# Problem Statement:

Hooli Inc. is working on a product for which there is a distributed development team across geographies. Coordinating between the development team is proving to be difficult but the code needs to be synchronized. The company has chosen Git as the source repository for the product development. To check the functionality some sample math helper functions have been developed. Check the Git integration with the help of the available code.

Functions:

* Addition
* Multiplication
* Calculation of sin/ cos
* Distance between 2 points

Create these sample functions

# Steps to Perform:

* Create user stories and task for the functionality and assign the sprints using Azure Board
* Assign the stories and tasks to the developer
* Change the status as per task execution
* Create a Git project holding the project
* Upload the project to Git
* Add new calculation functions to the project and check-in the changes to Git
* Compare the historical changes to see the changes introduced
* Creating a new local repository from the master repository for any new local repository creation

# Solution

User Stories and Tasks:

1. As a user, I want to be able to perform addition, so that I can add two numbers.

Tasks:

- Create a function for addition

- Test the function with different input values

- Add the function to the project

- Commit the changes to Git

2. As a user, I want to be able to perform multiplication, so that I can multiply two numbers.

Tasks:

- Create a function for multiplication

- Test the function with different input values

- Add the function to the project

- Commit the changes to Git

3. As a user, I want to be able to calculate sin/cos, so that I can calculate trigonometric functions.

Tasks:

- Create a function for sin/cos calculation

- Test the function with different input values

- Add the function to the project

- Commit the changes to Git

4. As a user, I want to be able to calculate the distance between two points, so that I can calculate the distance between two locations.

Tasks:

- Create a function for distance calculation

- Test the function with different input values

- Add the function to the project

- Commit the changes to Git

Sprints:

Sprint 1:

- User Story 1

- User Story 2

Sprint 2:

- User Story 3

- User Story 4

Assignment of Stories and Tasks:

- Assign User Story 1 and User Story 2 to Developer 1

- Assign User Story 3 and User Story 4 to Developer 2

Status Change:

- Developer 1 will update the status of User Story 1 and User Story 2 as "In Progress" and "Done" respectively.

- Developer 2 will update the status of User Story 3 and User Story 4 as "In Progress" and "Done" respectively.

Git Integration:

- Create a Git project for the product development.

- Upload the project to Git.

- Add new calculation functions to the project and check-in the changes to Git.

- Compare the historical changes to see the changes introduced.

Creating a New Local Repository:

- To create a new local repository from the master repository, follow the steps below:

1. Clone the master repository to your local system using the Git clone command.

2. Create a new branch for the new local repository using the Git branch command.

3. Switch to the new branch using the Git checkout command.

4. Make the changes to the code and commit the changes using the Git commit command.

5. Push the changes to the new branch using the Git push command.

# Step by Step Guide

1. Sign up for an Azure DevOps account at https://azure.microsoft.com/en-us/services/devops/
2. Create a new project in Azure DevOps by clicking on "New Project" button.
3. Choose a name for your project and select the appropriate version control system (Git) as the source control.
4. Create user stories in the Azure Boards section by selecting "Boards" in the left-hand menu and then selecting "Backlogs".
5. Add the four user stories mentioned in the problem statement. For each user story, add a description and acceptance criteria.
6. Create tasks for each user story by clicking on the user story and selecting "New Task".
7. Assign each task to a developer.
8. Once the tasks are assigned, each developer can move their assigned tasks to the "In Progress" status by clicking on the task and changing the status in the drop-down menu.
9. Once the developer completes a task, they can move it to the "Done" status.
10. Add the project code to the Azure DevOps repository by selecting "Repos" in the left-hand menu and then selecting "Import Repository".
11. Select "Git" as the repository type and provide the repository URL. Once the repository is created, you can clone the repository to your local machine.
12. Create a new branch in the local repository for each user story or task using the "git branch" command in the command prompt.
13. Switch to the new branch using the "git checkout" command.
14. Write the code for the task or user story and test it locally.
15. Once the code is tested and working correctly, commit the changes using the "git commit" command.
16. Push the changes to the remote repository using the "git push" command.
17. Once the changes are pushed, create a pull request in Azure DevOps by selecting "Pull Requests" in the left-hand menu and then selecting "New Pull Request".
18. Select the appropriate branch and provide a description of the changes made. Then, select the reviewers and assign the pull request to them.
19. Once the reviewers approve the pull request, merge the changes into the main branch.

Repeat steps 12-19 for each user story or task.

That's it! By using Azure DevOps, you can easily manage your project, collaborate with other developers, and ensure that your code is stored safely in a version control system.