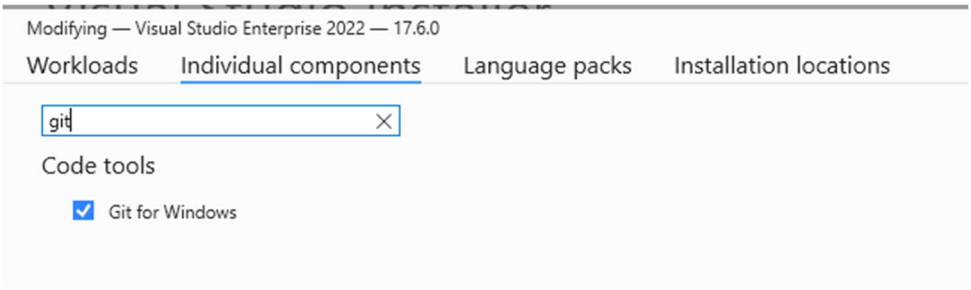
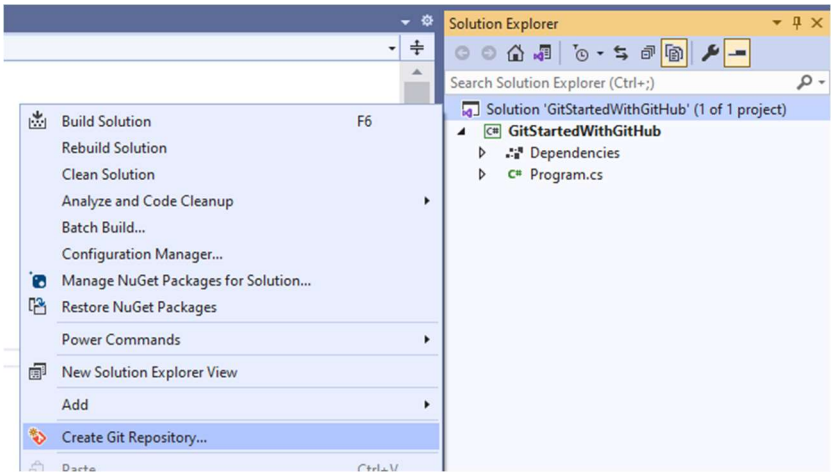
Getting Started with GitHub

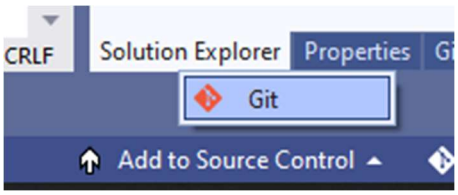
Episode 1: Using Source Control

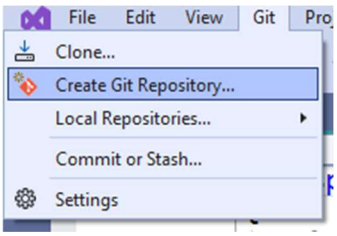
1. In Visual Studio Installer, click **Modify**.
2. On Individual Component tab, search for **Git**.



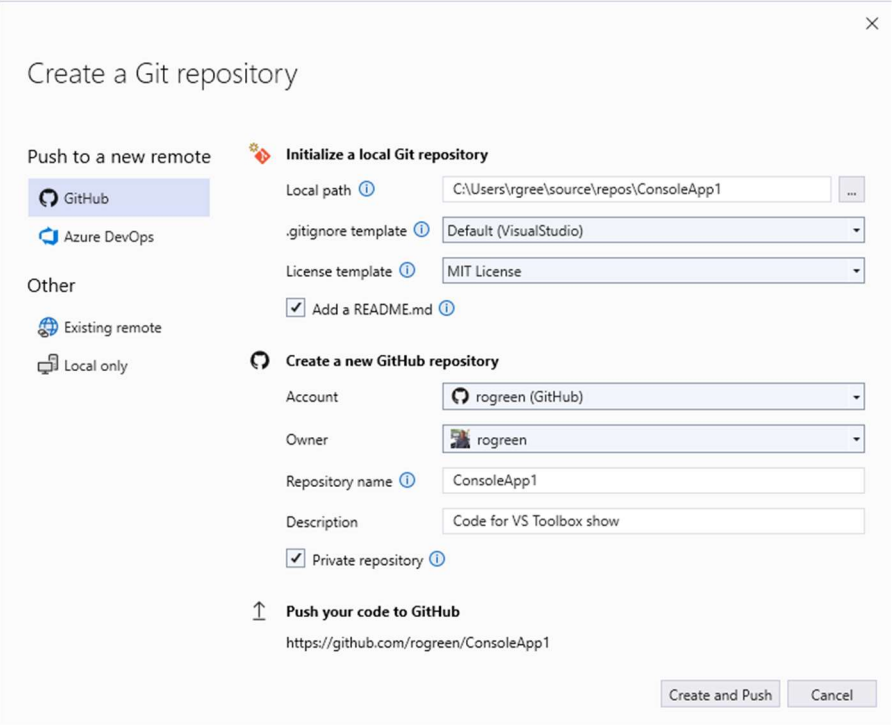
3. In Visual Studio, create Console app.
4. Review why you want to use source control:
    a. Backup of code
    b. Work on multiple machines
    c. Support multiple devs working on code
    d. Share code with others
5. Right-click solution and select **Create Git Repository** or select **Add to Source Control | Git** or select **Git | Create Git Repository**.



