**LEWIS UNIVERSITY**

**Course: OBJECTED ORIENTED DEVELOPMENT**

**Professor: Fadi Wedyan**

**Assignment-1**

**Empirical Study: Effect of Class size on software maintainability**

**Group members:**

**1: Bhavyothkalik Reddy Vontair (L30080552)**

**2. Sabavat Saikiran (L30079454)**

**GitHub URL: https://github.com/bhavyoth99/assignment-1-**

***Section1***

**GQM**

In the context of Object-Oriented Design (OOD), the Goal-Question-Metrics (GQM) is a framework used for defining and measuring software quality. It provides a structured approach to help guide the design process and assess the effectiveness of the design decisions. The GQM framework was originally proposed by Victor Basili and is widely used in software engineering.

Here's how the GQM framework works in the context of OOD:

Goal: The first step is to define the overall goals of the software design. These goals can be high-level objectives such as improving maintainability, enhancing reusability, or increasing performance. The goals should be specific, measurable, achievable, relevant, and time-bound (SMART goals).

Questions: Once the goals are established, a set of questions is formulated to assess the achievement of those goals. These questions should be specific and measurable, focusing on the aspects of the design that are relevant to the goals. For example, if the goal is to improve maintainability, questions could include "How many lines of code were reduced?" or "What is the cyclomatic complexity of the design?"

Metrics: Metrics are the measurements used to answer the questions and assess the design quality. They provide objective data that can be collected and analyzed to evaluate the effectiveness of the design decisions. Metrics can be quantitative, such as lines of code, cyclomatic complexity, coupling, cohesion, or qualitative, such as code readability or modularity

Goal: **In this Assignment1**, the goal is to study the effect of class size on software maintainability.

Question: Does really the class size has an impact on the software project maintainability?

Metric: Coming to the metrics, I will be using the mentioned CK metrics. Especially among all the ck metrics, I will be using LOC (Lines of code), WMC (Weighted Methods per Class)

***Section 2***

**Data Sets:**

I have downloaded 5 java projects from the 500 java project list provided which are 10k in size, at least 3 years old from git hub. Then I have unzipped the projects and executed the ck metrics tool jar using the command prompt. This develops output in the form of 4 files class, field, method, variable. But we will read the metrics values from the class file and develop a line chart so as to analyze the goal.

|  |  |  |
| --- | --- | --- |
| S.NO | Average WMC value | Average LOC value |
| Project1 | 2.83 | 10.55 |
| Project2 | 1.88 | 10.14 |
| Project3 | 2.86 | 16.33 |
| Project4 | 5.24 | 21.98 |
| Project5 | 3.12 | 13.37 |

Section3

For this assignment, I am using CK-Code metric tool.

This was introduced by Chidamber and Kemerer, best know as (C&K) metrics suite. It is designed to:

* Measure aspects of OO software
* Measure the complexity of the design
* Improve software development

Below are the major C&K 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)
* LOC (Lines of Code)

But here I would be considering WMC, LOC for the empirical study.

