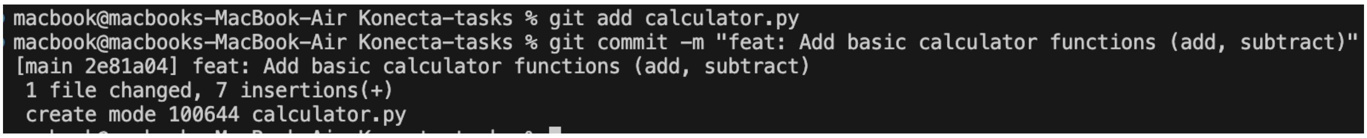
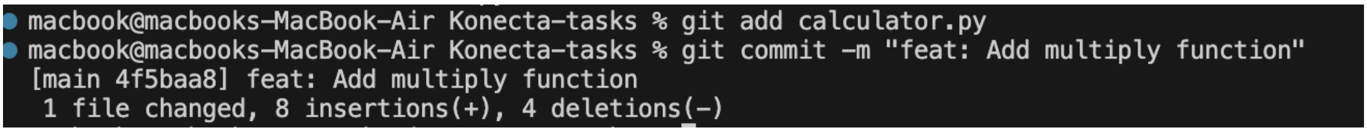
📌1- A list of all Git commands you used📌

1- git add calculator.py 2- git commit -m "feat: Add basic calculator functions (add, subtract)" 3- git add calculator.py 4- git commit -m "feat: Add multiply function" 5- git add calculator.py 6- git commit -m "fix: Optimize subtract function logic (introducing bug)" 7- git status 8- git log --oneline 9- git add calculator.py 10- git commit -m "feat: Add divide function with zero check" 11- git diff '7e2d4d1~1' 7e2d4d1 -- calculator.py 12- git status 13- git stash 14- git reset --hard 4f5baa8 15- git log --oneline 16- python3 calculator.py 17- git stash pop 18- git push --force origin