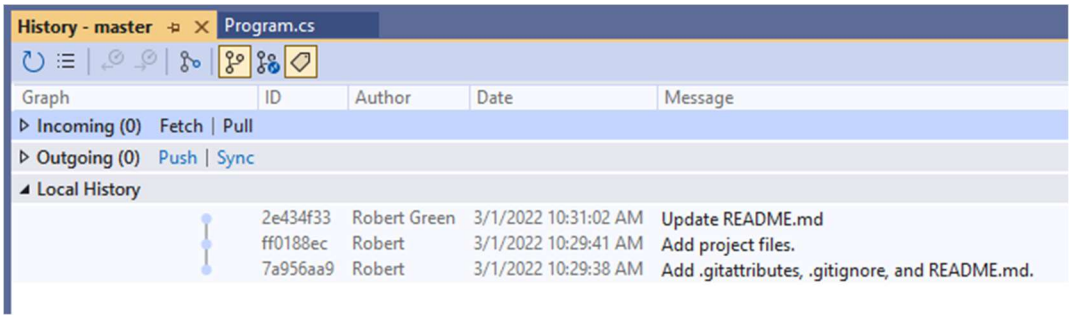




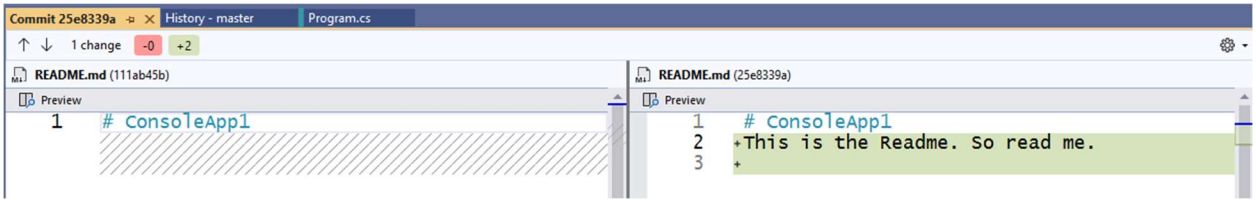
6. Fill out Create a Git Repository dialog.



7. Click **Create and Push**.
8. In GitHub, show the repo now exists.
9. Open .gitignore, license.txt and readme.md.
10. Update the Readme and commit.
11. In VS, select **Git | Open in File Explorer**.
12. Show .git folder, .gitattributes, .gitignore, readme.md.
13. Notice the readme is the original one.
14. In VS, select **Git | Pull**.
15. Notice the readme is now the current one.
16. Select **Git | View Branch History**.



17. Double click last commit to see what changes were made.



18. Close the solution.
19. Select **Git | Clone**.
20. In GitHub, go to main repo page.
21. Click **Code** and copy the URL.
22. Paste it back in Visual Studio.
23. Click **Clone**.
24. Select **Git | Open in File Explorer**.

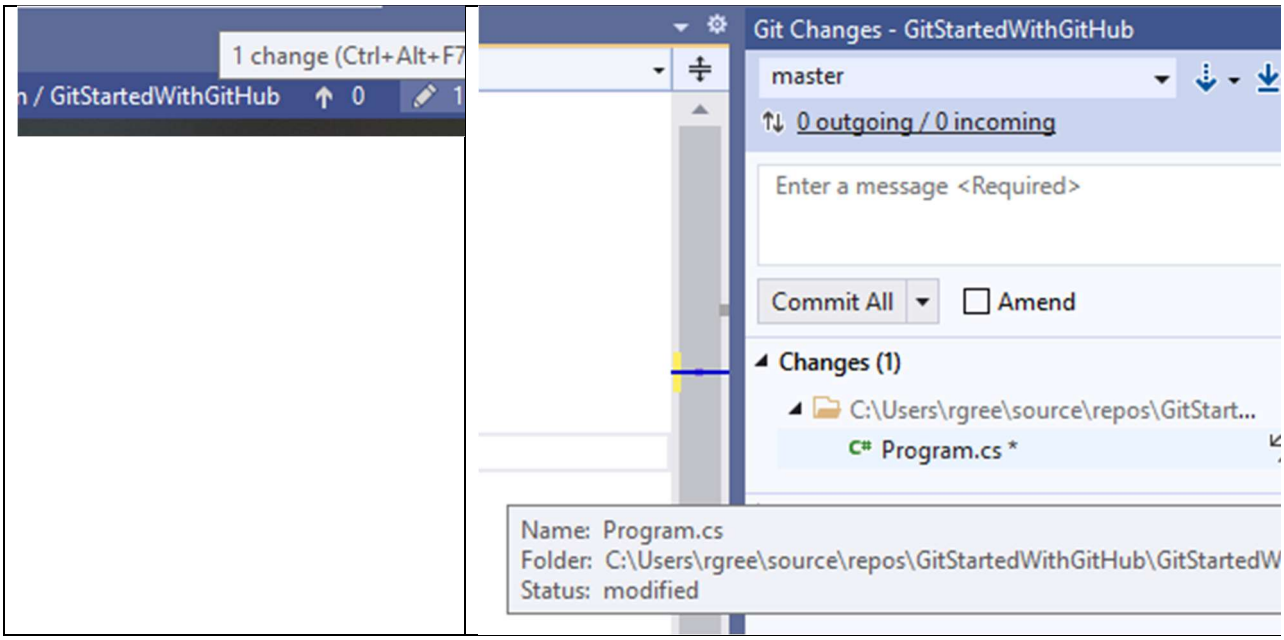
END

Episode 2: Committing Code Changes

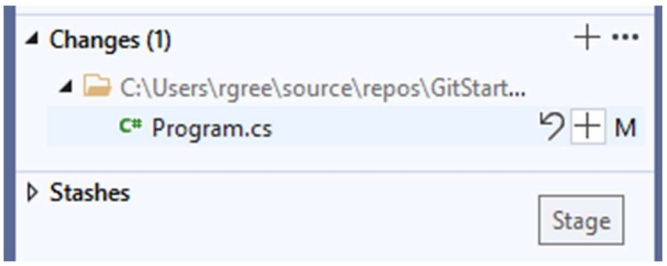
1. Make the following changes in Program.cs.

```
Console.WriteLine("Hello, Visual Studio Toolbox!");
```

