

Assignment2

Submitted

by

Pankaj Kumar

Level 3: Resolving Merge Conflicts

You have been hired by the client ABC Systems as a Developer, where in you are required to manage the source in a Git Repository. Client would like to assess your hands on knowledge and skills. Thus, Client has requested you to demonstrate the following actions using Git and GitHub.

You are requested to execute following task:

1. Create a file Demo.java which prints "Hello World".
2. Create 2 branches out of this master. Feature1 and Feature2.
3. Make changes in feature 1 branch in the same line of code.
4. Commit the code in feature1 branch.
5. Now checkout the feature2 branch and make a change in the same line as you did in feature1.
6. Commit the code in feature 2 branch
7. Merge the Feature2 branch in Feature1 branch. It should give a merge conflict.
8. Resolve the merge conflict with the help of Kdiff3 or Meld tool.
9. Verify the merge using git log command.

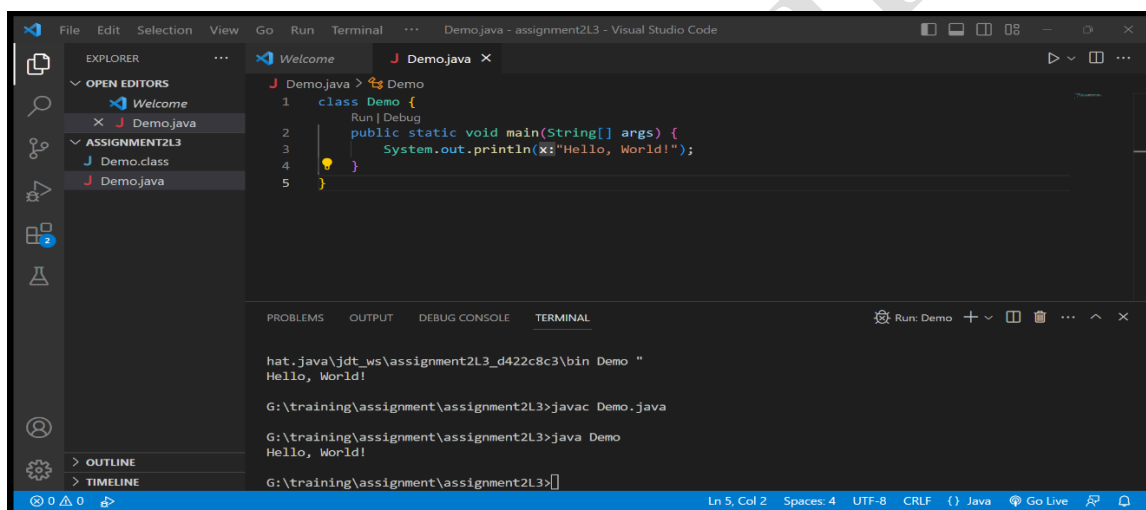
Step1: Create a file Demo.java which prints "Hello World".

- Create a file Demo.java which prints "Hello World".
- To create a file called Demo.java and print "Hello World" in it, you can use the following commands:

Code .

Demo.java

```
Class Demo {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```



Step2: - Create 2 branches out of this master. Feature1 and Feature2.

- To create two branches called Feature1 and Feature2 from the master branch, you can use the following commands:
 - **\$ git branch feature1**
 - **\$ git branch feature2**
- The first command creates a new branch called Feature1, while the second command creates a new branch called Feature2.

```
C:\Windows\System32\cmd.exe
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 637 bytes | 212.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Pankajch00/Assignment2L3.git
 * [new branch]      master -> master
branch 'master' set up to track 'master/master'.

G:\training\assignment\assignment2L3>git status
On branch master
Your branch is up to date with 'master/master'.

nothing to commit, working tree clean

G:\training\assignment\assignment2L3>git branch Feature1
G:\training\assignment\assignment2L3>git branch
  Feature1
 * master

G:\training\assignment\assignment2L3>git branch Feature2
G:\training\assignment\assignment2L3>git branch
  Feature1
  Feature2
 * master

G:\training\assignment\assignment2L3>
```

Step3: - Make changes in feature 1 branch in the same line of code.

- To make changes in the Feature1 branch, you can edit the Demo.java file using VScode editor and change the message from "Hello World" to something else, By using the following command:
- **\$ git checkout feature1**
- **\$ git branch**
- **\$ code .**
- **Make change in code.**
- **\$ javac Demo.java**
- **\$ java demo**

```
C:\Windows\System32\cmd.exe
  Feature1
  Feature2
 * master

G:\training\assignment\assignment2L3>git checkout Feature1
Switched to branch 'Feature1'