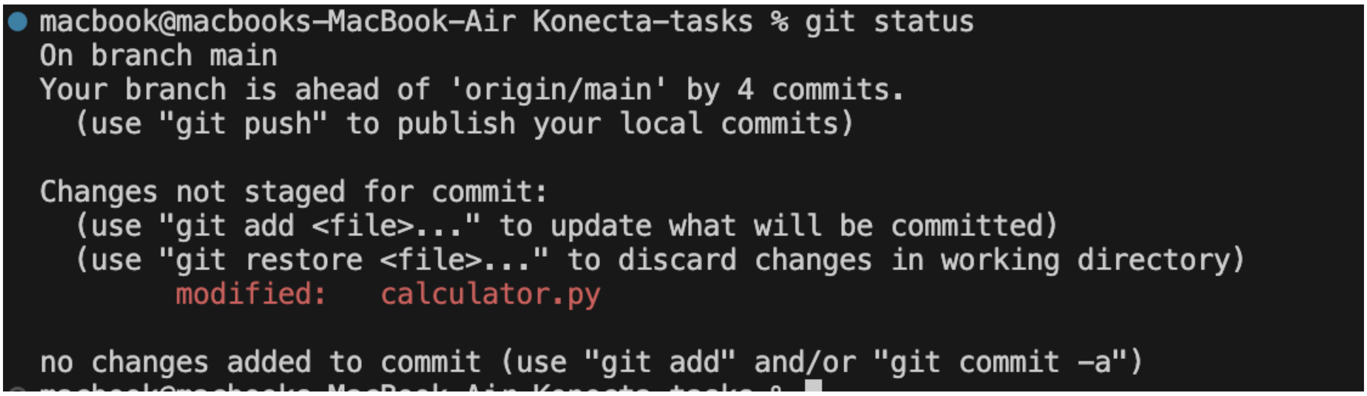
📌2-Screenshots of your terminal output📌



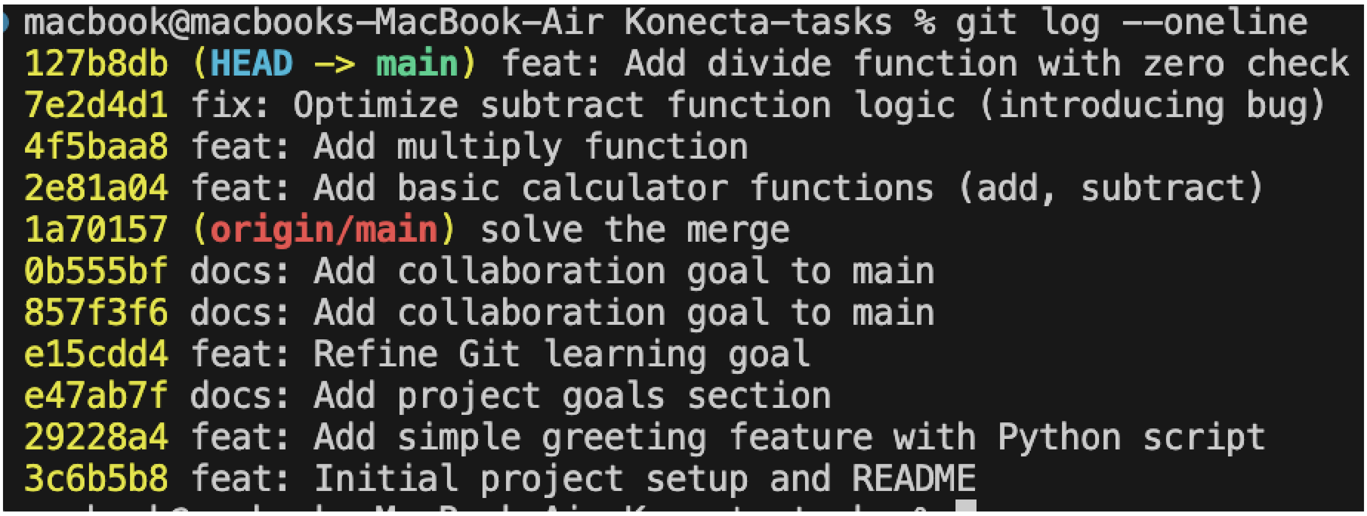
```
macbook@macbooks-MacBook-Air Konecta-tasks % git add calculator.py
macbook@macbooks-MacBook-Air Konecta-tasks % git commit -m "feat: Add basic calculator functions (add, subtract)"
[main 2e81a04] feat: Add basic calculator functions (add, subtract)
 1 file changed, 7 insertions(+)
 create mode 100644 calculator.py
```



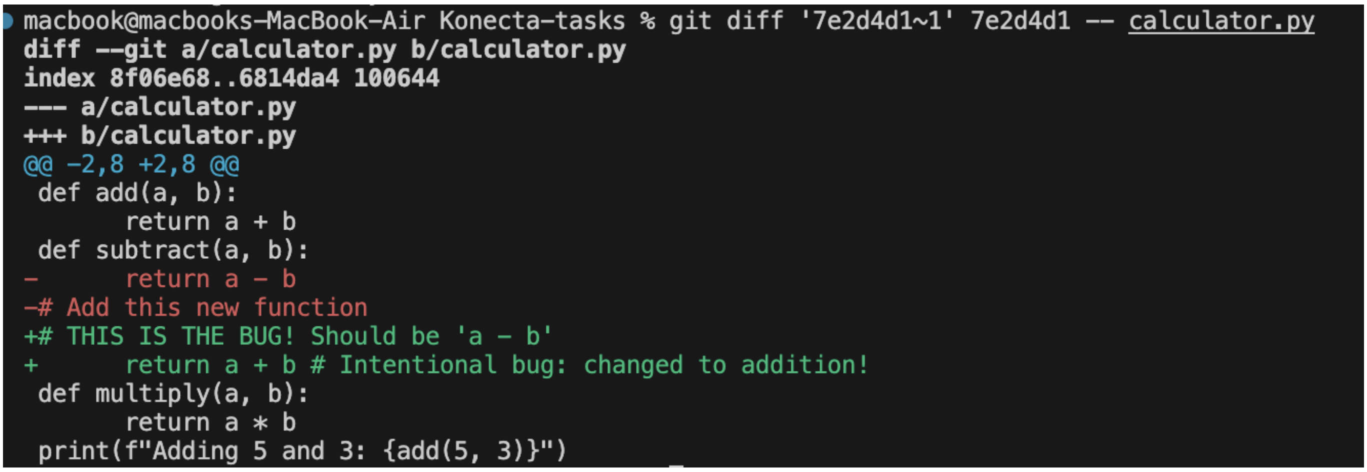
```
macbook@macbooks-MacBook-Air Konecta-tasks % git add calculator.py
macbook@macbooks-MacBook-Air Konecta-tasks % git commit -m "feat: Add multiply function"
[main 4f5baa8] feat: Add multiply function
 1 file changed, 8 insertions(+), 4 deletions(-)
```



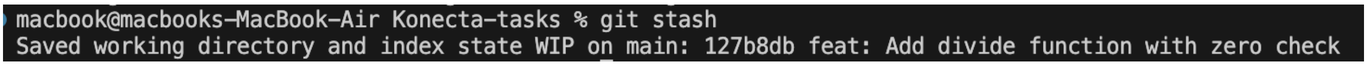
```
macbook@macbooks-MacBook-Air Konecta-tasks % git status
On branch main
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   calculator.py

no changes added to commit (use "git add" and/or "git commit -a")
```