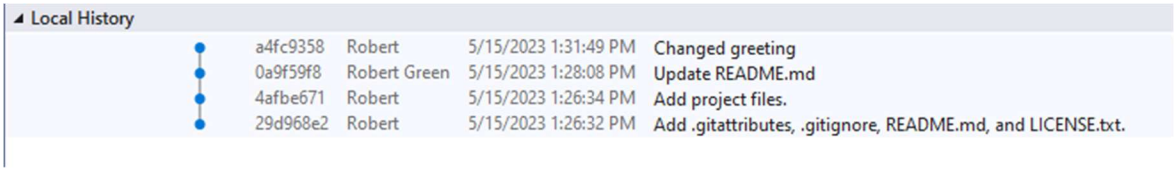
2. See that you have pending changes.



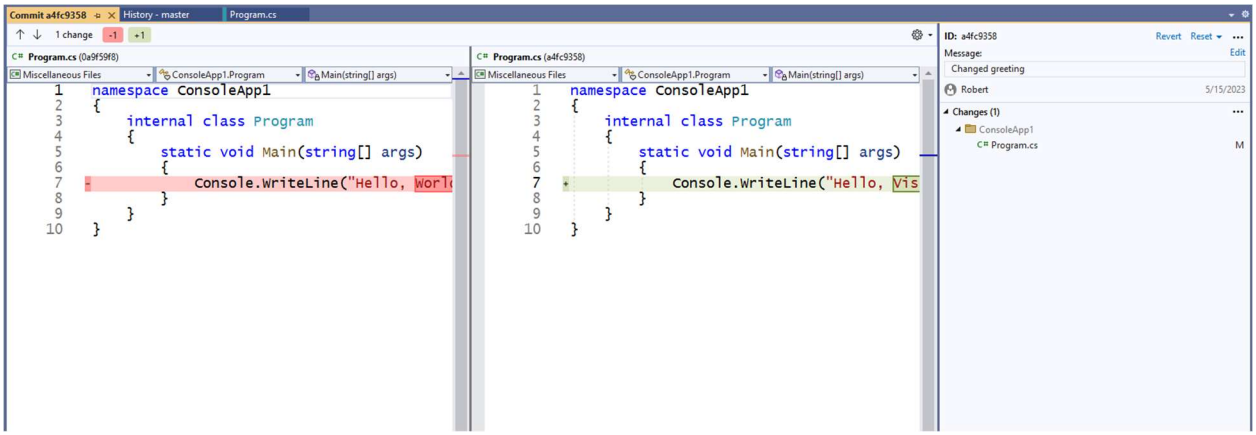
3. See that you can Undo changes and Stage/Stash changes.



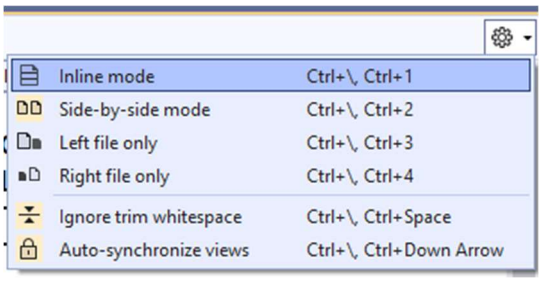
4. Enter a commit message and click **Commit All and Push**.
5. Show the changes are now in GitHub.
6. Click the commit message to see the commit.
7. In VS, view the branch history and see the commit.



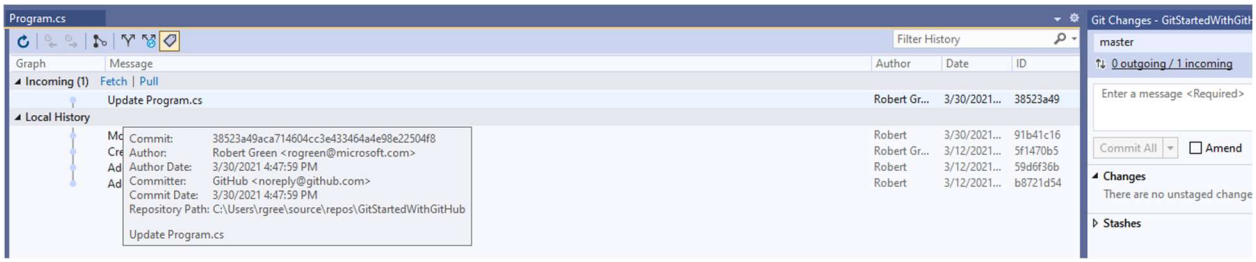
8. Double click the commit to see before and after. Notice that you can Revert.



