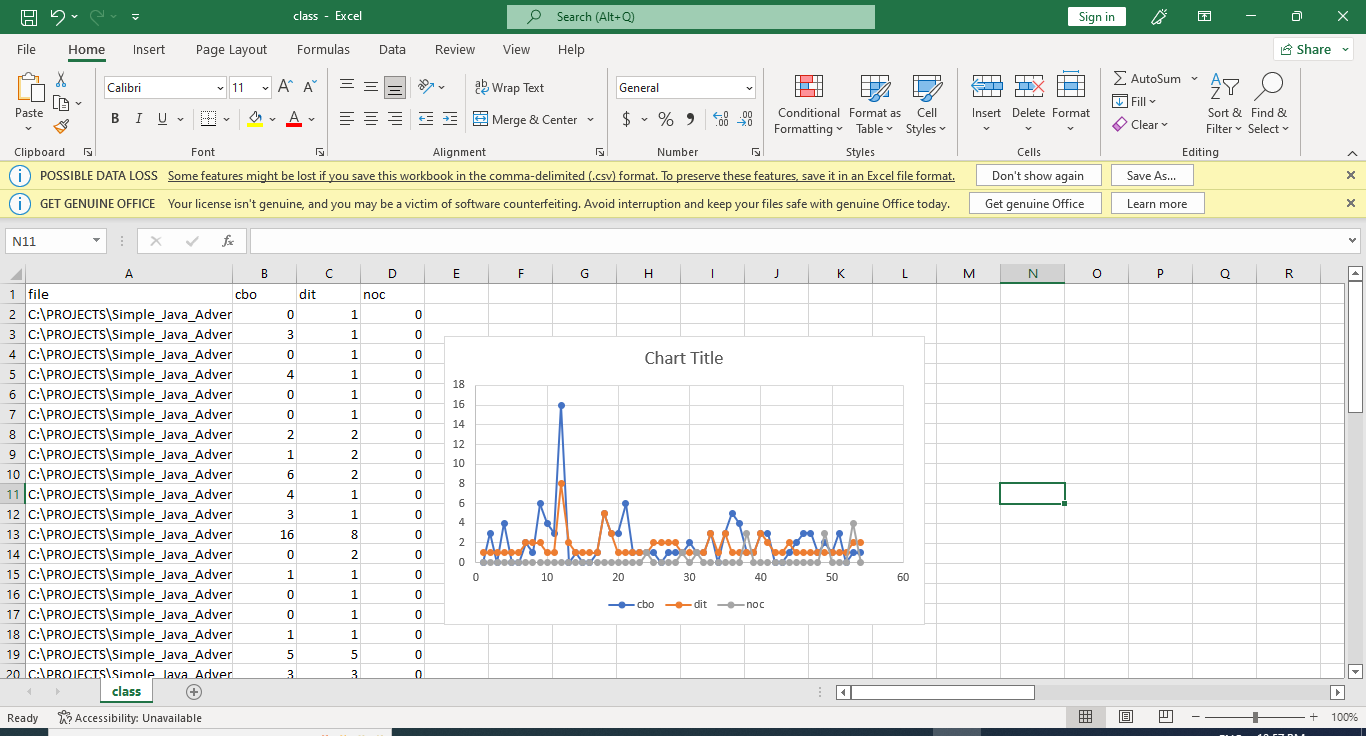
Project3:



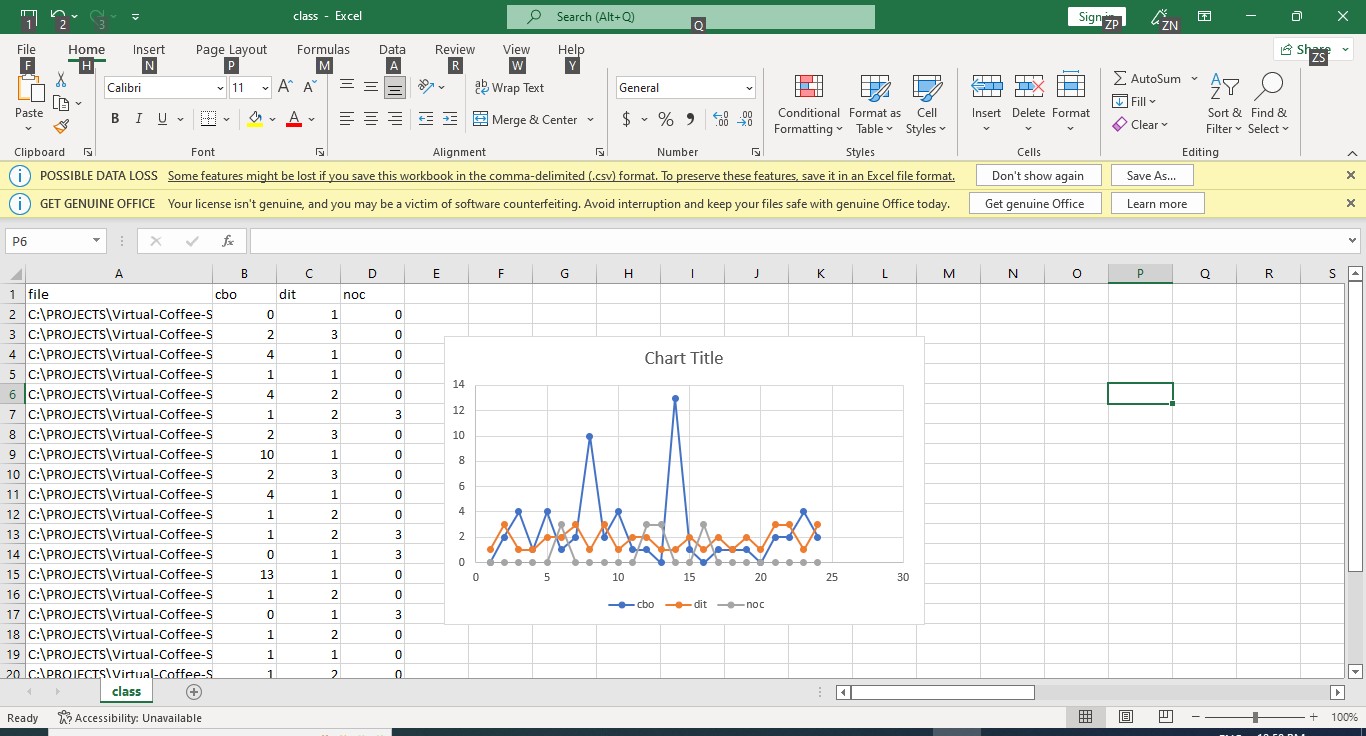
Project4:



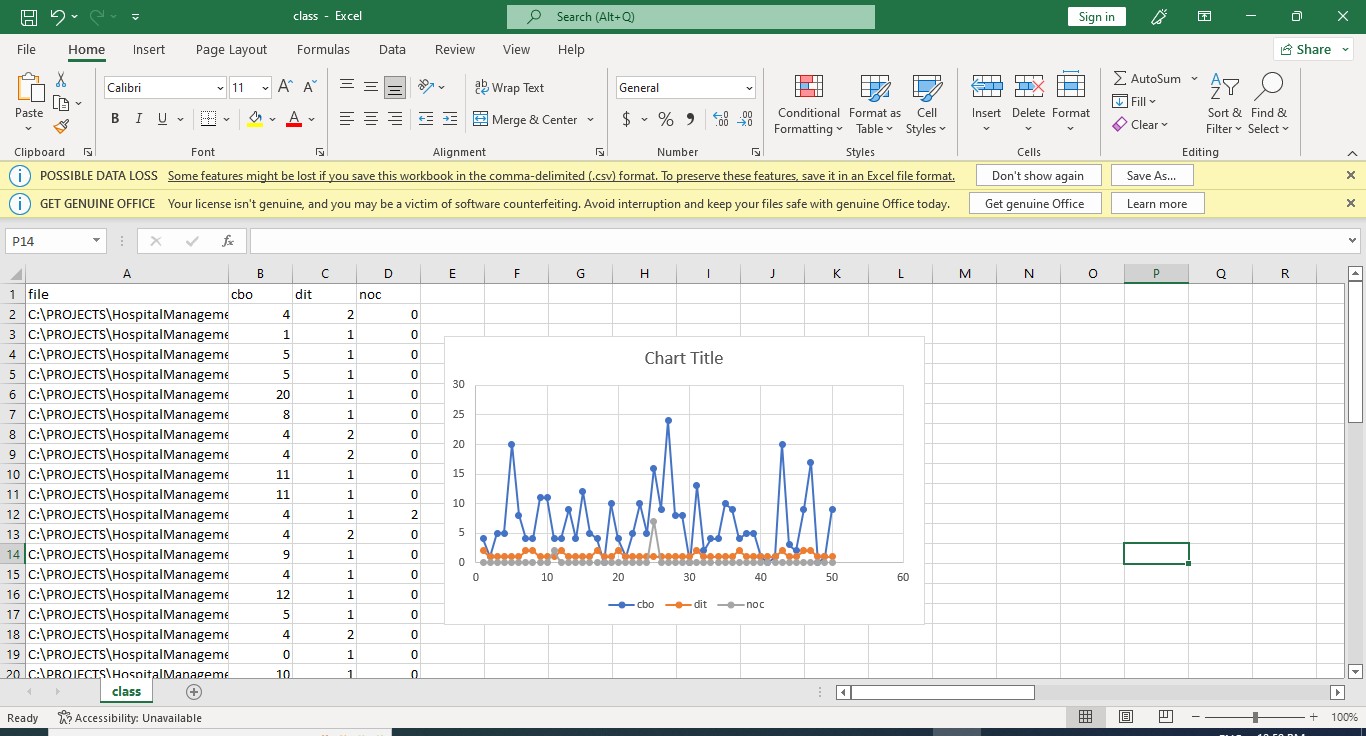
Project5:



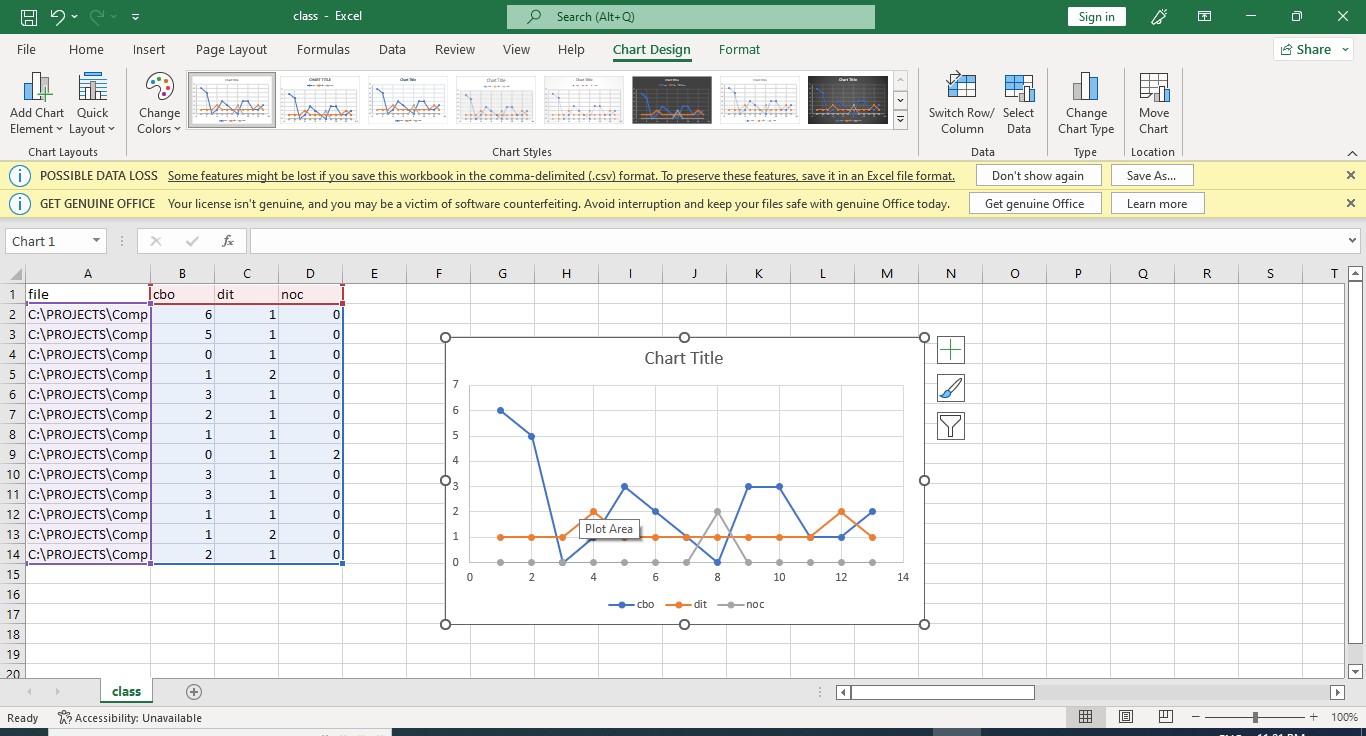
Project6:



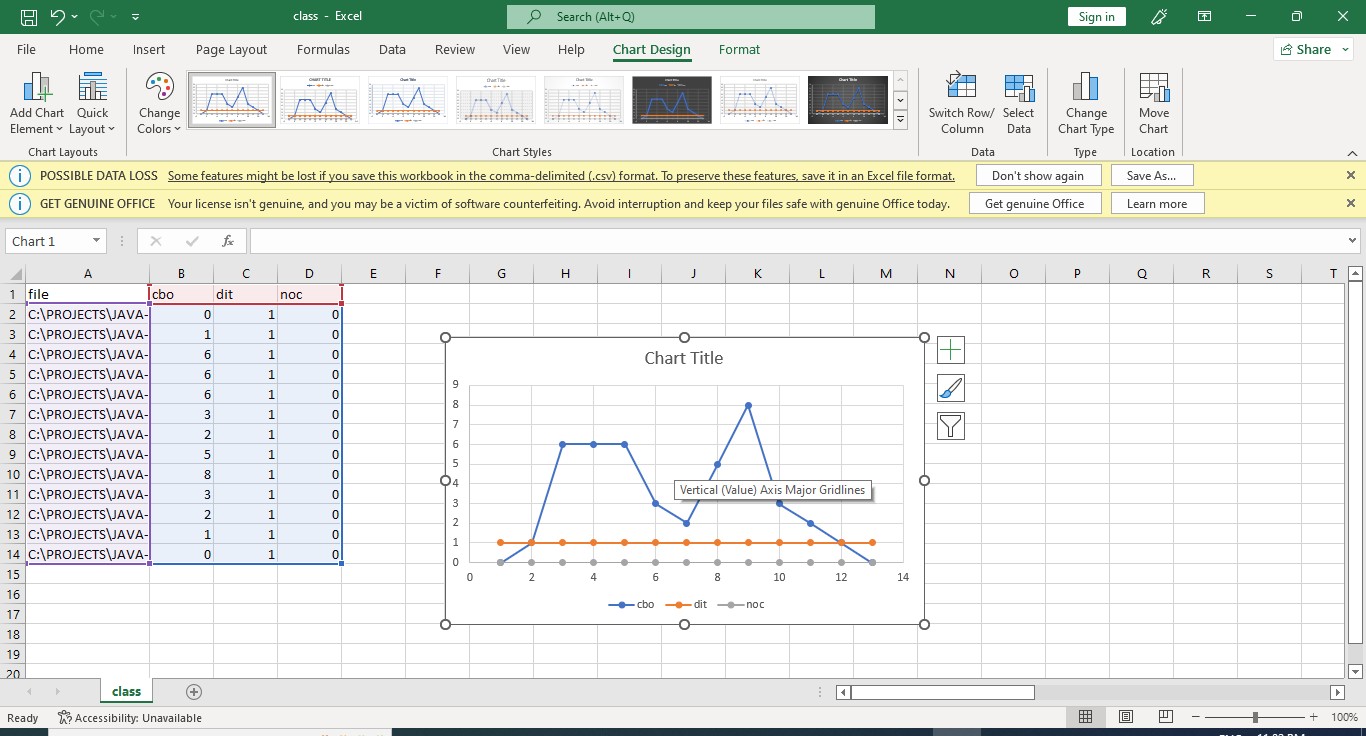
Project7:



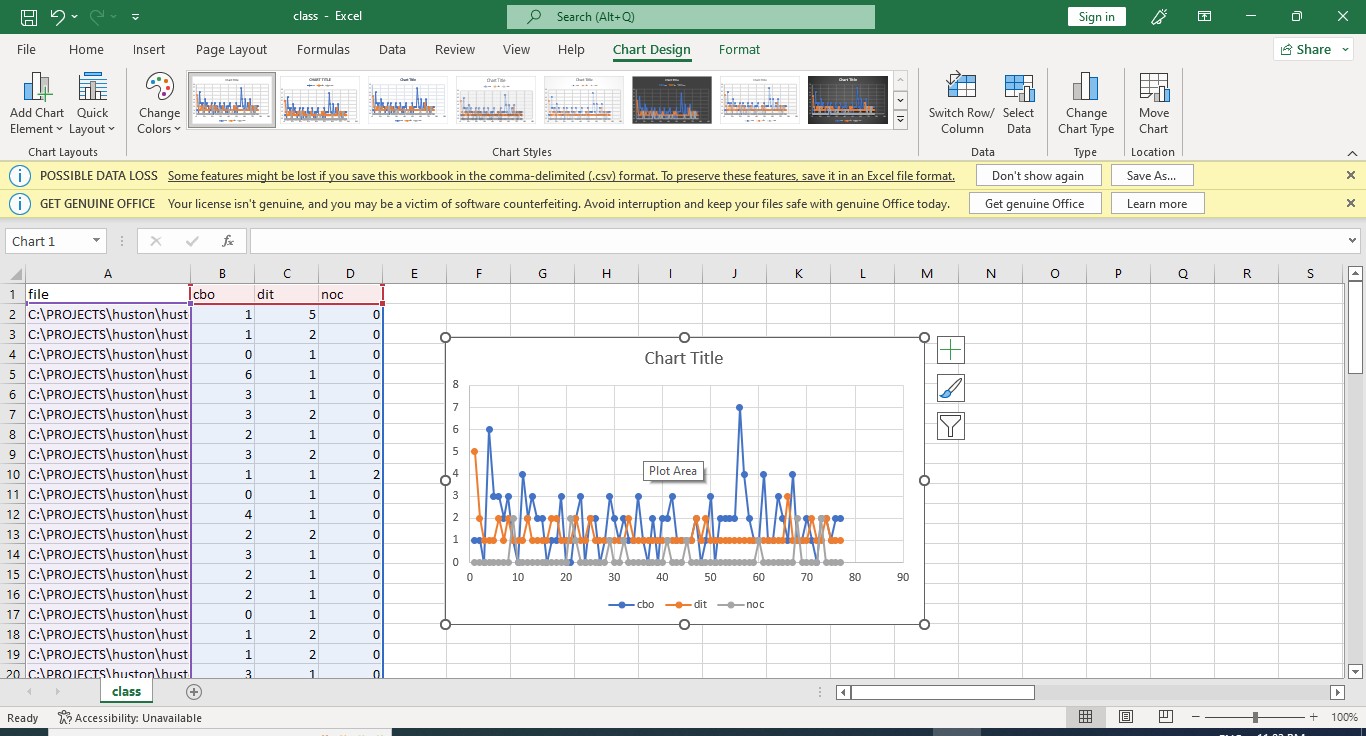
Project 8:



Project 9:



Project 10:



Section 5

Conclusion:

After studying the subject programs with the help of ‘check style’ and based on the ck metric values we can say that the bad code smells have a harmful effect on the modularity of the projects. And this in turn has the bad effect on all the software quality attributes. So, it is better to refactor the code at the earliest with the appropriate refactoring techniques. And after refactoring it is always advisable to test the code once more.

References:

https://github.com/mauricioaniche/ck

https://[www.youtube.com/watch?v=VVukBwl48KA](http://www.youtube.com/watch?v=VVukBwl48KA)