

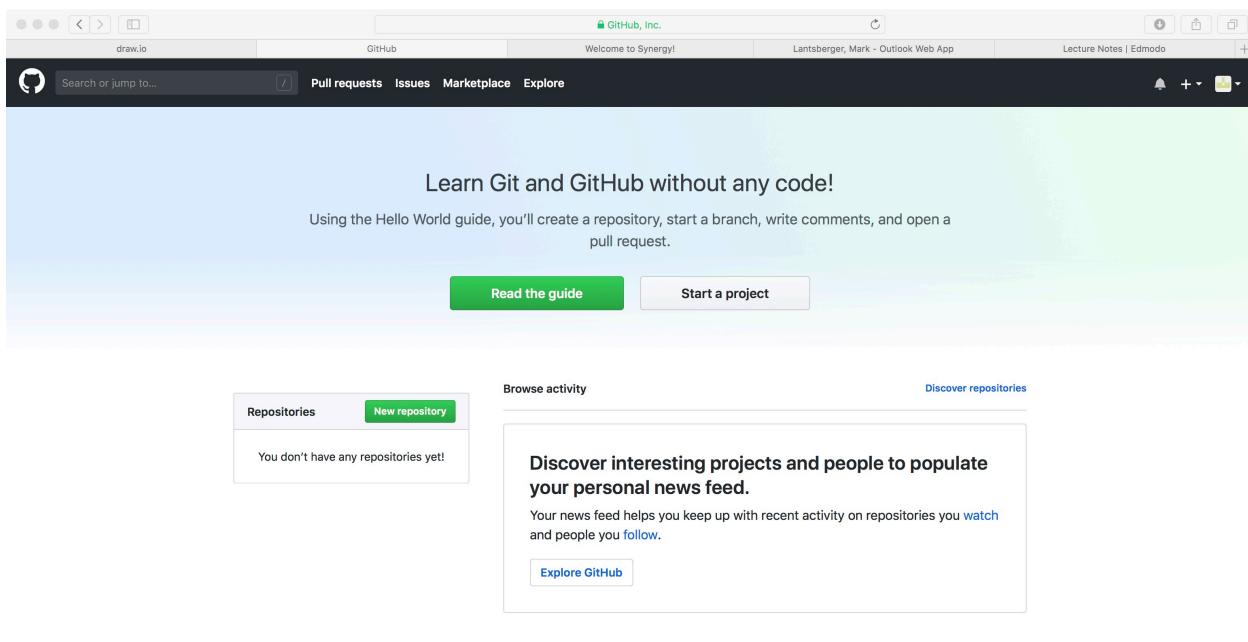
APCS

Lesson 6B

Assignment Labs

Accepting an Assignment and why you would do that

- With GitHub, I'm able to send you an assignment which is a repo with some starter files and code. In this system, you and I can communicate through GitHub and get your labs scored over the internet. Starting with Lab 6.1 MPG this is a requirement for your lab work and will be the official way your labs will be submitted and scored. **The color of the screenshot notes indicate the order of the steps. First - red, Second - green, Third - blue**
- Log on to github.com and you'll see this page (once you've created your account). You'll likely have some repos since you were supposed to practice with git already. If not, your screen will look like this:



© 2018 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)

[Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

- Next, log on to your Edmodo account. I'll begin posting links to our lab assignments.

The screenshot shows the Edmodo platform. On the left, there's a sidebar with class categories like 'APCS 4th', 'APCS 3rd', 'CS I 2nd', 'CS I 1st', and 'L Homeroom'. Below that are sections for 'Manage Classes', 'Create a Class', and 'Join a Class'. Another sidebar on the left lists 'Groups', 'Manage Groups', 'Create a Group', and 'Join a Group'. The main area is titled 'Discussions' and shows a post from 'Mr. Lantsberger' with the subject 'Lecture Notes'. The post includes the text 'Here's the link for Lab 6.1 MPG' and a link to 'classroom.github.com'. A large red arrow points to this link, and the text 'Click on this link' is overlaid in red. The GitHub logo is prominently displayed in the center of the screen.