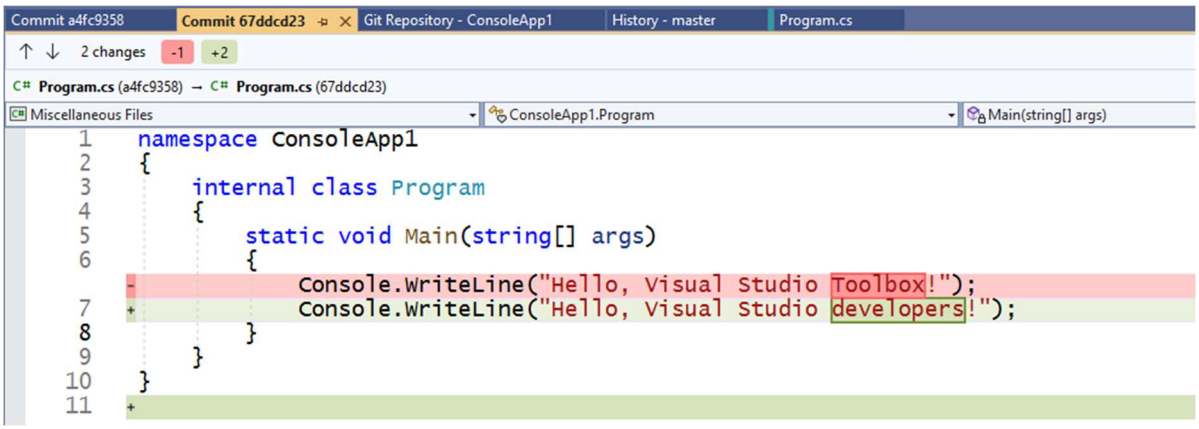
9. You can switch from side by side to inline mode if you prefer.



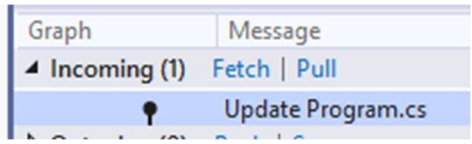
10. In GitHub change the greeting in Program.cs and Commit. Enter text into Extended Description.
11. Switch to VS.
12. Fetch. Master does not show the change.
13. Click on 0 outgoing / 1 incoming and expand Incoming.



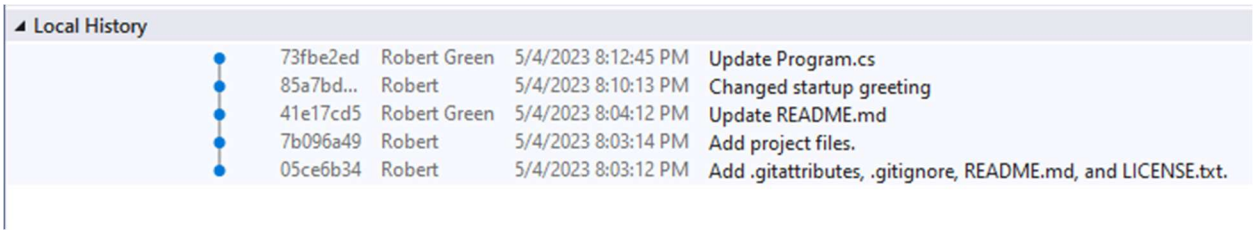
14. Double click Update Program.cs to compare the two versions.



15. Click **Pull**.



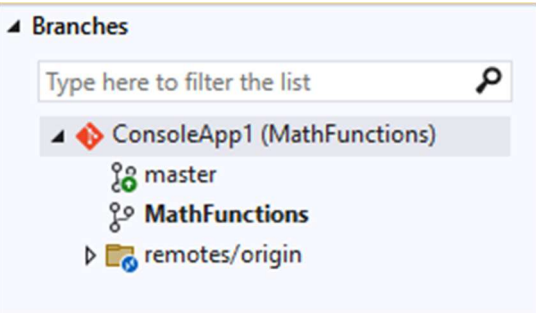
16. Refresh the branch history to see both changes to Program.cs (one in VS and one in GitHub).



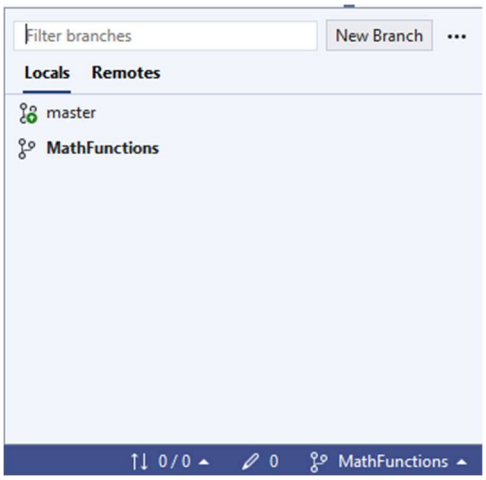
END

Episode 3: Working in branches

1. Select **Git | New Branch** or select **Git | Manage Branches** and then right-click main on master and select **New Local Branch From**
2. Enter **MathFunctions** as the branch name and **Create**.
3. Select **Git | Manage Branches**. See that MathFunctions is the current branch.