G:\training\assignment\assignment2L3>git branch
 * Feature1
  Feature2
  master

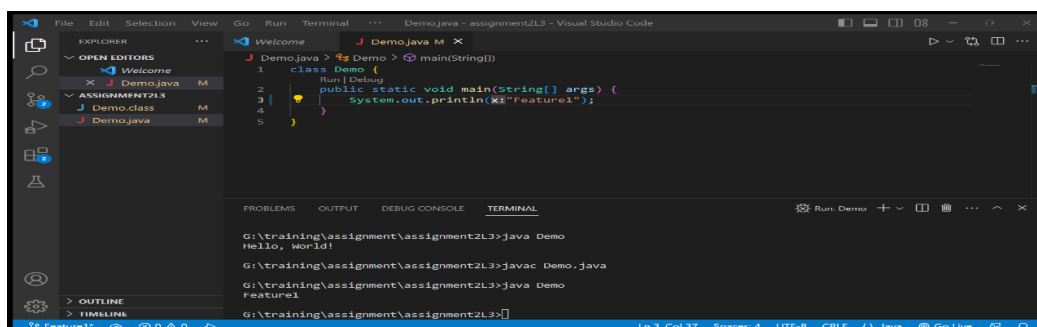
G:\training\assignment\assignment2L3>code .

G:\training\assignment\assignment2L3>git status
On branch Feature1
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:   Demo.class
        modified:   Demo.java

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

G:\training\assignment\assignment2L3>git commit -a -m "this is 2nd change in feature1"
[Feature1 6daf646] this is 2nd change in feature1
2 files changed, 1 insertion(+), 1 deletion(-)

G:\training\assignment\assignment2L3>git status
On branch Feature1
nothing to commit, working tree clean
```



Step4: - Commit the code in feature1 branch.

- **\$ git status**
- **\$git commit -a -m “this is 2nd change in feater2”**

```
C:\Windows\System32\cmd.exe
G:\training\assignment\assignment2L3>git checkout Feature1
Switched to branch 'Feature1'

G:\training\assignment\assignment2L3>git branch
* Feature1
  Feature2
  master

G:\training\assignment\assignment2L3>code .

G:\training\assignment\assignment2L3>git status
On branch Feature1
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:   Demo.class
        modified:   Demo.java

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

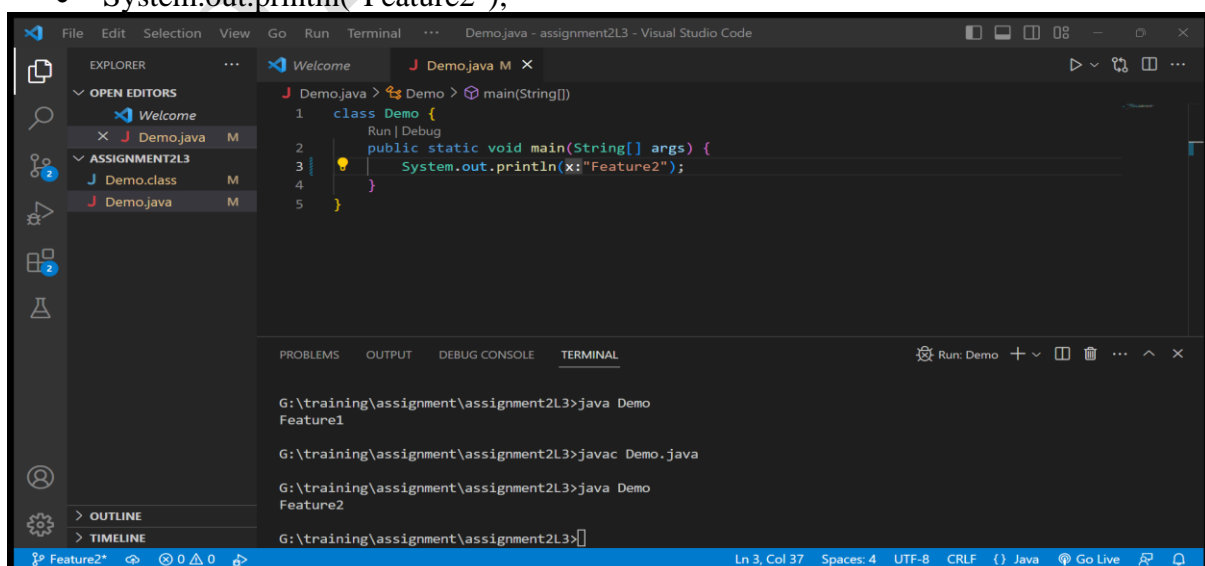
G:\training\assignment\assignment2L3>git commit -a -m "this is 2nd change in feature1"
[Feature1 6daf646] this is 2nd change in feature1
 2 files changed, 1 insertion(+), 1 deletion(-)

G:\training\assignment\assignment2L3>git status
On branch Feature1
nothing to commit, working tree clean

G:\training\assignment\assignment2L3>git push -u Feature1
fatal: 'Feature1' does not appear to be a git repository
fatal: Could not read from remote repository.
```

Step5: - Now checkout the feature2 branch and make a change in the same line as you did in feature1.

- To switch to the Feature2 branch and make a change in the same line as you did in Feature1, you can use the following commands:
- **\$git checkout Feature2**
- `System.out.println("Feature2");`



Step 6: Commit the code in feature 2 branch.

- **\$git status**
- **\$git commit -a -m “this is 3rd change in feature2”**

```
C:\Windows\System32\cmd.exe
branch 'Feature1' set up to track 'Feature1-server/Feature1'.

