Git Exercises

* The following exercises should be performed using Visual Studio instead of the Git command line
* We will use GitHub for the exercises (since you might not have access to VSTS yet)

# Create a GitHub account (if needed)

1. Go to <http://www.github.com>
2. Create an account if you don’t have one

# Get added to MicrosoftLeap GitHub team

1. Send your github username to the instructor so they can add you to <https://github.com/MicrosoftLeap>
2. You’ll get an invitation e-mail, accept it to be added to the team

# Cloning a repository

1. Browse to <https://github.com/MicrosoftLeap/leap>
2. Using Visual Studio, clone the repository on this page
3. You should now have a local copy of the master branch

# Creating a new local branch

1. Create a new local branch as a copy of local master
2. Name it whatever you like, something like “<yourAlias>-dev” or “<yourName>-dev” if you don’t have an alias yet
3. Check out your new branch

# Push your local branch

1. Push your local branch to the remote repo
2. Your local branch now has a tracked branch

# Add/Modify files

1. Browse to your repo directory in Windows Explorer using Visual Studio
2. Add a new text file
3. Back in Visual Studio, open the HelloWorld solution
4. Delete the App.config
5. Modify Program.cs
6. Click on Save All
7. View your pending changes in Visual Studio
8. Diff the modified file against the unchanged version

# Staging/Committing changes

1. Stage the file you added and Program.cs
2. Commit your staged changes into your local repo
3. Undo the delete and the HelloWorld.csproj changes

# Syncing with a tracked branch

1. Sync your changes so they get pushed to the remote branch

# Pull requests

1. Working with a partner, create a pull request from your remote branch into theirs
2. Add your partner as a reviewer of the PR
3. Have your partner provide some actionable feedback
4. Modify your files locally to respond to the feedback and push those changes into your remote branch
5. Your partner should see the change in the PR and should approve the PR
6. Proceed with the PR and merge the branches

# Pull request merge conflicts

1. Working with a partner, both of you change the same local file in the same place and commit it.
2. Both of you push the change to your personal remote branch
3. One of you create a pull request from your remote branch into your partner’s remote branch.
4. Resolve this conflict locally, commit the resolve, and push it. The pull request should update, and the conflict should go away. Complete the pull request.
5. Perform this exercise again with the other person resolving the merge conflict locally.