4. You can also see this in the status bar.

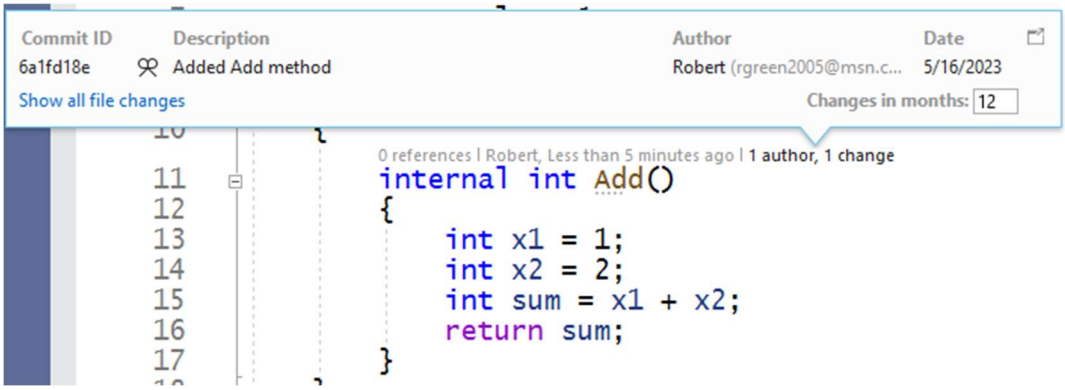


5. Add **Math** class.
6. Add the following method

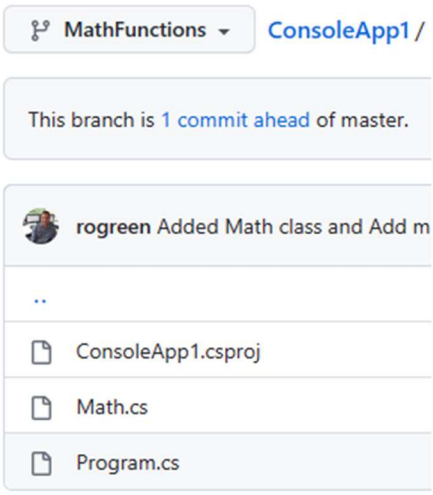
```
internal int Add()
{
    int x1 = 1;
    int x2 = 2;
    sum = x1 + x2;
```

```
        return sum;
    }
```

7. Commit and push. Notice that you pushed the branch.
8. You can see what happened in the CodeTips.

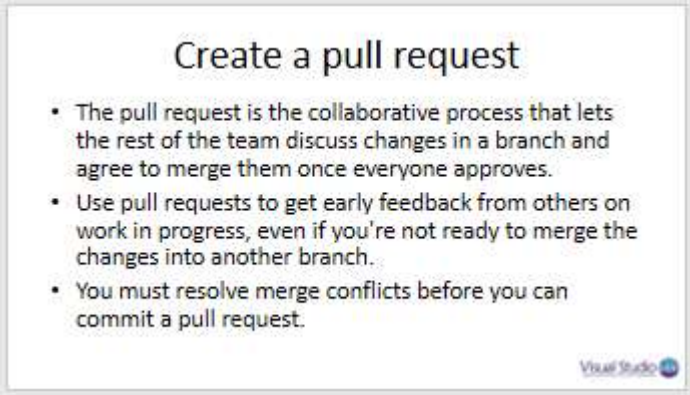


9. In GitHub, see that the master branch does not have your new code.
10. Switch to the MathFunctions branch and see your code.

