COP3330C Module 3 Graded Assignment

For this assignment we will complete tasks similar to the additions that were made to the practice exercise: implement a varargs method (also known as "variable arity") as discussed in Chapter 5 of our textbook ("Methods").

We will also add a feature to the application which will build and send calendar reminders for appointment which uses concepts from Chapter 6 ("Class Design").

- Create an "AppointmentApp" class which will serve as the main application class. The main method for the application should reside in this class.
- Add a feature to the AppointmentApp class which which uses a loop to create multiple appointments with randomized appointment times and reminder times (using the Random API). Be careful that your reminders do not occur after the appointment time.
    - The AppointmentApp class should store these appointments in an ArrayList (the "appointments list").
    - You may use the same contact for all appointments.
    - You can choose the random characteristic for the appointment times and reminder times, I used a random number of months added to the current date/time for each appointment and a different random value as the number of hours to subtract from the appointment time to produce the reminder.
- Add a method with variable arity which accepts a variable number of Appointment objects and adds them to the appointment list.
- Design a CalendarReminder interface with two abstract methods:

```
// build a reminder in the form of a formatted String
public String buildReminder(Appointment appt);

// send a reminder using contact's preferred notification method
public void sendReminder(String reminder);
```

- Store this interface in a separate .java file. Be sure to add an ID header to the top of the new file.
- Implement the methods declared in this interface in the application class.
- Use the Appointment attributes to build the reminder as a String.
- Use any other code you find necessary to format the reminder with the same appearance as the sample output shown below. Your format should pad the lines as necessary to achieve the vertical alignment shown, including a space before and after the longest line. The padding should be determined computationally based on the length of the longest line (which could vary), do not use numeric literals. Notice that these lines are left-aligned, they are not centered.
- Use the contact's preferred notification method to send the reminder (a simple output statement indicating how the reminder is being sent is adequate). Assume that for a preference of PHONE that an SMS text message will be sent.
- Add a method which iterates over the appointment list and determines if it is time to send a reminder; compare the reminder time of each appointment to the current time down to the minute; if equal, send the reminder.

- Generate test data which triggers one or more reminders to be sent (you can use the "now()" method demonstrated in the practice example to create one or more appointments in the near future with a reminder time of "now()", then call this method.

Submit your solution to the GitHub classroom repo created when you accepted the assignment invitation.

Sample output:

```
Sending the following SMS message to John Smith at (904) 555-1212

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+ Hello, John Smith!                                           +
+ This is a reminder that you have an upcoming appointment.    +
+                                                              +
+ Title: Dentist                                               +
+ Description: Cleaning appointment with Dr. Kildaire          +
+ Date: 7 December, 2023                                       +
+ Time: 08:00AM US/Eastern                                     +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```