Git repo: https://github.com/gimmeursocks/konecta-week-1

1) List of all commands used:

```
3101  git --version
3102  git init
3103  echo "# My First Git Project" > README.md
3104  git status
3105  git add .
3106  git commit -m "feat: Initial project setup and README"
3107  git remote add origin https://github.com/gimmeursocks/konecta-week-1.git
3108  git remote -c
3109  git remote -v
3110  git push
3111  git push -u origin main
3112  git push -u origin master
3113  git checkout -b feature/add-greeting
3114  git checkout -b main
3115  git push
3116  git push -u origin main
3117  git branch
3118  git branch -d master
3119  git push origin --delete master
3120  git checkout feature/add-greeting
3121  echo "This project will greet you >> README.md\n\nq
3122  echo "This project will greet you >> README.md
3123  echo "This project will greet you! " >> README.md
3124  echo 'print("Hello, Git!")' > greet.py
3125  git add .
3126  git commit -m "feat: Add simple greeting feature with Python script"
3127  git checkout main
3128  ls
3129  git merge feature/add-trackign
```

```
3130  git merge feature/add-tracking
3131  git merge feature/add-greeting
3132  git branch -d feature/add-greeting
3133  git push origin main
3134  echo "## Project Goals" >> README.md\necho "- Learn Git" >> README.md\n
3135  git add README.md\ngit commit -m "docs: Add project goals section"\n
3136  git push origin main
3137  git checkout -b conflict-feature
3138  git add .
3139  git commit -m "feat: Modify README"
3140  git amend
3141  git commit --amend
3142  git checkout main
3143  echo "- Collaborate Effectively" >> README.md
3144  git add .
3145  git commit -m "docs: Add collaboration goal to main"
3146  git merge conflict-merge
3147  git merge conflict-feature
3148  git add .
3149  git status
3150  git commit
3151  git branch -d conflict-feature
3152  git push origin main
3153  calculator.py
3154  git add calculator.py\ngit commit -m "feat: Add basic calculator functions (add, subtract)"\n
3155  git add calculator.py\ngit commit -m "feat: Add multiply function"\n
3156  git add calculator.py\ngit commit -m "fix: Optimize subtract function logic (introducing bug)"\n
3157  python calculator.py
3158  git add calculator.py\ngit commit -m "feat: Add divide function with zero check"\n
3159  git status
3160  git log --oneline
```