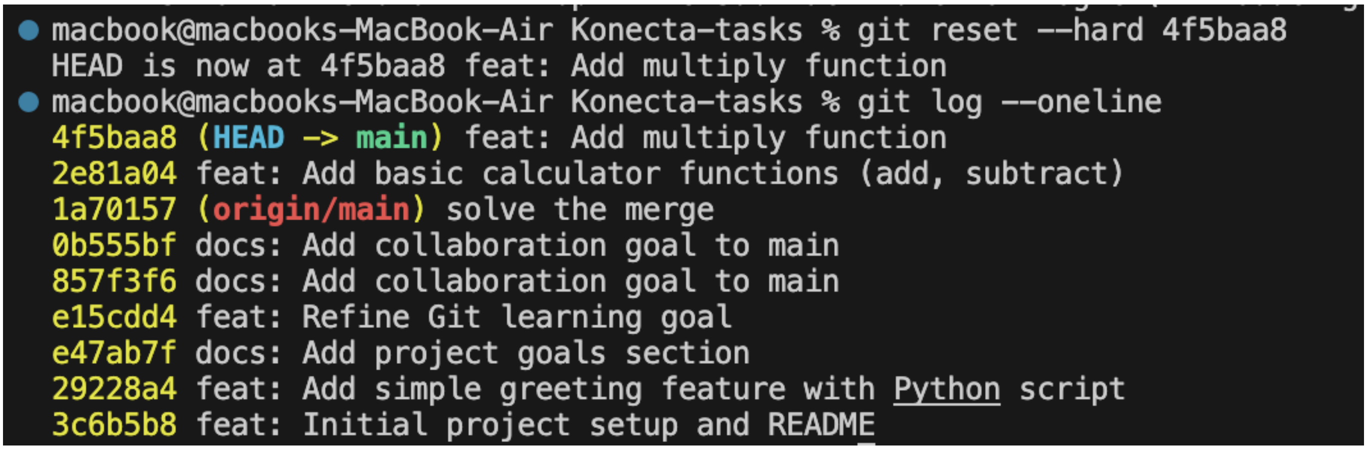
```
macbook@macbooks-MacBook-Air Konecta-tasks % git log --oneline
127b8db (HEAD -> main) feat: Add divide function with zero check
7e2d4d1 fix: Optimize subtract function logic (introducing bug)
4f5baa8 feat: Add multiply function
2e81a04 feat: Add basic calculator functions (add, subtract)
1a70157 (origin/main) solve the merge
0b555bf docs: Add collaboration goal to main
857f3f6 docs: Add collaboration goal to main
e15cdd4 feat: Refine Git learning goal
e47ab7f docs: Add project goals section
29228a4 feat: Add simple greeting feature with Python script
3c6b5b8 feat: Initial project setup and README
```



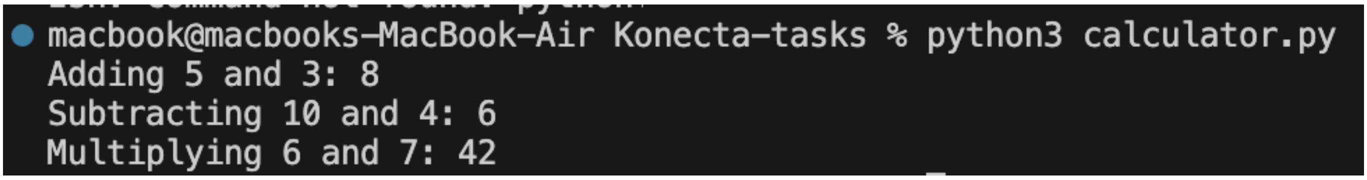
```
macbook@macbooks-MacBook-Air Konecta-tasks % git diff '7e2d4d1~1' 7e2d4d1 -- calculator.py
diff --git a/calculator.py b/calculator.py
index 8f06e68..6814da4 100644
--- a/calculator.py
+++ b/calculator.py
@@ -2,8 +2,8 @@
 def add(a, b):
       return a + b
 def subtract(a, b):
-      return a - b
-# Add this new function
+# THIS IS THE BUG! Should be 'a - b'
+      return a + b # Intentional bug: changed to addition!
 def multiply(a, b):
       return a * b
 print(f"Adding 5 and 3: {add(5, 3)}")
```



