Debra Cooperman
cs5010 assignment 3: Airline company email automation

My email automation program has 12 classes, including the main class (EmailGenerator), 2 Exception classes, and 3 enum classes for user input and flight information categories.

The three main classes used by the program to access information needed to generate the emails are:

- **Email:** provides access to email creation date and list of substitutions, which contains a HashMap with the email placeholder text (e.g., “[[departure-city]]”) as the key and the value (e.g., “Boston”) as the value. The EmailGenerator accesses this information to build an email.
- **Flight:** provides access to all the flight information needed for the email, specifically the flight number, flight departure city, flight destination city, passenger list, and event (departure or arrival).
- **Passenger:** provides access to all the passenger information, specifically first name, last name, address, city, county, state, zip, phone, email, rewards status.

The **EmailGenerator** class takes user input and uses the **EmailDataProcessor** and **EmailDataValidator** classes to process and verify that input. The **EmailDataProcessor** reads the information from the command line (the email template file, output directory, csv file with flight and passenger info, and the departure or arrival event) and uses the **EmailDataValidator** to verify it. The **EmailDataValidator** contains all the regex patterns needed to verify the validity of the user input, the template file, and the flight/passenger file, and a few helper functions that return true if the data is valid, false if not. If the data is verified, the **EmailDataProcessor** will populate the Email, Flight, and Passenger objects with the correct information. An **ErrorGenerator** is always used together with the **EmailDataValidator**, such that if there is a specific piece of data that is missing or invalid, the **ErrorGenerator** will spit out a specific message and then usage instructions. When an error happens, an **EmailDataErrorException** is thrown, which shows a “quitting program” message before quitting the program. In the specific instance of reading the email template, and substituting the email values for the placeholder text, a **NoSuchPlaceholder** exception is thrown when no replacement value exists for the placeholder text in the template.

The **UserInputCategory** and **FlightInfoCategory** enum classes contain all the categories of user input and categories of flight information used in the program with their values. I used these because I wanted to be able to manage them all in one place, and for example, if I wanted to add other events besides “departure” and “arrival” (e.g., “crash”), I could do that without affecting the makeup of the flight object.

Once all the data is processed and the Flight is populated with flight information and passenger list, the **EmailGenerator** goes through each passenger and builds an email for them by generating the substitutions list for the **Email** object. Once the Email object has a substitutions list, the **EmailGenerator** then reads each line of the template file, and if it finds placeholder text that exists as a key in the substitutions list, replaces the instances of the placeholder text with that line, then writes that line to an output file in the output directory specified by the user.