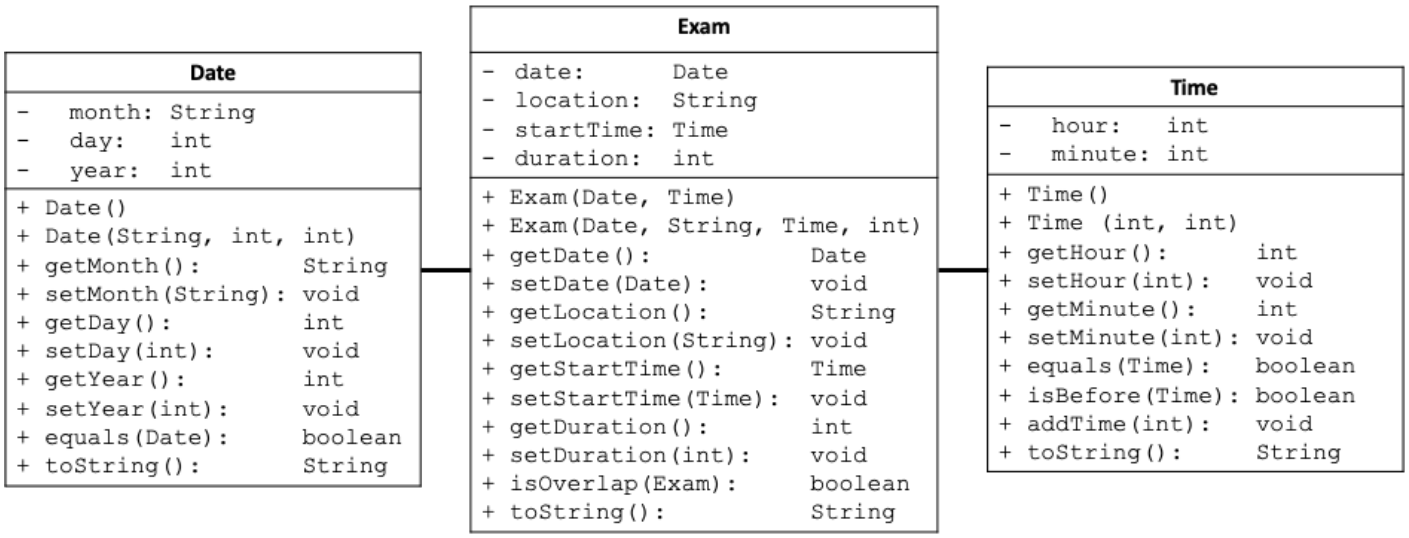
Assignment 2

Objectives

- Introduction to Objects
- More testing

Introduction

This assignment requires you to implement and test the two classes described in the following UML diagram:



NOTE: The `Exam` class has a field that is of type `Date` and `Time`, therefore you should implement the `Date` and `Time` classes first.

See instructions on the following page…

Quick Start

1. Download `A2Tester.java` `Date.java`, `Time.java` and `Exam.java` to the same directory.
2. Read `A2Tester.java` and `Date.java` carefully.
    a. Compile and run the `A2Tester` program from the directory you downloaded the files to.
    To compile: `javac A2Tester.java`
    To run: `java A2Tester`
3. Implement each method in `Date.java`, by repeating the following steps:
    a. Uncomment a test in the corresponding method in `A2Tester.java`
    b. Add stubs for the methods in `Date.java`
    NOTE: a stub is an empty method with only the signature and return value if there is one
    c. Compile and run to ensure your tests and stubs are correct
    d. Implement each method by completing the stub following the documentation within the `.java` file
    e. Compile and run (repeat steps d and e until all of your tests on the Date class pass)
    f. Add additional tests where needed, compile and run
    g. Repeat steps a-f for `Time.java` and Exam`.java`

Submission and Grading

CRITICAL: You **must** name the methods in the `*.java` files as specified in the documentation (shown in the UML above) and as used in `A2Tester.java` or you will receive a **zero grade** for that method.

Submit **ALL** of the following files with your name and student ID at the top of each file using conneX:
`Date.java` **and** `Time.java` **and** `Exam.java`

If you chose not to complete some of the methods required, you **must at least provide a stub for the incomplete method** in order for our tester to compile. If you submit files that do not compile with our tester, you will receive a **zero grade** for the assignment. It is your responsibility to ensure you follow the specification and submit the correct files.

Your code must **not** be written to specifically pass the test cases in the testers, instead, it must work on all valid inputs. We will change the input values, add extra test and we will inspect your code for hard-coded solutions.

Be sure you submit your assignment, not just save a draft. **ALL late** and **incorrect** submissions will be given a **ZERO** grade.

A reminder that it is OK to talk about your assignment with your classmates, and you are encouraged to design solutions together, but each student must implement their own solution. We will use plagiarism detection software on your assignment submissions.