03. Second Git Activity

In this activity, you'll practice cloning a provided repository, working with branches (creating, committing to, and merging them), and pushing your changes back to a shared repository. This practice will help prepare you to share code with your teammates in future weeks.

Setup

1. Log into your google cloud vm, and clone a copy of your remote repository for this assignment. To do this, replace both references to LOGIN with your CSL Username in the command below. Note that you may be prompted for your CSL password and Duo Authentication to access this repository across the internet.

   ```
   git clone ssh://LOGIN@best-linux.cs.wisc.edu:/p/course/cs400/activities/repos/03.SecondGitActivity_LOGIN
   ```

2. Change directories to the 03.SecondGitActivity_LOGIN folder that was just created (cloned) as a result of running that command. Within the main branch of this repository, you should find an ILibrary.java file. Review the contents of this file.

Activity

3. Create a new branch to use while developing an implementation of the ILibrary interface. Make the name of this new branch: DevelopLibrary. And checkout this branch so that it is the current branch that you are developing under.

4. Define a new public class called Library that implements the ILibrary interface. Your calculateAnswer() method in this class should simply return the String value "42" (this answer refers to a joke from the Hitchhiker's Guide to the Galaxy series of book).

5. You can compile your program with: `javac Library.java`
   You will need to run this command whenever you make changes to your Library.java file. Add and commit this work to your DevelopLibrary branch. Ensure that your commit message for this commit includes the exact string: "implemented new class named Library". ***(Note: Make sure that you remove .class files before commit).***
   Command to remove .class files: `rm *.class`

6. Your repository also has another branch called DevelopApp. Checkout this branch to review its contents (all three files: App.java, Library.java, and .gitignore files). Do not make any changes to Library.java present in DevelopApp branch. Then merge the contents of the DevelopApp branch into the main branch in your repository. This is one way of simulating a teammate contributing to this branch while you are working on a different one.

7. Next merge the contents of your DevelopLibrary branch into the main branch. Since some of your changes conflict with the changes merged from the DevelopApp branch, you'll need to resolve those conflicts to complete this merge. After merging, commit these changes with a message that contains the exact string: "merged DevelopLibrary and DevelopApp branches".

8. Edit the App.java source file so that an instance of the Library class is instantiated instead of an instance of the LibraryPlaceholder class.

9. Now, compile the Library.java and App.java files using `javac Library.java App.java`. To run App.java, use the command: `java App` Running this command should display the line as below:
   "42 is the answer."

10. If not, continue fixing errors and bugs until this happens. Then make one final commit with a message containing the exact string: "App is correctly using Library".

Submission

11. Run git status to confirm that your changes have all been committed to the main branch. You will see a message that your main branch is now ahead of the origin that you previous cloned. Run the command "git push" to share these changes back to that shared repository, so that your course staff can find them there for grading. Note that you may be prompted for your CSL password and Duo Authentication to access this repository across the internet.
12. If you'd like to confirm that your changes have been successfully pushed, you can try making a new clone of that remote repository to a different LOCAL_DIRECTORY, and then check the git log and file contents to confirm that they look correct. Here's the command to clone another copy of this remote repository to a different LOCAL_DIRECTORY:

    ```
    git clone ssh://LOGIN@best-linux.cs.wisc.edu:/p/course/cs400/activities/repos/03.SecondGit
    Activity_LOGIN LOCAL_DIRECTORY
    ```

    Note that you may be prompted for your CSL password and Duo Authentication to access this repository across the internet.