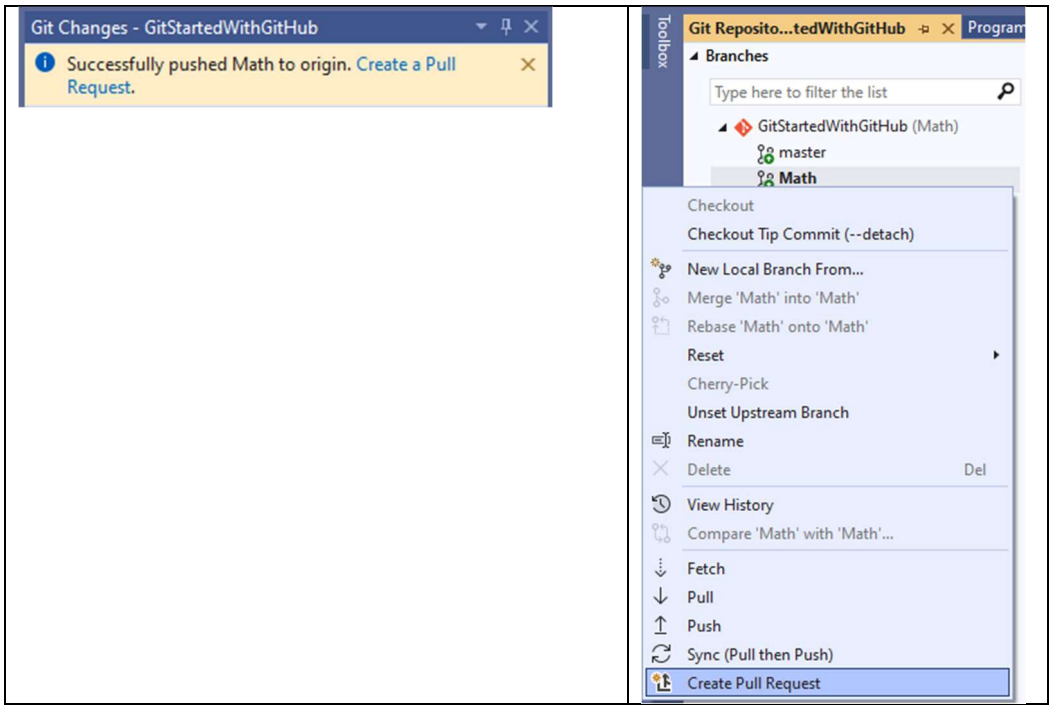


11. In VS, switch to Master branch and see that the changes disappear from VS.
12. Switch back to MathFunctions.

Create a pull request

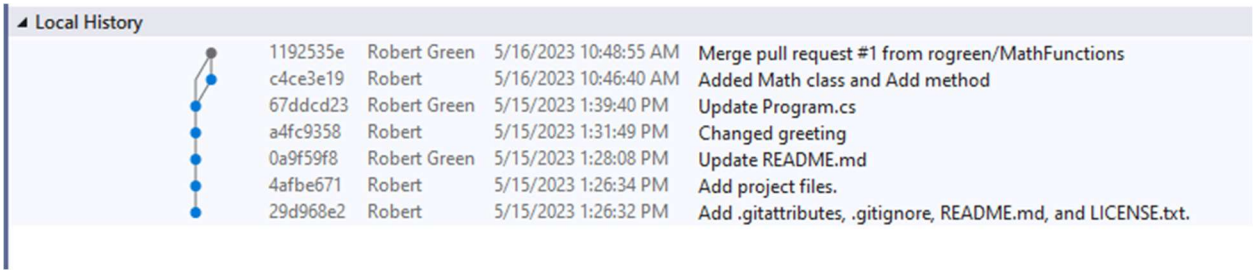


1. In VS, click Create a Pull Request



2. In GitHub you can request a review from or assign this to someone

3. Click **Create pull request**.
4. Click **Merge pull request** and then **Confirm merge**. Notice you can delete the branch.
5. Show that master branch now has the changes.
6. In VS, change the branch to master. Notice changes aren't there.
7. Select **Git | Pull**. Your local copy is now in sync.



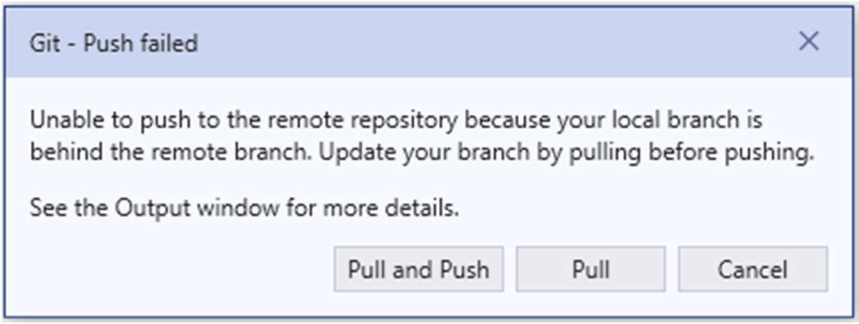
END

Episode 4: Resolving merge conflicts

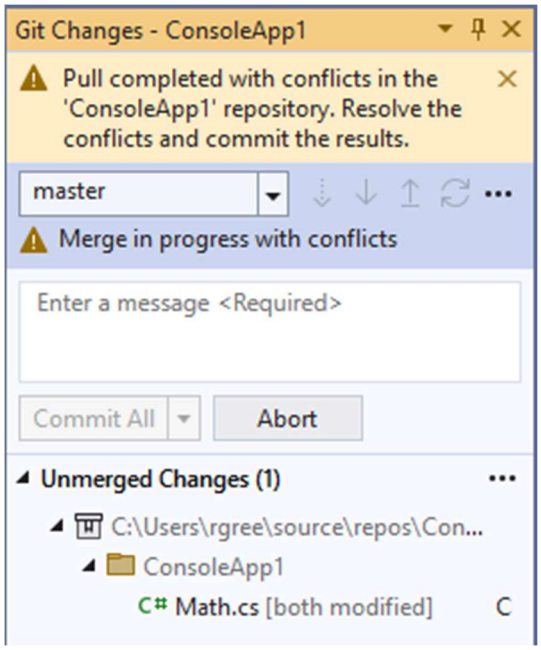
1. In GitHub, make sure master is the current branch.
2. Change int to var. Commit
3. In VS, make the following change:

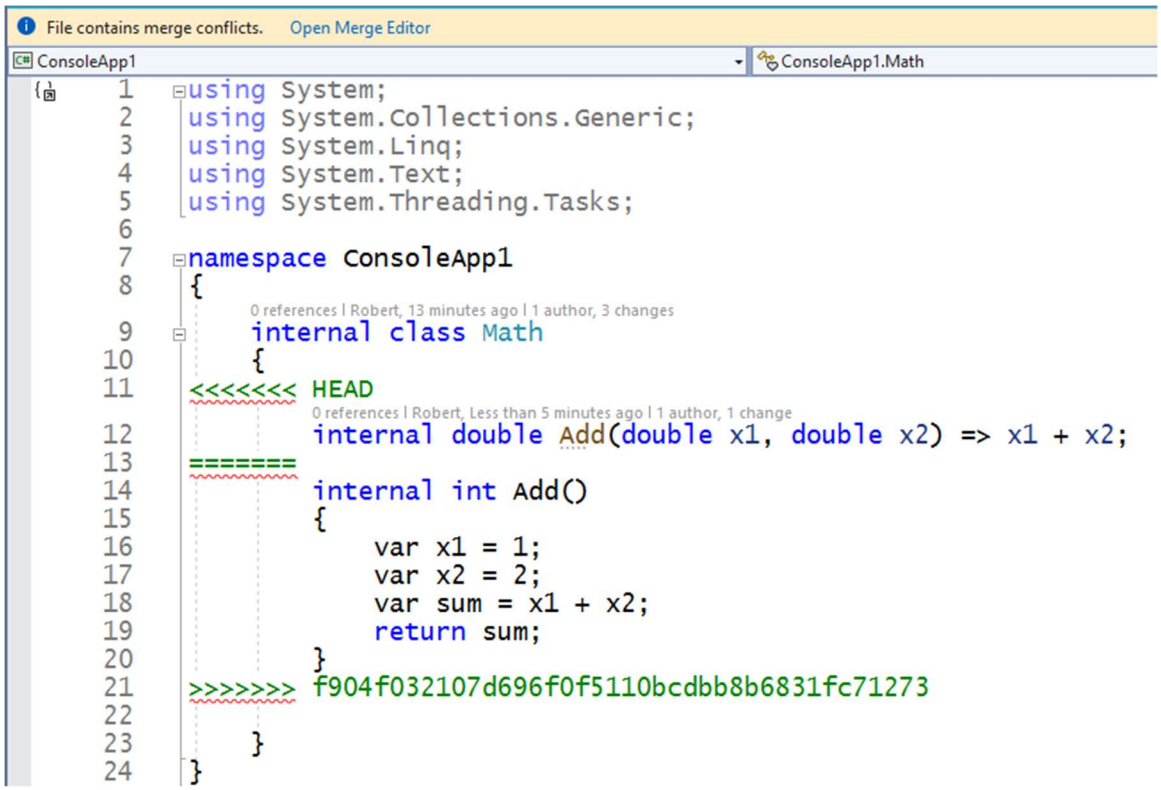
```
internal double Add(double  x1, double  x2) => x1 + x2;
```

