Note for evaluators:
Since my teammates use Eclipse, I chose to use Eclipse as well instead of IntelliJ for this assignment. I hope that's not an issue, as both are Java IDEs and this also helps me get more comfortable with Eclipse.

1. **What is the use of the git add command?**
   The git add command is used to stage changes. This means it tells Git which changes (new files, deletions, or modifications) we want to include in our next commit. It is like selecting items to put into a shopping cart, git add picks the changes to include in the next snapshot of our project.

2. **Write the command to create a branch “feature-group”.**
   git branch feature-group

3. **What is a pull request? Write the steps involved in creating a pull request.**
   A pull request (PR) is a way to ask others to review and merge your code into a main branch like main or develop. It’s commonly used in teams to collaborate and ensure code quality.

   Steps to create a pull request:
   1. First, create a branch and push your changes to it using git push origin <branch-name>.
   2. Go to your repository on GitHub.
   3. You’ll see a prompt to “Compare & pull request” – click it.
   4. Add a title and description explaining what your changes do.
   5. Choose the base branch (usually main) and compare it with your feature branch.
   6. Click “Create pull request”.
   7. Your team can now review and merge your PR after approval.

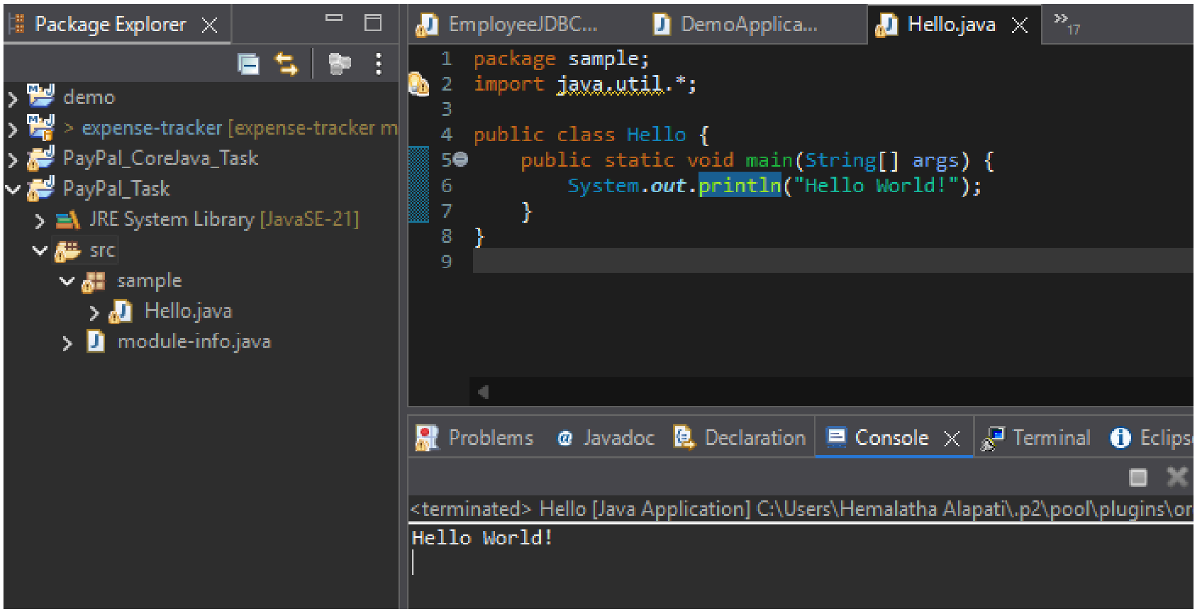
4. **How do you add collaborators?**
   - To add collaborators on GitHub:
   - Go to your repository page.
   - Click on the Settings tab.
   - From the sidebar, click Collaborators and teams (or just Collaborators).
   - Click the “Add people” button.
   - Type the GitHub username or email of the person you want to add.
   - Click “Add” or “Invite”.
   - They will receive an invitation to access your repo.

5. **What is the command used to commit our changes in github repo?**
   Once changes are staged using git add, we use:
   git commit -m "commit message here"
   This creates a commit with the message describing your changes. After this, you can push it to GitHub using: **git push origin <branch-name>**

6. **Create a github account and repository to save all your assignment tasks. Example repository name: rg-assignments**
   Done. I created a repository named:
   rg-tasks already

7. **Create a branch called feature-java(you can use it later to push all your java related tasks to this branch)**
   git checkout -b feature-java
   This creates and switches to the feature-java branch.

8. **Install IntelliJ IDE. Create a simple java project displaying “Hello World!!”and share the screenshot.**
   As I initially mentioned, I used Eclipse for this:



```java
package sample;
import java.util.*;

public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

```
<terminated> Hello [Java Application] C:\Users\Hemalatha Alapati\.p2\pool\plugins\or
Hello World!
```

9. **In IntelliJ IDE, explore the shortcut keys for the following**
   a. **To run the project -** Ctrl + F11
   b. **To select a line -** Shift + End
   c. **To comment/uncomment a line -** Ctrl + /
   d. **To delete a line -** Ctrl + D