

Module 1 Assignment Part 2

While you wait...

While you were waiting for your teammate to reply, you realized there is something you don't like with the first version but you don't want to modify the module1 branch because you aren't sure if your teammate has looked at it yet and it could be very confusing to figure out which version they were working from. So...you decide to create a "fix" branch and put your code there until you receive the tests from your teammate.

1. Create (and checkout) a branch named module1-fix1.
2. Modify the setFlightDesignator method so that it checks the number of characters in the designator. If there are fewer than 4 characters or more than 6 characters, throw `IllegalArgumentException`. This change should be only in the module1-fix1 branch at the moment.
3. Commit. Push the module1-fix1 branch.

Teammate's work

Wait for a Canvas message from your teammate (ie me or your GTA). If you don't receive one within 12 hours of your submission, send me a Canvas message to remind me that your project is ready. Note that normally GitHub users would issue a **pull request** instead of sending emails around but I didn't cover pull requests in this module.

1. Fetch your teammates changes by fetching the module1-test1 branch.
2. Switch to module1-test1 to see what code was added/modified. Switch back to module1.
3. Merge the module1-test1 branch into your module1 branch.
4. Push the module1 branch.

Running the tests

To run your teammate's tests make sure that a java compiler for java version 18 or later is available on the command line by typing:

```
javac --version
```

If an older version is displayed or you get an error that java isn't installed, go to [Oracle's Java Download site](#)[Links to an external site.](#) and install the latest compiler for your platform. Window's users *might* need to exit their terminal program before the updated command will be available. Do not proceed until the command shown above indicates java 18 or better. Post to our discussion forum if you're having problems.

The grader added a test and two jar files related to the JUnit framework. You can run their tests via the command line with:

```
javac -cp .:junit-4.13.2.jar:hamcrest-core-1.3.jar *.java
```

```
java -cp .:junit-4.13.2.jar:hamcrest-core-1.3.jar org.junit.runner.JUnitCore SeatReservationTest
```

Note that you might get errors during compilation or you might get test failures. That's OK. We'll fix them once you merge in your changes.

Merging your fix

Your teammate anticipated that flight designators must be between 4 and 6 characters so they included tests that check for this. You'll need to merge in your fix in order to get their tests to pass.

1. Merge the module1-fix1 branch into the module1 branch. Make sure you get the steps right here: switch to module1 then merge in module1-fix1. Resolve any conflicts (and make sure to continue the merge so that it results in an automatic commit).
2. Run the tests using the commands above.
3. Fix your class so that all tests are passing. **Do not modify the tests.** They represent a specification regarding how your class is *supposed* to work.
4. Commit. Push the module1 branch.

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.