Lab 11 -- Scheduling (CANCELLED)

The goal of this lab is to design and implement a strategy design pattern.

Scheduling

*Scheduling* is the task of determining the ordering of valuable resources given a limited resource. It may refer to the ordering in an assembly line fulfills order for customers, to the order in which patients are seen in an emergency room (usually determined by the order they arrived and the severity of their injuries), to the order in which a CPU executes processes, or even to the order in which you perform your homework assignments for the multiple classes you are taking this semester.

The best way to schedule such tasks can be determined by a number of different algorithms. As an example, let's consider the tasks that you need to complete in order to be successful in your classes. Each assignment has a description, a day that it was assigned, a due date. You might be interested in scheduling these assignments in several different orderings:

1. in the order that they were **assigned** defined by the day it is added to the task list
2. in **alphabetical** (lexicographical) order
3. in the order that they are due (by their **deadline**)
4. in order by their **difficulty** where assignments where the harder an assignment is, the more time that is given to you

You can imagine that you might want to know what to work on first based on any of these. It turns out this is perfect for the *strategy design pattern*.

What to do

**Package:** assignments

Start by downloading this [provided assignments.zip file](https://northeastern.instructure.com/courses/63372/files/7319865/preview)[download](https://northeastern.instructure.com/courses/63372/files/7319865/download?download_frd=1)that contains two classes. Take a moment to familiarize yourself with the classes by reading through their documentation You will be using these classes as the starting point for this assignment.

In this lab, we will be adding the ability to schedule assignments based on the strategies that are listed above. Your solution meeting the following requirements:

* Create and implement an interface SchedulingStrategy that defines a schedule method that takes a list of assignments and returns a string with the name of the strategy that was used.
* Implement each of the strategies named AssignedSchedulingStrategy, AlphabeticalSchedulingStrategy, DeadlineSchedulingStrategy, and DifficultySchedulingStrategy.
* Add a method scheduleAssignments in AssignmentList that takes a SchedulingStrategy and uses it to schedule the items in its task list. It should throw an IllegalArgumentException if no scheduling strategy is provided.
* Assignments that share a characteristic (e.g., their due date) should be scheduled in lexicographical order.
* Scheduling is not dynamic. In other words, if new assignments are added to the assignment list, they are added at the end. In order for them to be scheduled, the user will need to call schedule again.
* Trying to schedule a non-existing list should raise an IllegalArgumentException.

**Hint:** You may find using a [Comparator (Links to an external site.)](https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Comparator.html) useful in this assignment.