

## CS410J Project 2: Storing An Appointment Book in a Text File (9 points<sup>1</sup>)

In this project you will write classes that store your appointment book (and its appointments) in a text file using a format of your own design.

**Goals:** Learn how to use Java's I/O facilities and throw exceptions.

For this project you will have to implement the following three classes<sup>2</sup> in your `edu.pdx.cs410J.your-login-id` package:

- Class `TextDumper` that implements the `edu.pdx.cs410J.AppointmentBookDumper` interface. A `TextDumper` dumps the contents of an appointment book<sup>3</sup> (including appointments) to a text file.
- Class `TextParser` that implements the `edu.pdx.cs410J.AppointmentBookParser` interface. A `TextParser` reads the contents of a text file and from it creates an appointment book with its associated appointments.
- Class `Project2` that contains a `main` method that optionally reads an `AppointmentBook` from the contents of a text file, creates a new `Appointment` as specified on the command line, adds the `Appointment` to the `AppointmentBook`, and then optionally writes the `AppointmentBook` back to the text file. `Project2` should have the following command line interface:

```
usage: java edu.pdx.cs410J.<login-id>.Project2 [options] <args>
args are (in this order):
  owner           The person whose owns the appt book
  description      A description of the appointment
  beginTime        When the appt begins (24-hour time)
  endTime          When the appt ends (24-hour time)
options are (options may appear in any order):
  -textFile file   Where to read/write the appointment book
  -print           Prints a description of the new appointment
  -README          Prints a README for this project and exits
Dates and times should be in the format: mm/dd/yyyy hh:mm
```

If the text file does not exist, then the program should create an empty `AppointmentBook`. It should **not** issue an error. Dates and times are specified using the same format as in the previous assignment.

---

<sup>1</sup>8 for code, 1 for POA

<sup>2</sup>Make sure to submit **all** of your classes, even if you haven't modified them since the previous project.

<sup>3</sup>Note that the `dump` method dumps an `AbstractAppointmentBook`. You cannot change this fact. This is okay because your `AppointmentBook` **is an** `AbstractAppointmentBook`.

**Error handling:** Your program should exit “gracefully” with a user-friendly error message under all reasonable error conditions. Examples of such conditions include

- Something is missing from the command line or there are extraneous command line arguments
- The format of the day or time is incorrect
- The text file is malformed
- The owner name given on the command line is different than the one found in the text file.