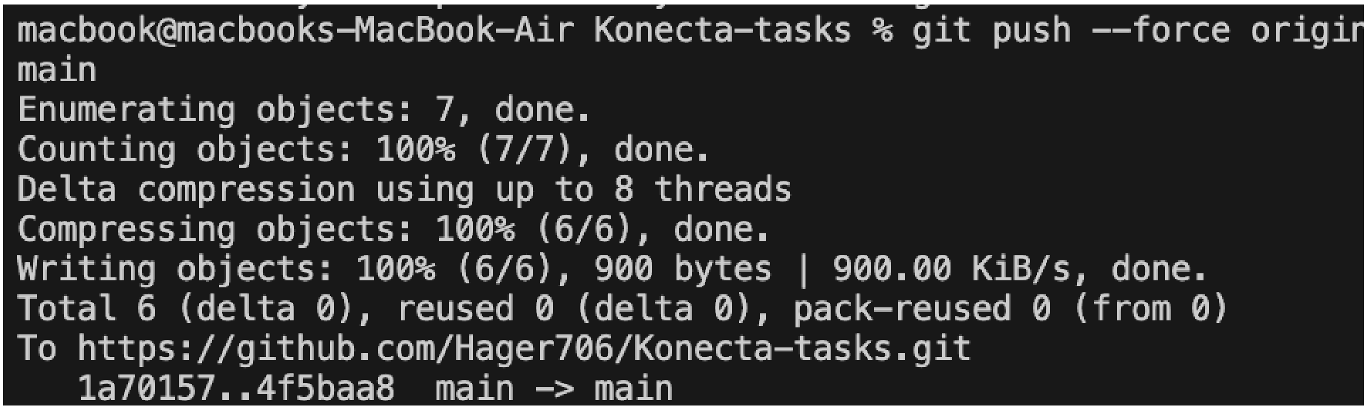
```
macbook@macbooks-MacBook-Air Konecta-tasks % git stash
Saved working directory and index state WIP on main: 127b8db feat: Add divide function with zero check
```



```
macbook@macbooks-MacBook-Air Konecta-tasks % git reset --hard 4f5baa8
HEAD is now at 4f5baa8 feat: Add multiply function
macbook@macbooks-MacBook-Air Konecta-tasks % git log --oneline
4f5baa8 (HEAD -> main) feat: Add multiply function
2e81a04 feat: Add basic calculator functions (add, subtract)
1a70157 (origin/main) solve the merge
0b555bf docs: Add collaboration goal to main
857f3f6 docs: Add collaboration goal to main
e15cdd4 feat: Refine Git learning goal
e47ab7f docs: Add project goals section
29228a4 feat: Add simple greeting feature with Python script
3c6b5b8 feat: Initial project setup and README
```



```
macbook@macbooks-MacBook-Air Konecta-tasks % python3 calculator.py
Adding 5 and 3: 8
Subtracting 10 and 4: 6
Multiplying 6 and 7: 42
```



```
macbook@macbooks-MacBook-Air Konecta-tasks % git push --force origin
main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 900 bytes | 900.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Hager706/Konecta-tasks.git
   1a70157..4f5baa8  main -> main
```

📌3-Your final calculator.py file📌

```
#calculator.py
def add(a, b):
      return a + b
```

```
def subtract(a, b):
      return a - b
#Add this new function
def multiply(a, b):
      return a * b
print(f"Adding 5 and 3: {add(5, 3)}")
print(f"Subtracting 10 and 4: {subtract(10, 4)}")
print(f"Multiplying 6 and 7: {multiply(6, 7)}")
print(f"Dividing 20 by 5: {divide(20, 5)}")
#calculator.py (at the very end of the file)
#Some new experimental print statement
print("Experimental feature: Fibonacci sequence next")
```

** 📌4-Written Explanations:** 📌

1-Briefly explain how git log and git diff helped you identify the erroneous commit.

git log helped me see the history of commits, so I could identify which commit introduced the bug. Then, git diff allowed me to compare the buggy commit with the previous one, so I could see exactly which line caused the problem.

2-Explain why git stash was necessary in this scenario.

git stash was necessary because I wanted to temporarily save my current code changes without committing them so git safely stored my changes, and I was able to bring them back later

3-Crucially, explain your choice between git reset --hard and git revert. Detail the pros and cons of the method you chose in this specific scenario, and when you would use the other method.

I chose git reset --hard because I wanted to completely remove the buggy commit from the history this method was faster and cleaner since I was working alone and didn't need to preserve the commit record
Pros: • Fully removes the commit and its changes • Keeps history clean

Cons: • Rewrites history • Dangerous on shared branches

I would use git revert instead if I was working in a team or on a shared branch, because it's safer and keeps the history intact by adding a new commit that undoes the previous one

5-Briefly explain the difference between git stash pop and git stash apply.

git stash apply applies the stashed changes but keeps them in the stash list, so you can reuse them again if needed

git stash pop also applies the stashed changes, but removes them from the stash list after applying

the repo link : https://github.com/Hager706/Konecta-tasks