3161 git diff f3e4ed7~1 f3e4ed7 calculator.py

3162 git stash

3163 git revert f3e4ed7

3164 python calculator.py

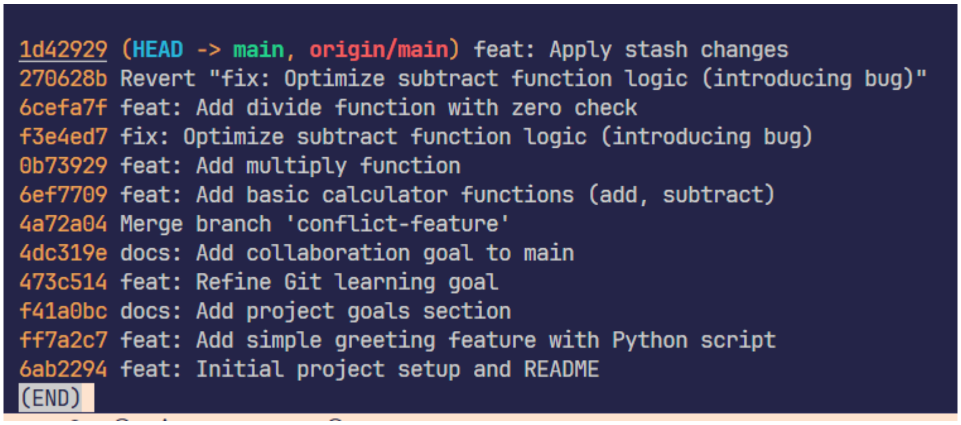
3165 git stash pop

3166 git push origin main

3167 history | tail -n 70

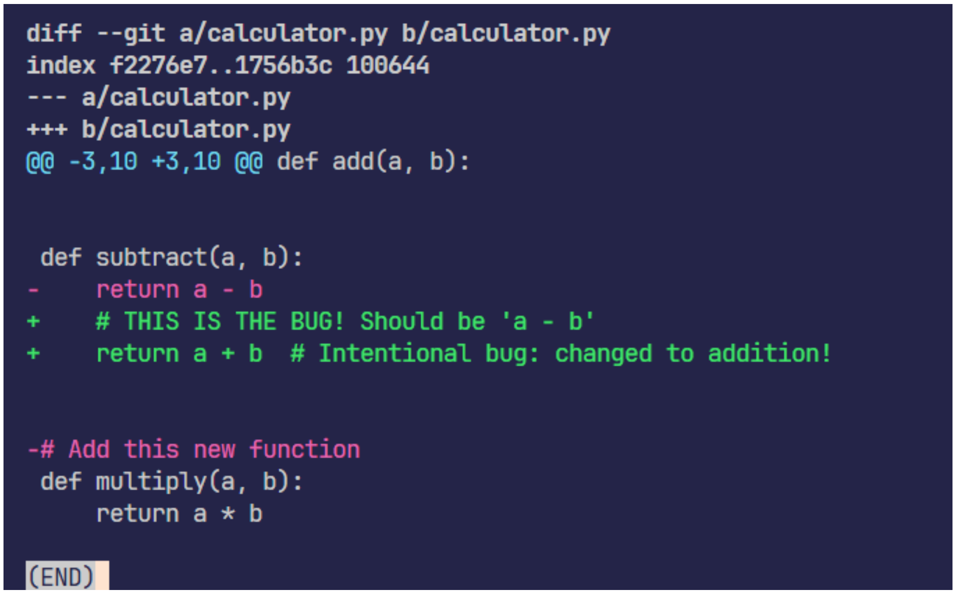
2) Screenshots after significant commands:

- git log –oneline



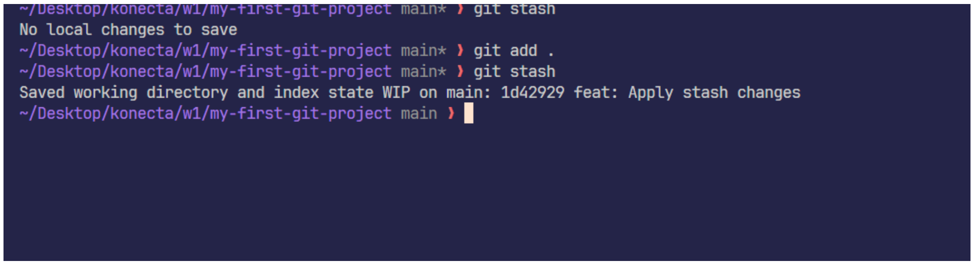
```
1d42929 (HEAD -> main, origin/main) feat: Apply stash changes
270628b Revert "fix: Optimize subtract function logic (introducing bug)"
6cefa7f feat: Add divide function with zero check
f3e4ed7 fix: Optimize subtract function logic (introducing bug)
0b73929 feat: Add multiply function
6ef7709 feat: Add basic calculator functions (add, subtract)
4a72a04 Merge branch 'conflict-feature'
4dc319e docs: Add collaboration goal to main
473c514 feat: Refine Git learning goal
f41a0bc docs: Add project goals section
ff7a2c7 feat: Add simple greeting feature with Python script
6ab2294 feat: Initial project setup and README
(END)
```

- git diff f3e4ed7~1 f3e4ed7 calculator.py



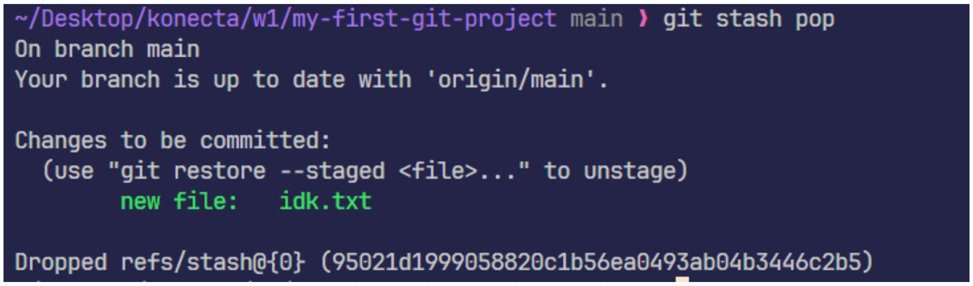
```
diff --git a/calculator.py b/calculator.py
index f2276e7..1756b3c 100644
--- a/calculator.py
+++ b/calculator.py
@@ -3,10 +3,10 @@ def add(a, b):


 def subtract(a, b):
-    return a - b
+    # THIS IS THE BUG! Should be 'a - b'
+    return a + b  # Intentional bug: changed to addition!


-# Add this new function
 def multiply(a, b):
     return a * b

(END)
```