G:\training\assignment\assignment2L3>git checkout Feature2
Switched to branch 'Feature2'

G:\training\assignment\assignment2L3>git branch
  Feature1
* Feature2
  master

G:\training\assignment\assignment2L3>git status
On branch Feature2
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:   Demo.class
        modified:   Demo.java

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

G:\training\assignment\assignment2L3>git commit -a -m "this is 3rd change in Fearture2 branch"
[Feature2 fec47e9] this is 3rd change in Fearture2 branch
 2 files changed, 1 insertion(+), 1 deletion(-)

G:\training\assignment\assignment2L3>git status
On branch Feature2
nothing to commit, working tree clean

G:\training\assignment\assignment2L3>
```

Step 7: Merge the Feature2 branch in Feature1 branch. It should give a merge conflict.

- **\$git merge Feartue1**

```
C:\Windows\System32\cmd.exe
master

G:\training\assignment\assignment2L3>git status
On branch Feature2
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:   Demo.class
        modified:   Demo.java

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

G:\training\assignment\assignment2L3>git commit -a -m "this is 3rd change in Fearture2 branch"
[Feature2 fec47e9] this is 3rd change in Fearture2 branch
 2 files changed, 1 insertion(+), 1 deletion(-)

G:\training\assignment\assignment2L3>git status
On branch Feature2
nothing to commit, working tree clean

G:\training\assignment\assignment2L3>git merge Feature1
warning: Cannot merge binary files: Demo.class (HEAD vs. Feature1)
Auto-merging Demo.class
CONFLICT (content): Merge conflict in Demo.class
Auto-merging Demo.java
CONFLICT (content): Merge conflict in Demo.java
Automatic merge failed; fix conflicts and then commit the result.

G:\training\assignment\assignment2L3>
```

Step 8: Resolve the merge conflict with the help of Kdiff3 or Meld tool.

- **\$ git mergetool -t kdiff3**

```
C:\Windows\System32\cmd.exe

G:\training\assignment\assignment2L3>git status
On branch Feature2
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
  modified:   Demo.java

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   Demo.class

G:\training\assignment\assignment2L3>git add .
G:\training\assignment\assignment2L3>git commit -m "mearge feature1 and feature2"
[Feature2 c3f3205] mearge feature1 and feature2

G:\training\assignment\assignment2L3>git status
On branch Feature2
nothing to commit, working tree clean

G:\training\assignment\assignment2L3>git mergetool -t kdiff3
No files need merging

G:\training\assignment\assignment2L3>git add remote Feature2-server https://github.com/Pankajchy00/Assignment2L3.git
fatal: pathspec 'remote' did not match any files
```

Step 9: Verify the merge using git log command.

- \$git log

```
C:\Windows\System32\cmd.exe

G:\training\assignment\assignment2L3>git log
commit c3f320569c09ef8ddc4e9e6344ac1719f9f682e6 (HEAD -> Feature2, Feature2-server/Feature2)
Merge: fec47e9 6daf646
Author: Pankajchy00 <pankajchy00@gmail.com>
Date:   Fri Apr 14 18:04:55 2023 +0530

    mearge feature1 and feature2

commit fec47e9b7d5e945bbe7519d8db3ef5a8aa9909f4
Author: Pankajchy00 <pankajchy00@gmail.com>
Date:   Fri Apr 14 17:56:27 2023 +0530

    this is 3rd change in Fearture2 branch

commit 6daf64601bdc96b5c5de635f3b748a9473cfc26b (Feature1-server/Feature1, Feature1)
Author: Pankajchy00 <pankajchy00@gmail.com>
Date:   Fri Apr 14 17:32:08 2023 +0530

    this is 2nd cange in feature1

commit e4defb5536fce66db65c4f991f4a14b02896f896 (master/master, master)
Author: Pankajchy00 <pankajchy00@gmail.com>
Date:   Fri Apr 14 13:36:47 2023 +0530

    rum hello world

G:\training\assignment\assignment2L3>
```

- Github [url:- https://github.com/Pankajchy00/Assignment2L3.git](https://github.com/Pankajchy00/Assignment2L3.git)

Pankaj Kumar