- The link will bring you back to GitHub and you'll be asked to accept the assignment. Once you do so, you'll see some processing screens and then end up on a page which looks like this:

The screenshot shows the GitHub Classroom interface for the assignment 'Lab6.1 MPG'. The top navigation bar includes 'draw.io', 'GitHub', 'Welcome to Synergy!', 'Lantsberger, Mark - Outlook Web App', 'Lecture Notes | Edmodo', and 'GitHub Classroom'. The main content area is titled 'GitHub Classroom' and shows the course 'CS1-2nd-2017' and the user '@CS1-2nd-2017'. It displays a message: 'Accepted the Lab6.1 MPG assignment' and 'You are ready to go!'. Below this, it says 'You may receive an invitation to join @CS1-2nd-2017 via email invitation on your behalf. No further action is necessary.' and 'Your assignment has been created here: <https://github.com/CS1-2nd-2017/lab6-1-MPG-bassgruv>'. A large red arrow points to this GitHub link, which is circled in red. The text 'Clicking this link will take you to a new private repo which has been added to your account.' is overlaid in red at the bottom of the circled area.

- Following the above link will bring you to your newly created repo:

The screenshot shows a GitHub repository page for 'CS1-2nd-2017 / lab-6-1-mpg-bassgruv'. The repository was created by GitHub Classroom. It has 2 commits, 1 branch, 0 releases, and 1 contributor. A green circle highlights the 'Branch: master' dropdown menu. A red arrow points from the text 'Here is the button to press to create a branch' to the 'New pull request' button. A red box highlights the 'New pull request' button. A red annotation box contains the text: 'All these starter files are here for you to work in; including the .pdf of the lab specification. Sometimes I may include the spec in repo README shown below here as well.' Below the annotation box is the repository's README.txt content.

```

Branch: master ▾ New pull request
bassgruv Added spec.pdf
Car.java Initial sharing of project 11 days ago
L.A.6.1 - MPG.pdf Added spec.pdf 19 minutes ago
MPGDriver.java Initial sharing of project 11 days ago
README.TXT Initial sharing of project 11 days ago
package.bluej Initial sharing of project 11 days ago

```

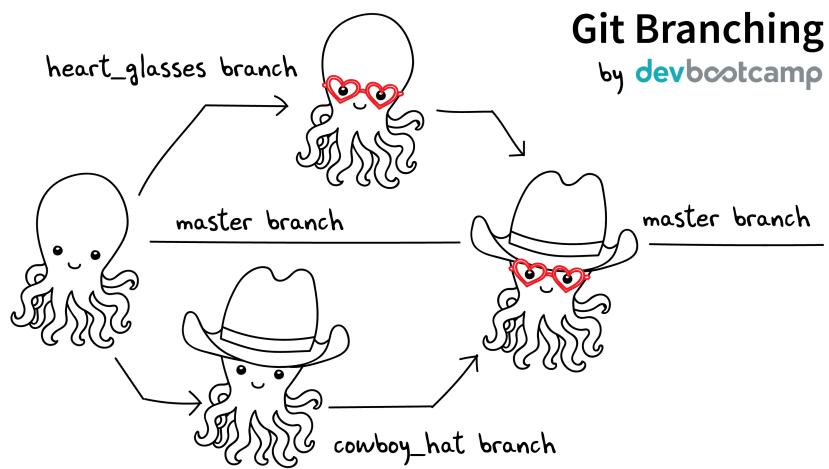
README.TXT

LAB EXERCISE
MPG
Background:
1. Professional programmers carefully design the classes they need before any coding is done. With well-designed classes, programming is much easier and the program has fewer bugs. Object-oriented design consists of deciding what classes are needed, what data they will hold, and how they will behave. These decisions are documented (written up) and then examined. If something doesn't look right, it is fixed before any programming is done.
2. The specifications of a class that models the fuel efficiency of a car would be:
Variables
 int myStartMiles; // Starting odometer reading
 int myEndMiles; // Ending odometer reading
 double myGallonsUsed; // Gallons of gas used between the readings
Constructors
 // Creates a new instance of a Car object with the starting
 // odometer reading;

