

Course: CSC340.04

Student: Mesoma Esonwune

Teammate: Rocio Camacho

Assignment Number: 01

Assignment Due Date & Time: 09/18/2018 at 11:59pm

Assignment 01

- Part A – OOP Class Design Guidelines

- 1. Please choose 4 guidelines and discuss them in depth. For each guideline, use at least half a page for your discussion.**

- 1) Encapsulation

- a. A class should use the private modifier to hide its data from direct access by clients. This makes the class easy to maintain.
- b. Provide a getter method only if you want the data field to be readable, and provide a setter method only if you want the data field to be updateable.
 - a. I feel like I don't follow the second part all the time. Even in this project, I think I created some setters when I actually never needed it. I just don't remember if I removed it or if I kept it there on accident. I know that I'm always confused on when I need to use private. It's something that was quickly mentioned but wasn't really enforced in class when submitting assignments. Now, it's something that I have to consciously think about or else I keep everything default or public.
 - b. I guess it's useful because it helps hide instance variables in a class from an illegal direct access and makes flexible code (super easy to maintain in the long run).

- 2) Consistency

- a. Follow standard Java programming style and naming conventions. Choose informative names for classes, data fields, and methods. A popular style is to place the data declaration before the constructor and place constructors before methods.
- b. Make the names consistent. It is not a good practice to choose different names for similar operations.
- c. In general, you should consistently provide a public no-arg constructor for constructing a default instance. If a class does not

Course: CSC340.04

Student: Mesoma Esonwune

Teammate: Rocio Camacho

Assignment Number: 01

Assignment Due Date & Time: 09/18/2018 at 11:59pm

support a no-arg constructor, document the reason. If no constructors are defined explicitly, a public default no-arg constructor with an empty body is assumed.

- d. If you want to prevent users from creating an object for a class, you can declare a private constructor in the class, as is the case for the Math class.
 - a. I definitely do this. I always want to make sure that if I look back at this code in a month that I know what I did and why each class, method, and variable was created. But I definitely don't do a public no-arg constructor often. I should work on that more.

3) Cohesion

- a. A class should describe a single entity, and all the class operations should logically fit together to support a coherent purpose.
- b. A single entity with many responsibilities can be broken into several classes to separate the responsibilities.
 - a. I don't do this as often as I should, OR I just don't know if I'm doing it correctly. I know that everything in the class should relate to each other and fit logically but I sometimes can't make the connection. Or I don't know if I need to make a new class or separate a class because they don't belong together. But it's all about the relationship between the methods and data of a class. They should have an agreement on what their purpose is.

4) Instance vs. Static

- a. A variable or method that is dependent on a specific instance of the class must be an instance variable or method. A variable that is shared by all the instances of a class should be declared static.
- b. Always reference static variables and methods from a class name (rather than a reference variable) to improve readability and avoid errors.

Course: CSC340.04

Student: Mesoma Esonwune

Teammate: Rocio Camacho

Assignment Number: 01

Assignment Due Date & Time: 09/18/2018 at 11:59pm

- c. Do not pass a parameter from a constructor to initialize a static data field. It is better to use a setter method to change the static data field.
 - d. Instance and static are integral parts of object-oriented programming. A data field or method is either instance or static. Do not mistakenly overlook static data fields or methods. It is a common design error to define an instance method should have been static.
 - e. A constructor is always instance, because it is used to create a specific instance. A static variable or method can be invoked from an instance method, but an instance variable or method cannot be invoked from a static method.
 - a. This is something that I've been confused sometimes. The word instance is used very often and I get confused. But the general idea is that static is used when the method/variable needs to belong to the class and shouldn't have different values in different objects (but same object types). If it's an instance variable, it needs to describe that particular object.
- Part B – Java Programming, Data Design, and Data Structures
 - I used IntelliJ to code my project.
 - 1. Program Analysis to Program Design
 - 1) In 1 full page, please explain the following in detail: Your analysis of the provided information and the provided sample output, what problem you are solving, how you store data in enum objects. and why, and which data structure(s) you use/create for your dictionary. And why.**
 - a. First of all, this was very hard to start because we weren't given any idea on how to start this. Most of us who have been at SFSU has always been given a skeleton to work with but here we had nothing to go off of. It just made it so much harder than I

Course: CSC340.04

Student: Mesoma Esonwune

Teammate: Rocio Camacho

Assignment Number: 01

Assignment Due Date & Time: 09/18/2018 at 11:59pm

expected. Also, I didn't like how we had to use enums. That made me so confused and wasted a lot of my time.

- b. The problem that we are solving is creating a dictionary that will print out entries or a pre-made error based on what the user entered. But the data must come from enum objects and be stored in a data structure so that we don't give access to the original data.
- c. I stored the enum objects by having the name of the enum and then in parentheses having 3 Strings: one for the word, one for the part of speech, and the last one for the definition.
- d. I used ArrayLists for my dictionary. At first, I wanted to use HashMaps but I couldn't get over the fact that I couldn't have one key and multiple values or same named keys. It just got very difficult for me to use. Then, as I was talking with my friends about the different kinds of data structures I can use, we both realized that using an ArrayList to store the info would be amazing but I would need to make a class that stores the entry into an object and can be printed with the format necessary. So I created a separate Words class that kept it working properly.

2. Program Implementation

1) Does your program work properly? How will you improve your program?

- a. My program does work properly and I added some other touches to make some cases have a proper error instead of just printing nothing.
- b. I would want to use Google Guava or see how I could use HashMap. My code is really long right now and I feel like it could probably be shortened but I just don't have the skills right now to figure out how.