- git stash



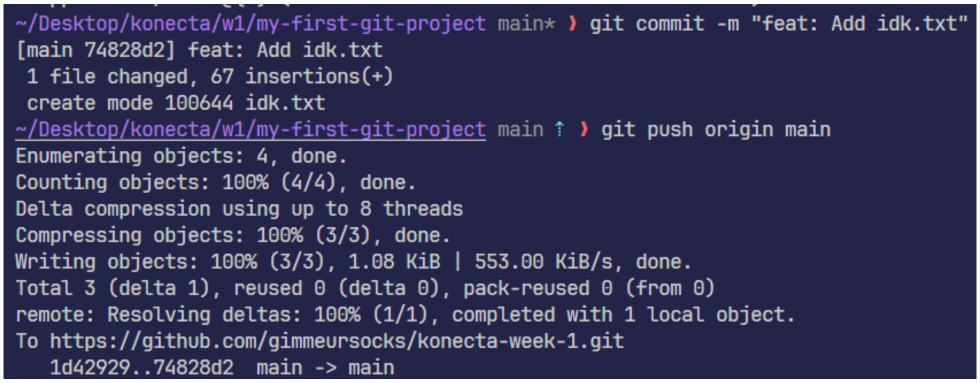
```
~/Desktop/konecta/w1/my-first-git-project main* ❯ git stash
No local changes to save
~/Desktop/konecta/w1/my-first-git-project main* ❯ git add .
~/Desktop/konecta/w1/my-first-git-project main* ❯ git stash
Saved working directory and index state WIP on main: 1d42929 feat: Apply stash changes
~/Desktop/konecta/w1/my-first-git-project main ❯
```

- git stash pop



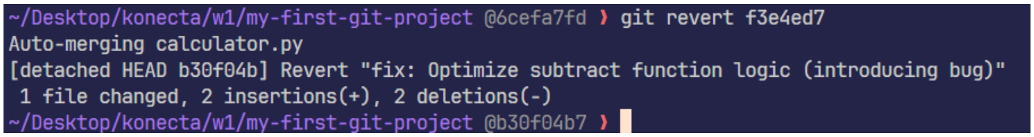
```
~/Desktop/konecta/w1/my-first-git-project main ❯ git stash pop
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   idk.txt

Dropped refs/stash@{0} (95021d1999058820c1b56ea0493ab04b3446c2b5)
```

- git commit and git push



```
~/Desktop/konecta/w1/my-first-git-project main* ❯ git commit -m "feat: Add idk.txt"
[main 74828d2] feat: Add idk.txt
 1 file changed, 67 insertions(+)
 create mode 100644 idk.txt
~/Desktop/konecta/w1/my-first-git-project main ⇡ ❯ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.08 KiB | 553.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/gimmeursocks/konecta-week-1.git
   1d42929..74828d2  main -> main
```

- git revert



```
~/Desktop/konecta/w1/my-first-git-project @6cefa7fd ❯ git revert f3e4ed7
Auto-merging calculator.py
[detached HEAD b30f04b] Revert "fix: Optimize subtract function logic (introducing bug)"
 1 file changed, 2 insertions(+), 2 deletions(-)
~/Desktop/konecta/w1/my-first-git-project @b30f04b7 ❯
```

3) Final Calculator.py file:

```
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


# Add this new function
def divide(a, b):
  if b == 0:
    return "Cannot divide by zero!"
  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)}")

# Some new experimental print statement
print("Experimental feature: Fibonacci sequence next")
```

4) Written explanations:

- How git log and git diff helped me identify the erroneous commit:

  Git log allowed me to view the commit history in order, by reviewing the commit messages I was able to identify the commit that introduced the issue.

  Git diff <commit_id>~1 <commit_id> allowed me to inspect the changes for the calculator.py file. Which helped confirm the root cause of the current bug.
- Why git stash was necessary?

  Because I currently had a bug in my code base and I had local changes that I did not want to commit for now, but still wanted a clean working directory to investigate the issue. Stashing saved my work temporarily so that I can inspect the code with losing the progress.
- Difference between git stash pop and git stash apply:

  Stash apply, applies the stashed changes to the working directory, but keeps the changes in the stash list. While stash pop applies the changes and removes that stash entry from the list.

Apply when you need to reuse the changes, and pop when you are done with the changes.

- Choice between git reset –hard and git revert:

  I chose git revert because rewriting the history is not safe and would break the history when multiple users are on the same repo.

  Pros of revert is that it creates a new commit and preserves history and is transparent and traceable. Cons of revert is that it does not remove the bad commit but just negates its effects, and that it can get messy when multiple commits are reverted.

  The only time where I would use git reset –hard, is when I'm in a private local repo where rewriting history is fine because I'm the only user, or if I want to completely remove one or more commits completely and reset the working directory and staging area.

  All in all, I used revert because it is the best practice in this scenario and wouldn't break the shared history. And that reset is only used experimentally in private branches.