

ASSIGNMENT INSTRUCTIONS

1. Assignment 02: **100 points w/ 0 E.C. points**
2. Due Date & Time: **06-21-2024 at 11:55 PM**

WHAT TO SUBMIT

1. Assignment Report
2. Code

PERFORMANCE TRACKER		
ASMT	GRADE	YOUR GRADE
CANVAS	05	
01	15	
02	100	
TOTAL	120	

HOW TO SUBMIT AND THE RULES TO FOLLOW

- The Guidelines for All Assignments
- The Course Policy on Student Conduct and Academic Honesty
- The assignment instructions for this assignment
- The additional instructions provided in class and on Canvas.
- Submit via Canvas, the Assignment Submission section.

A: 90-100% B: 80-89% C: 70-79% D: 60-69% F: 0-60%
The course grader provides feedback to your assignments on Canvas.

ABOUT

CSC 220 is among CSC 340's course prerequisites. CSC 210 and CSC 220 learn Java. Java is used in CSC 413 and other upper-division courses. A good foundation in Java usually helps us pick up C++ faster. For these reasons, we were supposed to have a prerequisite exam to demonstrate our comprehensive knowledge of Java programming.

Instead of a test, this assignment is our opportunity to review (and to learn) important Java topics and get us ready for C++. This assignment is to be done in JAVA, not in C++.

Please start this assignment as soon as possible. We want to make sure that we understand the problems and that we solve the right problems.

PART A – OOP Class Design Guidelines, 15 points

Y. Daniel Liang's 8 Class Design Guidelines: <http://csc340.ducta.net/WEEK-01/JAVAToCPP-ClassDesignGuidelines.pdf>

Please choose 5 guidelines and discuss them in depth. For each guideline, use at least **one page** for your discussion. It is OK to use code to help demonstrate your points. The code portion, if any, should not take up more than 1/3 of each guideline's discussion.

PART B – Java Programming, Data Structures, and Data Design, 85 points

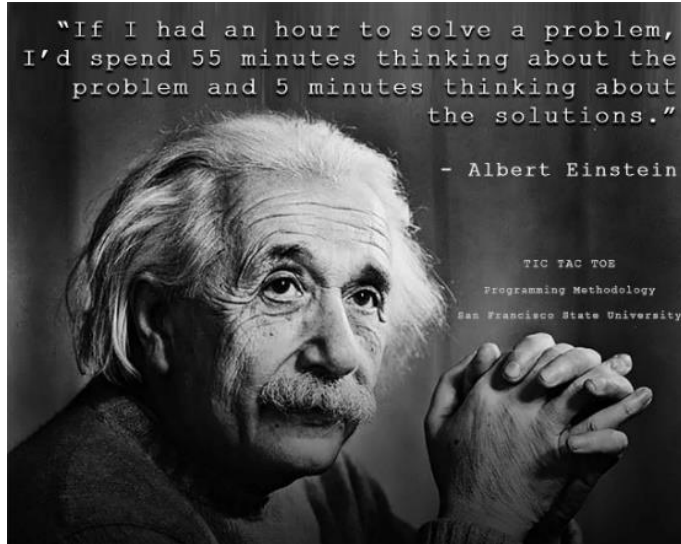
We are hired to implement an interactive dictionary. Our dictionary takes input from users and uses the input as a search key to look up values associated with the key. Requirements:

- **Coding:** No hard coding, https://en.wikipedia.org/wiki/Hard_coding. *Please think about Dynamic and Scalable.*
- **Data Source:** Store the original data in a set of **enum** objects. Each keyword, each part of speech, and each definition must be stored in a separate data field. Do not combine them such as storing three parts in one String.
- **Data Structure:** Use existing data structures or create new data structures to store our dictionary's data.
- **Data Loading:** When our program starts, it loads all the original data from the Data Source into our dictionary's data structure. Data Loading must finish before our program starts interacting with users.
- **User Interface:** A program interface allows users to input search keys. This interface then displays returned results. Our program searches the dictionary's data (not the Data Source) for values associated with the search keys.

- **Identical Output:** Our program's output must be identical to the complete sample run's output. The **complete sample output** is posted at <http://csc340.ducta.net/Assignments/Assignment-02/Assignment-02-PartB-CompleteSampleOutput.pdf>.

1. **Program Analysis to Program Design, 15 points.** *Please think about Interviews.*

In at least 1 full page, please explain the following in detail:



- Your analysis of the provided information and the provided complete sample output. *Please think about Clients and Sales.*
- What problem are you solving? Please explain it clearly then define it concisely. *Please think about Problem Solving and Interviews.*
- How do you store data in enum objects? And why? *Please think about Data Structures and Data Design.*
- Which data structures do you use/create for your dictionary? And why? *Please think about Data Structures and Data Design.*

2. **Program Implementation, 70 points.** *Please think about Interviews.*

- Implement your program to meet all the requirements.
- In your assignment report, demonstrate your program to your grader/client.
- Does your program work properly?
- How will you improve your program?

Happy problem-solving and happy coding!