**Dungo, Jinnel I.**

**IV-ACSAD**

General Instructions: This activity is expected to take approximately 1.5 hours. Please ensure careful adherence to the provided instructions.

Objective:

• This exercise focuses on using Git in a local environment without remote repositories. You'll practice setting up a local Git repository, working with branches, managing commits, and resolving conflicts locally.

Problem Statement: You are working on a project locally, and need to use Git for version control. Your task is to initialize a Git repository, implement features using branches, manage commits, and handle conflicts that may arise when merging different branches

1. Open Microsoft Word or any word processing software that supports document creation.

2. Start a new document.

3. Copy and paste the tasks for Machine Problem #1 into the document. Ensure proper formatting and readability.

4. Insert your source code into the document. You can either copy and paste it directly or take a screenshot of the code and insert the image

into the document.

5. Capture screenshots of the output of your code execution. Make sure the screenshots are clear and readable.

6. Insert the screenshots into the document. You can either directly insert the images or embed them as links, depending on your preference. and the document's requirements

7. Arrange the content in a logical and organized manner. You may want to use headings, subheadings, and bullet points to make the document easy to follow.

8. Review the document for any errors or formatting issues. Make necessary adjustments to ensure clarity and correctness. 9. Save the document as a PDF file. Most ward processing software offers an option to save or export documents as PDFs. Choose this option and follow the prompts to save your document in PDF format.

10. Once saved, review the PDF file to ensure that all content, including instructions, source code, and screenshots, is accurately captured and presented

11. If everything looks good, your Word document containing instructions, source code, and screenshots, saved as a PDF file, is ready for submission or sharing.

Please follow the filename format (minus 10 for the wrong filename) MP1<Lastname>.pdf

Example: MP1Mansueto.pdf

Scenario:

You're working on a personal project named "<YourLastname>\_local\_project and want to experiment with new features without affecting the main codebase

Example Mansueto local\_project

Note:

Ensure Git is installed on your machine.

Set up your Git usemame and email (if not already configuredi

Create a directory called Your Lastname\_local project, navigate into it, and initialize it as a Git repository

Multiple commits reflecting changes in different branches

Evidence of conflict resolution during branch merging (optional)

A Git tag-marking version v1.0 (optional).

Tasks:

1. Create a new branch:

Create a new branch named "feature\_x" to isolate your experimental changes

2. Stage changes:

Make modifications to your project files.

Stage the changes using the gat add command.

3. Commit changes:

Commit the staged changes to your local repository using the git commit command. Provide a clear and concise commit message.

4. Switch back to the main branch:

Use the git checkout command to switch back to the main branch.

5. Merge changes from the feature branch:

Use the git merge command to merge the changes from the feature x branch into the main branch. Resolve any merge conflicts that may arise

6. Viewing the Commit History:

Use Git commands to view the commit history and ensure that all commits are in place.

7. Tagging Versions (Optional):

After completing the merge, use the it tags command to create a tag named v1.0 to mark the first version of your project.







