

Seneca College

May 21, 2019

Applied Arts & Technology

SCHOOL OF COMPUTER STUDIES

JAC444 Demo & Final Code Due date : May 29, 2019

Workshop 1

Notes:

- i. Each task should be presented during the lab, demo worth 70% of the workshop marks and code uploading worth the other 30%.
- ii. Make sure you have all security and check measures in place, like wrong data types etc., no need to implement Exception as we haven't covered yet. There are other ways to handle bad input data.
- iii. Given output structure is just for student to have a glimpse what the output can look, student are free to make the output better in any way.
- iv. The final should be submitted by the midnight to avoid late penalties which are 10% each day late.

Other inputs can be given during demo, so make sure you test your program properly.

Task 1: Design a class named **MyDate**. The class contains:

- The data fields **year**, **month**, and **day** that represent a date. **month** is 0-based, i.e., **0** is for January.
- A no-arg constructor that creates a **MyDate** object for the current date.
- A constructor that constructs a **MyDate** object with a specified elapsed time since midnight, January 1, 1970, in milliseconds.
- A constructor that constructs a **MyDate** object with the specified year, month, and day.
- Three getter methods for the data fields **year**, **month**, and **day**, respectively.
- A method named **setDate(long elapsedTime)** that sets a new date for the object using the elapsed time.

Write a test program that creates two **MyDate** objects (using **new MyDate()** and **new MyDate(34355555133101L)**) and displays their year, month, and day.

Hint: The first two constructors will extract the year, month, and day from the elapsed time. For example, if the elapsed time is **561555550000** milliseconds, the year is **1987**, the month is **9**, and the day is **18**. You may use the **GregorianCalendar** class from Java API to simplify coding.

Task 2: Design a class named **Person** and its two subclasses named **Student** and **Employee**. Make **Faculty** and **Staff** subclasses of **Employee**. A person has a name, address, phone number, and email address. A student has a class status (freshman, sophomore, junior, or senior). Define the status as a constant. An employee has an office, salary, and date hired. Use the **MyDate** class defined in **Task 1**, to create an object for date hired. A faculty member has office hours and a rank. A staff member has a title. Override the **toString** method in each class to display the class name and the person's name.

Write a test program that creates a **Person**, **Student**, **Employee**, **Faculty**, and **Staff**, and invokes their **toString()** methods.