**CS2030 PE#1 Tips: OOP**

**P.S. These are just my takes, so please take it with a pinch of salt!**

1. **First 20 minutes**

* **1st step:** Read through the questions to have an overview of what the entire program is about and what you need to do
  + **Identify all the nouns and verbs** in the **main problem statement** as you are reading!
    - This includes nouns which describe places like farms or a “container” like menu
  + Try to identify the **general nouns** also, e.g. animal, food, etc.
  + Also see how different levels are related to each other
* **2nd step:** Design a general skeleton for the entire program
  + **Identify ALL the possible classes, abstract classes, and interfaces that you need**, and what are the things that they know (fields) and what are their behaviors (methods)
    - **Do this by basing on the nouns** that you have identify when reading the main problem statement
      * Note that **if there are nouns that describes a place or a “container**”, then you should consider making them as a class because they can simplify your driver class a lot
    - You don’t have to think about how you gonna implement the methods first, but you must at least **identify all the fields, methods that are required for the class, abstract class & interface**
      * i.e. think what a class knows (fields), and what it can do (methods)
  + Try to encapsulate as many things as possible **to make your driver class clean.**
    - However, only encapsulate the necessary stuffs.
  + **Design the overall structure of all the classes well**
    - i.e. inheritance, implementation etc.

1. **The next 1 hour and 10 minutes**

* Code!
  + Make use of polymorphism if needed, e.g. **method overloading and overriding!**
  + Code out the ADT first, then modify the driver class after you have prepared everything
    - i.e. Code from bottom to up>: code the ADT that does not depend on any other thing first
  + When coding out a class and interface, leave out the access modifier **for the class,** like public, interface, etc. unless you are asked to put them in a package
  + When you code, code out the things one by one, e.g. let’s say if in this level a certain functionality is not needed, then don’t code it out first.
* Don’t make silly mistakes so that you don’t have to waste time on debugging!
* Remember to save your file as you move on from one level to another!
* Save all your work before the 90 minutes mark!

**Some tips**

1. **Methods**

* Use this kind of generic, e.g. ArrayList<? extends Item> **as the type of the method parameter** when you want your method to work for all the collection which consist of the different subclasses of the extended class.
* E.g. in printing
* E.g. if I want my method to work for ArrayList<Burger>, ArrayList<Drink>, then I would use a method in whichI use ArrayList<? extends Item> as the type of my mehod parameter.

# Remember, it’s worth your time to design your skeleton well so that you don’t have to refactorize your entire code when you hit the high level. Coding out your solution won’t take a lot of time, as long as you design your skeleton well.

**Practical 2 (18.19 Sem 2)**

1. In this practical, **declarative programming** is tested!!! This can be done using stream, although not explicitly stated in the question.