4. Commit and push
5. Push fails.

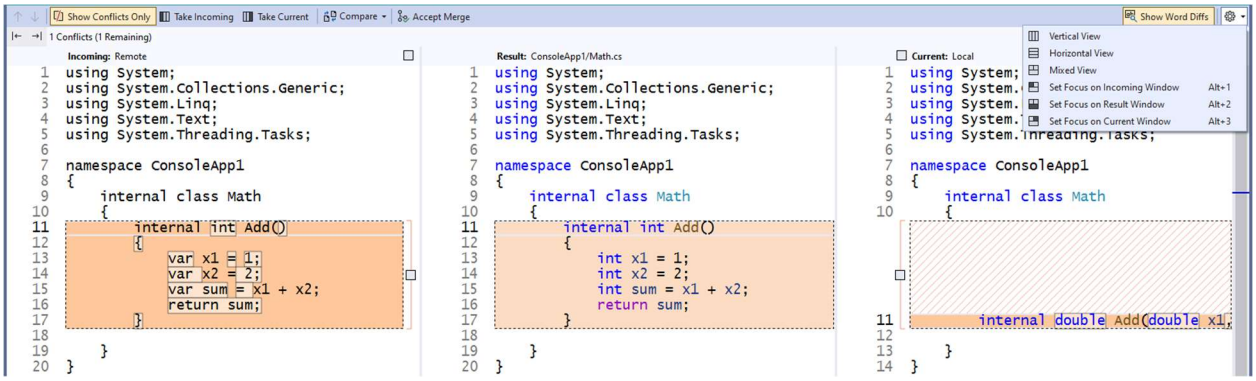


6. Click **Pull**.

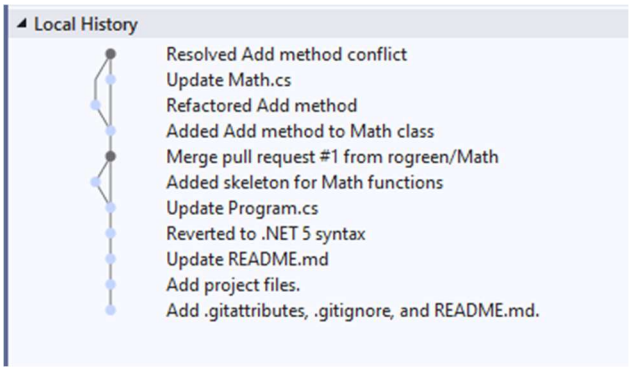




7. Click **Open Merge Editor**.
8. Compare Files



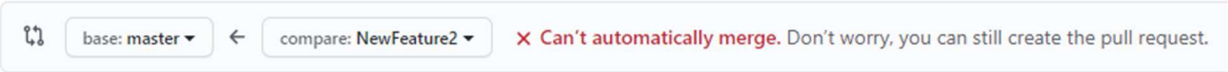
9. Click **Take Current** or use check boxes.
10. Click **Accept Merge**.
11. Commit and Push.
12. Confirm that GitHub now has refactored Add method.



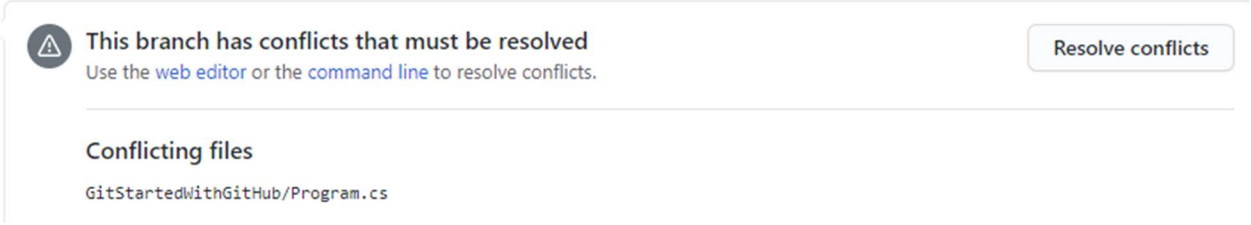
END

Episode 5: More on merge conflicts

1. Select **Git | New Branch**.
2. Enter **Utilities** as the branch name and **Create**.
3. Add a **Utilities** class.
4. Add code to call Utilities in Main.
5. Commit and push.
6. In GitHub add a Services class and add code to call Services in Main. Commit.
7. Create a pull request in VS.
8. Notice that you are warned there are conflicts



