CSCI 3675 – Principles of Programming Languages
Fall 2021
Homework 1 – Inheritance and polymorphism with Python
Due Sunday, September 12, at 11:59 PM

Write a program that implements an appointment book. There are two types of events that can be recorded in an appointment book: a personal appointment, and a meeting. Both types of events consist of a contact (the name and telephone number of a person), and the start time of the event. Events always begin on the hour, and last for one hour. An appointment book is a collection of events. Events can be added to the appointment book, and the list of events for a given date can be displayed. (Note: To complete this assignment you might have to consult The Python Tutorial and/or Python 3 Tutorial on classes.)

Data

- Create a class for a `Contact`. A `Contact` has attributes for a person's name and telephone number.
- Create an abstract class for `Event`, and two subclasses, `Appointment` and `Meeting`. An `Event` has an attribute for a contact, and an attribute for the starting date/time of the event. An `Appointment` has an additional attribute specifying the type of the appointment (e.g., "doctor's visit"). A `Meeting` has an additional attribute that maintains the names of the attendees.
- The `AppointmentBook` class should maintain an attribute that stores the list of events.

Methods

Hint: Don't duplicate code in the `Appointment` and `Meeting` classes. If you find yourself writing the same code in both classes, consider putting that code into the abstract class.

- Provide a `toString()` method for these classes: `Contact`, `Event`, `Appointment`, and `Meeting`. The `toString()` method for the `Meeting` class does not need to include the attendees at the meeting (just the contact information and the date/time of the meeting). The exact format of the produced strings is not important (for example, it doesn't matter if the string produced for the date part of an `Event` uses "November", "Nov", or "11" to represent the month of November).
- Write getters for the two attributes in the `Contact` class. Call them `get_name` and `get_phone`.
- Write two methods for the `Meeting` class. The `addAttendee` method adds the given name to the meeting's list of attendees. The `get_attendees` method returns the list of attendee names.
- The `Appointment` class should define a getter for the type of appointment, `get_type`.
- Write a method `add_event` in the `AppointmentBook` class that attempts to add the given event to the appointment book. The method should not add an event if the new event will conflict with an existing event in the appointment book. If there is no conflict, the given event is added to the appointment book. If there is a conflict, the method should return without adding the given event to the appointment book.
- Write a method for the `AppointmentBook` class called `get_events_for_date`. This method returns a list of `Events` for the given date.
- Write any other supporting methods needed to accomplish the behavior described above.
- Provide a main method that tests the functionality of your appointment book. Your main method will not be graded, but it's in your best interest to test your program thoroughly. When grading your program we'll be running our own set of test cases against your classes.
- Document all classes, constructors, and methods.

Submit your work as a Python file (with extension `.py`), via Canvas.