ISLAMIC UNIVERSITY OF TECHNOLOGY

SWE 4302

LAB 1 GROUP B

September 10, 2024

PREPARED BY

Maliha Noushin Raida

Lecturer, Department of CSE

Contents

1 Introduction **3**

2 Today's Task **4**

1 INTRODUCTION

When we were new to programming we wrote simple code. Although we may not have appreciated it at the time, this was a great strength. Since then, we've learned new skills, tackled harder problems, and produced increasingly complex solutions. In the theory class, we have learned that most code will someday change. As systems grow, complexity increases, and we might need to introduce abstractions (like classes or methods) to handle this. However, the complexity of the abstractions should not hinder any developer's understanding. That's why we have a concept of clean code.

Clean code is code that is simple, readable, and easy to understand, both for the original developer and others who may work on it later. It is well-structured, with clear and descriptive naming, minimal complexity, and no unnecessary duplication. Clean code is also maintainable, allowing for easy updates and changes, and testable, ensuring it can be reliably checked for correctness.

By priority, we will be focusing on two attributes,

- **Easy to read**
- Easy to change

Note that easy to change is hard to define because we do not know in advance what changes are coming. Therefore, it is prescribed that we do not focus on "easy to change" too early.

As we know, writing clean code is a lot like painting a picture. Most of us know when a picture is painted well or badly. But being able to recognize good art from bad does not mean that we know how to paint. So too, being able to recognize clean code from dirty code does not mean that we know how to write clean code.

In today's lab, we will be testing out your ability to write understandable, clean code. The objective of today's lab,

1. Reviewing the basics of OOC-I
2. Program considering just readability and understandablity

Note, you are allowed to use any language you are comfortable in. Please design your own architectural solution. It does not matter whether you can solve the problem completely; it matters how you attempt to solve it.

2 TODAY'S TASK

Book list Tracker Requirements

Alice Johnson is a book lover and a content creator who manages a blog where she reviews books and engages with readers. She reads dozens of books every year and keeps track of them through an online book catalog. Recently, Alice has realized that she is having difficulty managing and reviewing her reading patterns, which slows her down when writing her blog posts. She wants a better system to analyze her reading habits automatically.

You had a meeting with Alice Johnson to gather requirements. Because Alice is pretty tech-savvy, she tells you that the book list tracker just needs to read a text file containing a list of books read by her. She exports the file from her "Goodreads" online profile. The file contains data about the **title, author, genre, number of pages, and the date she finished reading each book**. This text is structured using a comma-separated values (CSV) format. Here is a sample of the exported file:

```
The Great Gatsby,F. Scott Fitzgerald,Classic,180,2023-01-05
Atomic Habits,James Clear,Self-Help,320,2023-02-14
The Hobbit,J.R.R. Tolkien,Fantasy,310,2023-03-01
Sapiens,Yuval Noah Harari,Non-Fiction,498,2023-04-10
Pride and Prejudice,Jane Austen,Classic,279,2023-04-25
Dune,Frank Herbert,Sci-Fi,412,2023-05-18
Educated,Tara Westover,Memoir,334,2023-06-12
```

He would like to get an answer to the following queries:

- What is the total number of books she has read in the last year?
- How many books did she read each month?
- What are her top 10 longest books by page count?
- Which genre does she read the most?
- What is the average length (in pages) of the books she reads?