Section4

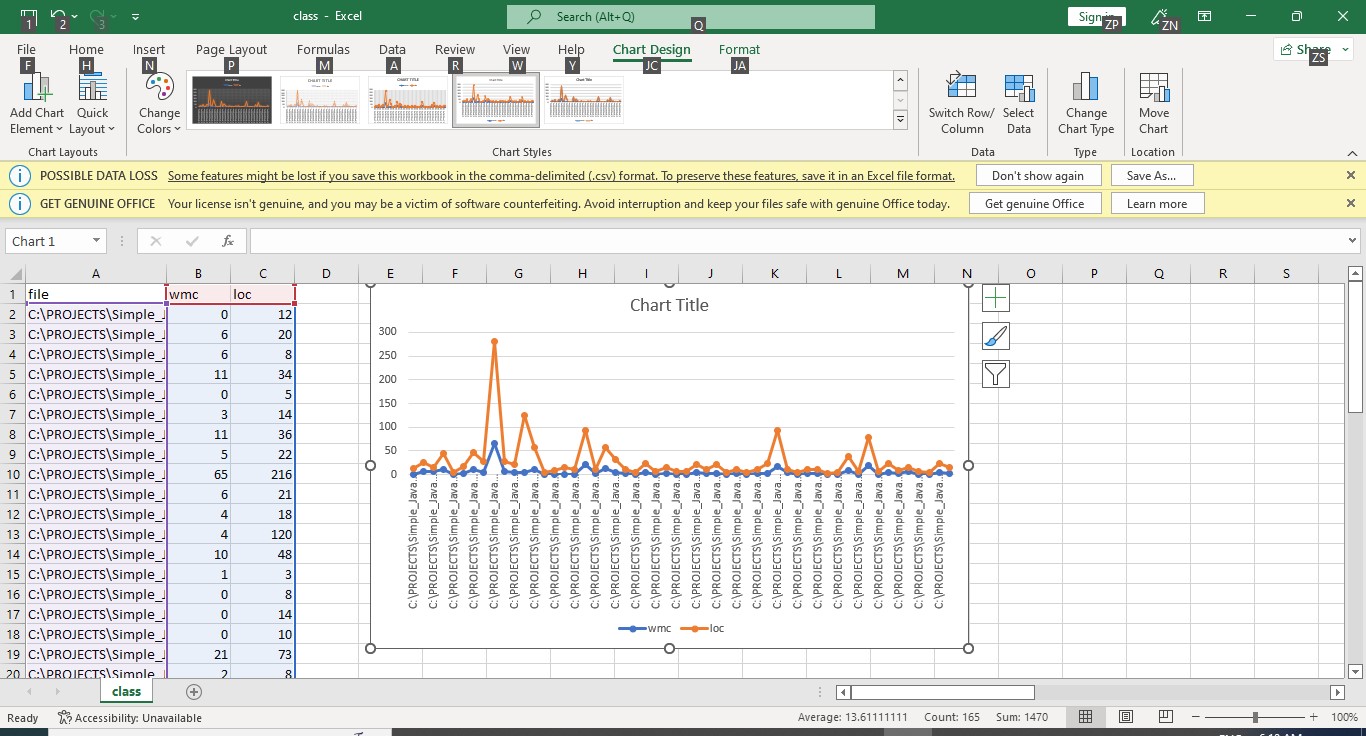
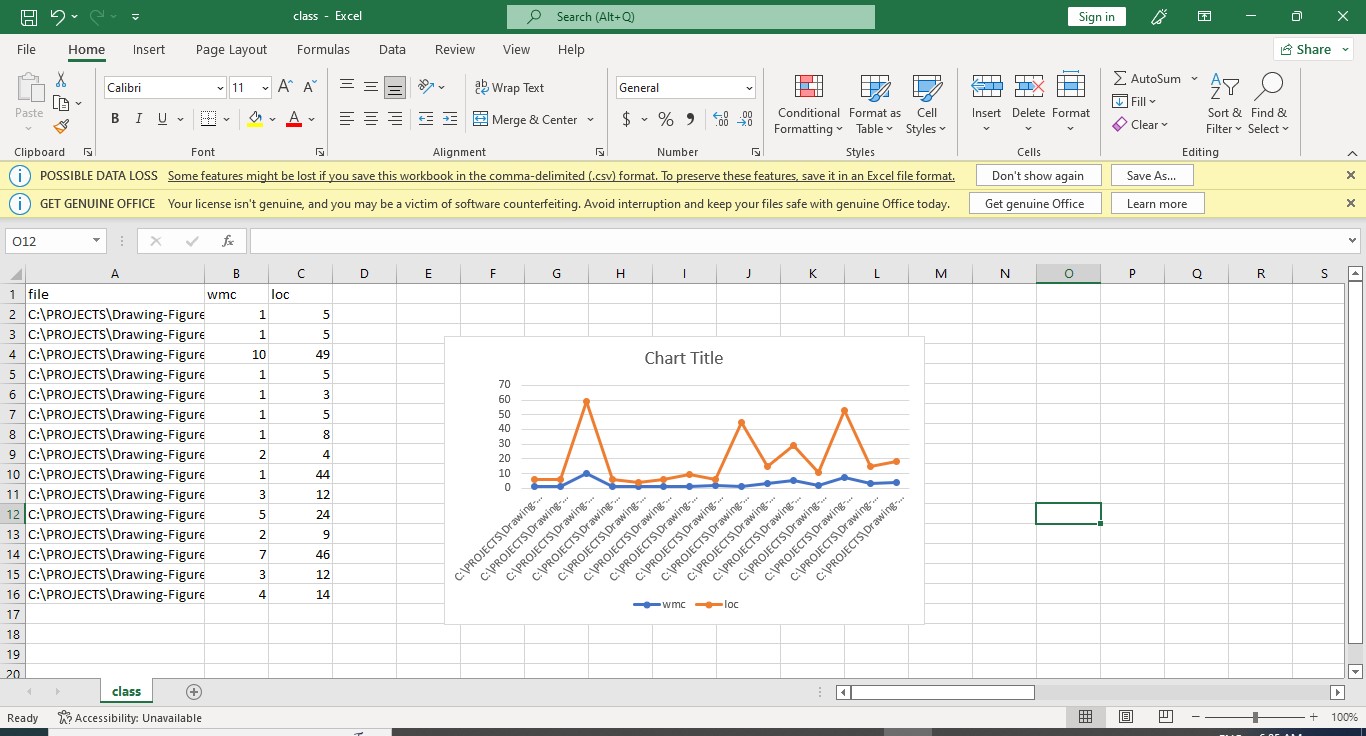
Graphs

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence



A screenshot of a computer

Description automatically generated with medium confidence

Section5

Conclusion:

Considering the ck metric WMC, LOC values I am drawing the conclusions for the question mentioned in the section1. Yes, by conducting the empirical study I can say that the class size has an effect on the software maintainability. For better maintainability of the software project the WMC, LOC values should be low. From the average values of WMC, LOC mentioned in the table (Section2) and the graphs from (Section4) we can conclude that the values of the 2 metrics are not so high and the software project maintainability is high.

References:

Projects: [www.github.com](http://www.github.com/)

CK meric suite tool[: https://github.com/mauricioaniche/ck.](https://github.com/mauricioaniche/ck)