

Final Project: Overview and Grading Criteria

Estimated duration: 15 minutes

The objective of this final assignment is to demonstrate your ability to apply the skills and knowledge gained in both the GitHub User Interface (UI) and Git Command Line Interface (CLI). This assignment is structured in two distinct parts to guide you through both the UI-based and CLI-based approaches, ensuring a comprehensive understanding of version control workflows.

In this hands-on project, you will create an open source project in Git, make changes to that project and make it available to the community. You will begin by forking an existing public GitHub repository to create a personal copy of the project in your GitHub account. Next, you will clone this forked repository to your GitHub cloud account, where you will make meaningful changes or enhancements to the project. After completing your updates, you will use Git commands to stage, commit, and push these changes back to your remote fork on GitHub. The final step involves creating a pull request to propose your changes be merged into the original repository, making your contributions publicly available to the broader developer community. By completing this assignment, you will gain practical experience in managing and contributing to collaborative coding projects using Git and GitHub.

Part 1: GitHub UI

You recently got hired as a developer in a micro-finance startup with a mission to empower and provide opportunities to low income individuals. The core team currently uses Subversion (SVN) for managing code. They want to slowly move their code to Git. You are asked to host their sample code to calculate simple interest on GitHub in a new repository as the first step in this journey. You will not only host the script, but also follow best practices introduced in this course and create supporting documents for the open source project including code of conduct, and contributing guidelines. Additionally, the repository should be available to the community under the Apache License 2.0.

Part 2: Git CLI

Congratulations on starting the journey with your company by creating an open source Simple Interest Calculator bash script on GitHub. Your changes have been accepted and merged and the company has created a new global [repository](#) for the teams to collaborate. Other developers have contributed to this repository over time. Your team has found a mistake in one of the markdown files. You are asked to fork this repository and fix the mistake by using Git CLI in the provided lab environment and open a pull request.

Grading Criteria

You can submit your project deliverables in one of the following ways:

- **Option 1: AI-Graded Submission and Evaluation**

When you choose Option 1, you will be redirected to an AI tool where you can upload your deliverables, which may include URLs, terminal outputs, code snippets, or screenshots. You will then receive an AI-generated grade that will automatically reflect on your Coursera progress page.

- **Option 2: Peer-Graded Submission and Evaluation**

When you choose Option 2, you will upload your deliverables—such as URLs, terminal outputs, code snippets, or screenshots—through the My Submission section. Your submission will then be reviewed either by your peers or by Coursera's AI grader.

We recommend using Option 1 for faster grading. However, if you face any issues or cannot access it, you may use Option 2 instead.

If you encounter any grading problems, please reach out to the Course Team through the Discussion Forums.

Please find the details of the Grading Criteria below:

▼ Criteria for Option 1: AI-Graded Submission and Evaluation:

- A URL submission for 5 tasks
- A text response based submission for 4 tasks

The final assignment grading criteria for the two parts are as follows:

Part 1: GitHub UI (12 points)

Task 1: Submit the GitHub repository **URL** where the updated **README.md** file is located.

- Make sure the repository is named **github-final-project** and the README.md file is updated with the project name and details about the Simple Interest Calculator. (4 points)

Task 2: Submit the GitHub repository **URL** that redirects to the **Apache 2.0 license** file.

- Make sure the file is named **LICENSE** and contains the following sections: Definitions, Grant of Copyright License, Grant of Patent License, Redistribution, Submission of Contributions, Trademarks, Disclaimer of Warranty, Limitation of Liability, and Accepting Warranty or Additional Liability. (2 points)

Task 3: Submit the GitHub repository **URL** where the **CODE_OF_CONDUCT.md** file is present.

- Make sure the file includes the following sections: Our Pledge, Our Standards, Enforcement Responsibilities, Scope, Enforcement, Enforcement Guidelines, and Attribution. (2 points)

Task 4: Submit the GitHub repository **URL** where the **CONTRIBUTING.md** file is present.

- Make sure the file includes content stating that all contributions, bug reports, bug fixes, documentation improvements, enhancements, and ideas are welcome. (2 points)

Task 5: Submit the GitHub repository **URL** where the **simple-interest.sh** file is present.

- Make sure the file contains a Bash script that calculates simple interest based on user input for principal, rate of interest, and time period. (2 points)

Part 2: Git CLI (8 points)

Task 1: Submit the **curl command and terminal output** saved in a file named **forked-repo**, showing that your repository has been successfully forked.

- Make sure to fork the repository **ibm-developer-skills-network/mcino-Introduction-to-Git-and-GitHub**. (2 points)

Task 2: Submit the terminal output saved in a file named **merge_branches**, showing that one file was changed during the first merge.

- Make sure to include the commands **git checkout main** and **git merge bug-fix-typo** to demonstrate that the typo was fixed and the changes were merged with the main branch. (2 points)

Task 3: Submit the **curl command and terminal output** saved in a file named **bug-fix-revert**, confirming that the pull request is valid and shows the forked repository from which it was created.

- Make sure to revert the typo and create a pull request from your forked repository. (2 points)

Task 4: Submit the **curl command and terminal output** from the Branches page, showing the branch names and their status in your repository. The terminal output must clearly display the **bug-fix-typo**, **bug-fix-revert**, and **main** branches along with their status information.

- Make sure to check the branch status before submission to ensure all branches are visible in the output. (2 points)

▼ Criteria for Option 2: Peer Review Submission and Evaluation:

The final assignment grading criteria for the two parts are as follows:

Part 1 GitHub UI (12 points)

- Task 1: Submit the URL of the repository named ‘github-final-project’ ensuring it contains all the required files. (2 pts)
- Task 2: Submit the URL of the **LICENSE** file (Apache 2.0 License) present in the repository. (2 pts)
- Task 3: Submit the URL of **README.md** file present in the repository. (2 pts)
- Task 4: Submit the URL of **CODE_OF_CONDUCT.md** file present in the repository. (2 pts)
- Task 5: Submit the URL of **CONTRIBUTING.md** file present in the repository. (2 pts)
- Task 6: Submit the URL of the file **simple-interest.sh** present in the repository. (2 pts)

Part 2 Git CLI (8 points)

- Task 1: Submit the URL of the forked repository named as **mcino-Introduction-to-Git-and-GitHub** ensuring it contains all the required contents. (2 pts)
- Task 2: Submit a screenshot of the first merge, showing that the current branch is main and the changes made to the **README.md** file in the **bug-fix-typo** branch have been successfully merged into main. (2 pts)
- Task 3: Submit the URL of the pull request that has been accepted, ensuring the link is accessible and reflects the created pull request. (2 pts)
- Task 4: Submit the URL of the Branches page in your repository, showing all three branches - **main**, **bug-fix-typo**, and **bug-fix-revert**, along with their status. (2 pts)

Prework: Sign up for GitHub

Create a [GitHub](#) account, if you don't have one already.

Next Steps

Follow the instructions for each part of the lab. You will be prompted to note the URLs into a text file for later use in submission. You also have to take screenshot for one of the tasks and save the image as jpg or png. In the Final Submission section of the course, you will be submitting your responses for an automated grading. You can use any editor app to keep note of your URLs.

Author(s)

Md Haroon Hussain



Skills Network