Frank's Suggestions - OOP Project

1. **One team member** should:
    a. Create the repo for the project in GitHub.
    b. Open a GitBash window
    c. Navigate to the folder where you have other GitHub repos
    DO NOT USE A FOLDER THAT ALREADY HAS A REPO IN IT.
    if you are not sure if it has a repo in in it:
        1. **Open Windows File Explorer**
        2. **Navigate to the folder you want to use for the OOP project repo**
        3. **Be sure it does not have a folder called "`.git`" (it should be the first one)**
        4. **If it has a "`.git`" folder, it already contains a repo and should not have another repo in it. Choose another folder to hold the OOP project repo.**
    d. Clone the repo down to the folder you have chosen on your local machine.
    e. Open Visual Studio and create a new project for the OOP project in the cloned repo folder.
    f. Save all the files in Visual Studio.
    g. Exit Visual Studio.
    h. Push the Visual Studio project just created to GitHub. (`git add`, `git commit`, `git push`)
    i. Copy the GitHub repo URL.
    j. Send the GitHub URL to other team member(s).

2. **Other team members** (not the one that created the repo and pushed the project):
    a. Open a GitBash window
    b. Navigate to the folder where you have other GitHub repos.
    DO NOT USE A FOLDER THAT ALREADY HAS A REPO IN IT.
    (Use the instructions above if you are not sure if it already contains a repo or not)
    c. Copy the URL given to you by the team member that created the OOP project repo.
    d. Clone the OOP project repo down to your machine using the URL given to you.
    e. Issue the command '`ls`' to be sure you have successfully cloned the repo.

3. **All team members**:
    a. Open the OOP project in Visual Studio and be sure you have all the files pushed by the creator of the project.
    b. Exit Visual Studio.
    c. Go back to your GitBash window (open another one).
    d. Navigate to the OOP project folder (if you are not already there).
    e. Issue the command "`git branch`" to see all current branches of the repo. You should just see one called "`main`" (or "`master`" depending on how the repo was created).
    f. Create a new branch for your work: `git checkout -b branch-name`
    g. Save the branch to GitHub: `git push origin branch-name`

Frank's Suggestions - OOP Project

h. Issue the command: `git branch`

You should see your new branch listed with an "*" next to it and any other branches for the repo.

i. Open the OOP project in Visual.

j. Look in the lower right corner. You should see your branch name as the one you are working in. (if not: click the arrow next to the branch name; you should be able to select your branch)

k. You will be working and saving your changes in your branch.

4. To save your changes from your branch to GitHub:

   a. Verify you are in your branch: `git branch` (look for the name with the "*" next to it)
   (if you are not in branch you want to be in: **`git checkout`** *`branch-you-want-to-be-in`*)

   b.
   ```
   git add -A
   git commit -m'meaningful-comment-about-what-you-did'
   git push origin branch-name
   ```

5. When you are ready to merge your changes back to base code:

   a. Navigate to the folder with the OOP Project on your machine.

   b. Be sure all the changes from your branch have been pushed.

   c. Navigate to the "**`main`**" (or "**`master`**") branch: **`git checkout main`** (or **`master`**)

   d. Issue the command: **`git merge`** *`branch-name`*
   (*`branch-name`* *name of the branch* containing code to be merged)

   e. Verify all changes have been pushed: **`git status`**

   If you receive merge conflict errors contact your Instructor or Teaching Assistant to help you resolve them unless you are ABSOLUTLEY sure you can do it yourself.

   f. Notify your teammates you have merged your changes so they "add, commit and push" their changes and then pull yours: `git pull origin main (or master)`

Don't forget, you can always go to GitHub using the URL to verify your changes have been saved.

If you don't know the GitHub URL: `git remote -v` will show it to you when you are in the repo folder on your local machine.

Please do not hesitate to contact your instructor or Teaching Assistant if your unsure of any of this, or anything else, before you attempt it. It is much easier to avoid a problem by asking for help than trying to fix one.

Frank's Suggestions - OOP Project

Once you have your repos and branches setup, you should start thinking about the project.

As a team:

1. Analyze the problem statement/requirements identifying the major/big pieces /processes.
2. Break the major pieces down into smaller manageable pieces
3. Decide which features you want to implement first.
4. Assign on feature or part of a feature to each team member.

   It is not unusually for the analysis of the problem and designing the pieces of the solution to take longer than expected. **Resist the urge to just start coding.**

Team members should only be working on and making changes to files associated with their piece.

When your piece of work is complete, save your changes and merge your branch into main.

Consider using a separate branch for each feature you work on.

Avoid having more than one person making changes to any file in the project. This will usually lead to merge conflicts. Even adding or changing spacing in a file that someone is working on may cause a merge conflict.

Remember you are a team of equals, respect each other's skills and abilities. If one of your team mates is struggling look at as a learning opportunity for YOU to help them. Don't just give them the answer, help them understand and come to the answer themselves.

Communication is the lifeblood of the team, be sure your teammates are always aware of your status, availability and skill level.