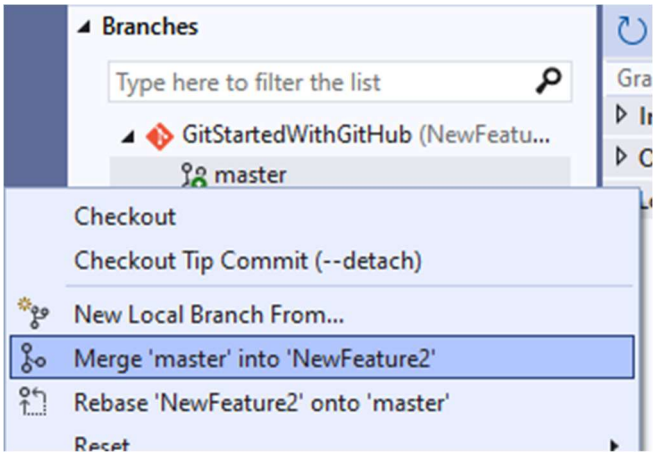
9. Click **Create pull request**.
10. You are alerted to conflicts.

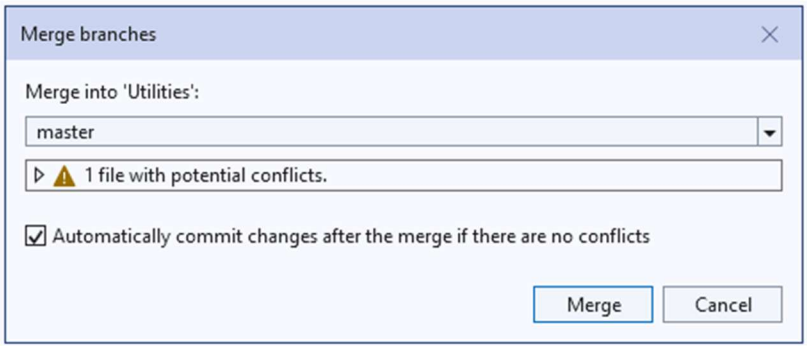


11. Click **Resolve conflicts** to see the problem.

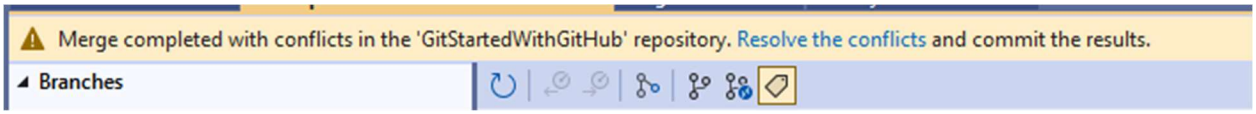
```
<<<<<<< NewFeature4
            // Add code here to call NewFeature2

            // Add code here to call NewFeature4
=======
            // Add different code here to call NewFeature2
>>>>>>> master
        }
```

12. In VS, make master the current branch.
13. Pull
14. Make Utilities the current branch.
15. In Branches dialog, right click master and select **Merge master into Utilities**.

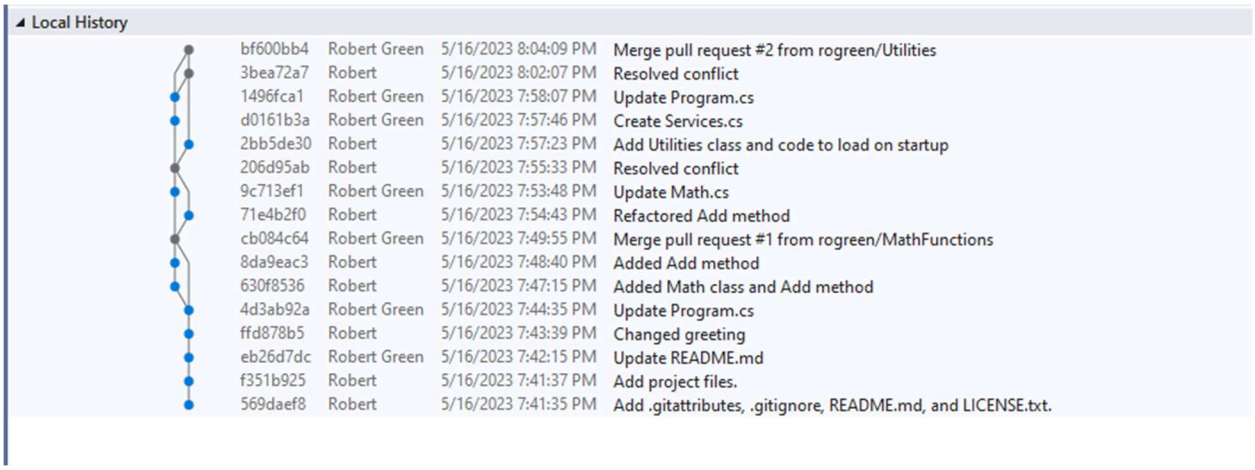




16. Click **Merge**.



17. Click **Resolve the conflicts**.
18. In the Git Changes window, double click **Program.cs** to see the conflict.
19. Click **Open Merge Editor**.
20. Select both lines of code. Note that the order you click controls what order they are added.
21. Click **Accept Merge**.
22. Commit Staged and Push.
23. Go to Pull Requests in GitHub and click **Merge pull request**.
24. Click **Confirm merge**.
25. Back in VS change branch to master and pull.
26. Confirm Program in master calls both Utilities and Services.



END