**Steps Involved in Creating the TASK**

**Task 1:**

**- Create a new Git repository on GitHub or GitLab.**

**- Add a Java program (e.g., the Factorial program I provided earlier) to the repository.**

**- Create a new Jenkins Freestyle project.**

**- Configure the project to:**

**- Check out the Git repository.**

**- Compile and run the Java program.**

**- Display the output in the Jenkins console.**

Step1: Create a folder ‘Task3’ in local machine

Step2: write a Java code for the factorial of a number and save it with .java extension

Step3: open a command prompt window in this folder path “D:\Task3”

Step4: Create a new git repo named “java factorial” in git hub

Step5: config the git with username and email with following commands

git config –-global user.name”chinnuanna123”

git config –-global user.email.com”[kurianchinnuanna@gmail.com](mailto:kurianchinnuanna@gmail.com)”

Step6: Now create a path to connect Git repo and local machine

git config –-global init.default branch main

Step7: To initialize the git repo we use the command

git init

Step8: now add remote origin using the command

git remote add origin <url>

Step9: Now add the Factorial.java file to the staging area

git add Factorial.java

Step10: To push the file from staging area to local repo

git commit -m”pushing the file”

Step11: Now finally to push to the remote repo

git push -u origin master

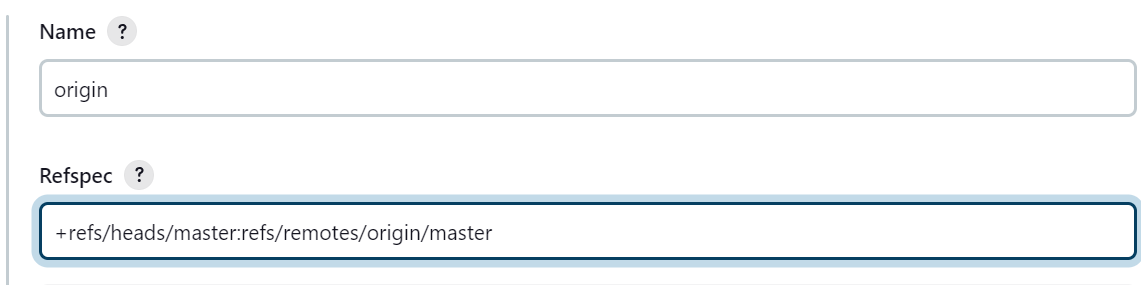
Steps12: Next creating a build of the source code in github using Jenkins

Steps13: Open Jenkins dashboard ,click on create “new item” to create a job

Steps14: create Jenkins free style project

Step15: In SCM select GIT. Add repository url then add credentials to configure git with Jenkins.

Step16:



Step17: In build steps click on execute windows batch command to compile and run java program

javac Factorial.java

java Factorial.java

Step18: Finally apply and Save

Step19: click on Build button



**Task2**

**- Create a new Git branch (e.g., "feature/new-program") and add a new Java program (e.g., a simple calculator). - Update the Jenkins project to: - Check out the new branch. - Compile and run the new Java program. - Display the output in the Jenkins console. - Merge the new branch into the main branch (e.g., "master").**

Step1: In the same above git repo created another branch named “new-program”

Step2: Add ‘Calculator.java’ file to the branch

Steps3: git branch new-program

Step4: git push -u origin new-program

Step5: git add Calculator.java

Step6: git commit -m”pushing files”

Step7: git push -u origin new-branch

Step8: Next created a Jenkins freestyle project for calculator.java

Step9: 