ASSIGNMENT 1

In this assignment, you will work with important concepts such as inheritance, abstraction, arrays, code documentations etc. This is an individual assignment and you are not allowed to work or submit the assignment with anyone else.

The **deadline for submitting** this assignment is **Sunday, June 19 at 11:59PM.** The **due date** for demonstrating the assignment to your professor is the week of June 19, whichever day your lab sessions falls into within that week. This assignment **weighs 8%** of the entire course marks**.**

Problem Statement

Most people in Ontario especially small businesses have been adversely affected by the lock-down situation in the province due to COVID-19 for about two years. The Ontario Public Health (OPH) office has the task of creating a strategic plan for the gradually re-opening of schools and business places. The strategic plan by OPH must specify actions that must be undertaken by schools and businesses following a certain routine. The routine approved would indicate actions that must be **regular, occasional** and **rare** on given dates in a year. For example, the regular actions by schools and businesses will be done daily. Occasional actions will be done on a specific day of each month, while actions classified as rarely will be done only on a certain day of the year.

OPH knows the reputation of ICT-AP department at Algonquin College and has enlisted a team of five students from the department, which includes yourself, to implement a system that will provide details all categories of actions that should be taken by schools and businesses on a specific date so that any user can access and retrieve relevant actions (regular, occasional and rare) to ensure compliance with OPH directives. Users of this system your team is implementing can enter a date of their choice and obtain all actions stipulated for that date as output.

In this assignment, you have been provided with:

- Some code files to help you get started.
- Additional information slides (PDF) to provide insight into your tasks and the requirements
- A sample output to guide your solution
- Grading rubrics to show the possible marks.

You are required to read all documents carefully and completely to understand what to do and how to do it. Certain comments have also been included in the code to guide your work. Missing such comments will make your implementation difficult. Some helpful links have also been uploaded for you.

This assignment has **three parts**. Note that you are required to comprehensively document your work using Javadoc style documentation in every relevant section of your code clearly describing what you have done. All tasks in this assignment needs to be completed and demonstrated in order to get your marks.

HOW DO I SUBMIT MY ASSIGNMENT?

1. You must submit your solutions on Brightspace or before the due date.
2. Submit all your solutions through ACTIVITIES >> ASSIGNMENTS >> ASSIGNMENT 1 to Section 300.
3. If your Lab section professor provides an alternative submission information to a designated submission portal, then use that portal which your Professor will create for this assignment.
4. Name your submission using the convention: FirstName_LastName_SectXXX_Assignment1.

WHAT DO I SUBMIT?

You must submit your solutions (diagrams and source code) including the generated Javadoc file on the specified portal on Brightspace.

YOUR TASKS:

1. You are required to **draw a representation of the inheritance hierarchy between the classes** provided in this assignment**.** You do NOT need to show the variables, methods or constructors in the classes. Do not use auto-generating apps that generates UML from code. You can use UML editing tools tools only like draw.io, MS Word, MS VISIO or a similar drawing tool.

2. Inspect the starter-code files provided for you. There are loopholes you are expected to fill.

PART 1

In this part of the assignment, you will work with and submit three code files. You are required to:

1. **Review** the starter-code that has been supplied for you actions named Action.java.
2. This Superclass (Action.java) has a subclass named RegularAction.java.
3. Action has a description and could be carried out on one or more due dates. An example of a description could be *"Wash your hands"* or *"Wear a face mask".*
4. You should write an **abstract method** in the Superclass named occursOn(). This abstract method has parameters *year*, *month* and *day* which checks if the action occurs on that specific date.
5. **Complete** the starter-code provided for you on the RegularAction class (this shows the actions that happened regularly on an everyday basis). ***Hint:*** Observe the sample output provided to you for compliance.
6. Write an ActionDriver class that contains the main method to run your code. In writing the class named ActionDriver.java carefully follow the pattern of the output that has been provided. Instantiate required action objects to test this portion of your code.

PART 2

In this part of the assignment, you are required to:

1. Include two more subclasses of the Superclass namely, OccasionalAction and RareAction. OccasionalAction shows actions that are conducted once each month, while RareAction depicts actions that take place only once on a specified day of the year.
2. **Complete** the starter-code provided for you for the two classes OccasionalAction.java and RareAction.java. Hint: Be guided by observing the sample output provided.
3. Your code must check if the action is carried out on that date. For example, if dealing with an OccasionalAction that happens once a month, your code must ensure that the day of the month matches.
4. Write an ActionDriver2 class that contains the main method to run this second part of your code. In writing the class named ActionDriver2.java carefully follow the pattern of the output that has been provided. Instantiate required action objects that suit both the OccasionalAction and RareAction subclasses.

PART 3

In this part of the assignment, you are required to:

1. Create a new class called AllActionTest.java with a main method to demonstrate your work.
2. In this class, create and fill an array of Action objects with different types of actions permitted by Ontario Public Health for the COVID-19 protocol including actions that should occur everyday, actions that occur once a month, and actions designated to occur only on a certain date in a year.
3. Implement your code (entire classes) such that the users can input a date and retrieve an output of all actions that would happen on that date.

RUBRICS

Item	Neglected (%)	Partially completed (%)	Completed correctly including output (%)	Possible Mark deduction (%)
Submission with compliance to instruction	0	0	1	Incorrect
Javadoc Style documentation of code	0	0.5	2	Incorrect
Diagram representing relationship	0	0	1	Incorrect
Part 1	0	1	4	Code completed but not running and/or incorrect output
Part 2	0	1	5	Code completed but not running and/or incorrect output
Part 3	0	1	3	Code completed but not running and/or incorrect output

Demonstration of this assignment to your professor is compulsory to gain any marks for your work.