Module 1Assignment

Introduction

Quick summary: you will create a Java class, share it with your "teammates" (via GitHub) who will make modifications to your project. In the second part of this assignment you will merge these changes and make some additional changes of your own.

Please do not use an IDE for this project. Stick with whatever editor you selected (such as Visual Studio Code) and complete the git operations via the command line. Honestly there's no way for us to check this but I think this homework will be more beneficial to you if you work via the command line.

To really appreciate working with Git, you need to work in a team. This is impractical in this course so, for the sake of this homework, either our GTA or I will be your teammate.

This assignment has been split into two assignments so that the due dates show up clearly on Canvas. At the end of this first part you will submit a link to your GitHub repository. Your teammates (aka me or our GTA) will get back to you within 12 hours of your submission so that you can continue on to the second part.

Creating your repository

Throughout the semester I will say "Create (and checkout) a branch named ...". To do this **you only need to use the switch -c branch-name** command. This will automatically checkout the newly created branch. See the video on branches for more information.

1. Create a directory for your solutions to all projects in this course. Name it however you like but I recommend something like CPSC2710 since you will use this directory all semester. I'll refer to it as the "solutions directory".

2. Initialize a git repository in your solutions directory.

3. Place this .gitignore fileLinks to an external site. in your solutions directory. (Note: You'll need to rename it to .gitignore without any other extensions).

4. Commit.

5. Create a private repository on GitHub and add it as a remote to the local repository in your solutions directory. You will use this repository all semester so name it something like CPSC2710 or something else meaningful to you.

6. Push the main branch.

7. Create (and checkout) a branch called module1.

8. In your solutions directory, create a directory called module1. All of your work for this module will go in that directory. It might seem redundant to have both a branch and a directory named module1 but it will make sense later in the semester.

9. Commit. Push the module1 branch.

First class

Suppose you have some ideas about how to get going on your next great startup company's software system. You're creating an airline reservation management system since we all know how bad the current ones are. You create the first class and share it with your teammate so that they can test it.

1. Create a file called SeatReservation.java and place an outline for a class in it.
2. Commit. Push the module1 branch.
3. Add the following **private** instance variables:
   - String flightDesignator
   - java.time.LocalDate flightDate
   - String firstName
   - String lastName
4. Commit. Push the module1 branch.
5. Add the following **public** methods (these are just getters/setters so you should be able to implement them easily):
   - String getFlightDesignator()
   - void setFlightDesignator(String fd)
   - java.time.LocalDate getFlightDate()
   - void setFlightDate(java.time.LocalDate date)
   - String getFirstName()
   - void setFirstName(String fn)
   - String getLastName()
   - void setLastName(String ln)
   - String toString() -- return a string representing this object using "null" for all instance variables that are null. Here is a sample return String: SeatReservation{flightDesignator=DL1331,flightDate=2023-07-30,firstName=null,lastName=null}. Note that the date format shown here is the default for LocalDate (ie the value returned by LocalDate.toString()) so you don't need to do any fancy date formatting.
6. Commit. Push the module1 branch.
7. Add these two email addresses as collaborators on your solutions GitHub repository: Note: To add collaborators go to your repository on GitHub, select "Settings" (the gear icon), click on Collaborators and add these email addresses.

Submitting

Submit this assignment on Canvas. Paste the URL of your GitHub repository into the supplied input. Note that Canvas will try to display a preview of it but that preview may show an error since the repository is private. Don't worry, as long as your URL is correct, everything is OK.