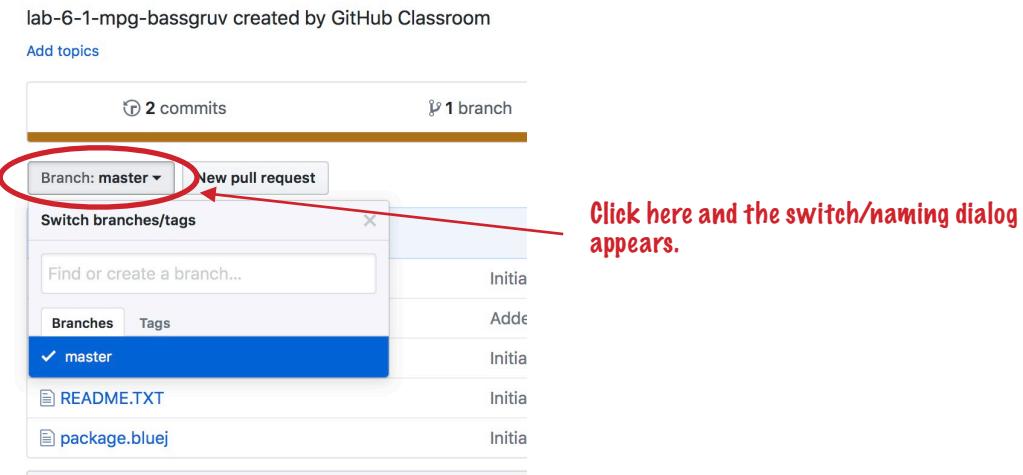
All these starter files are here for you to work in; including the .pdf of the lab specification. Sometimes I may include the spec in repo README shown below here as well.

Here is the button to press to create a branch

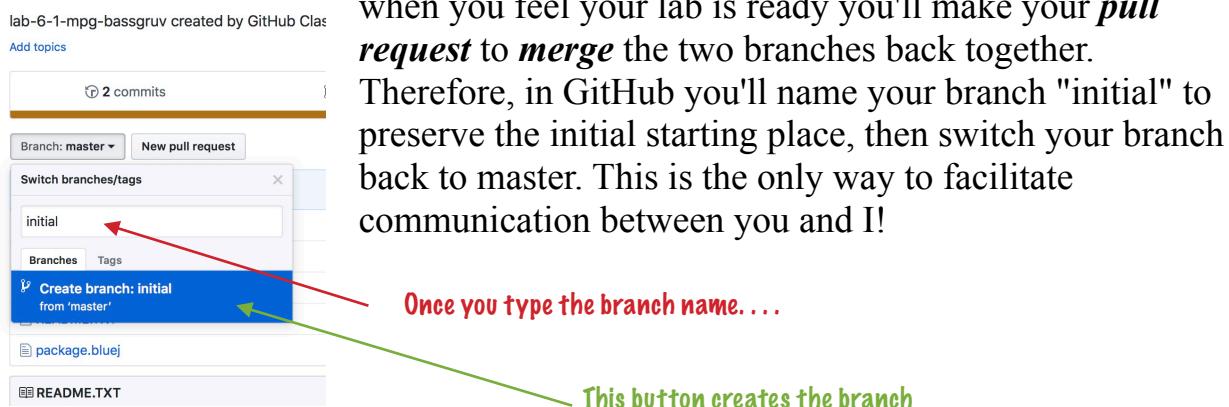
- For you and I to communicate through this system, you'll need to create a **branch** in your repo. For now, think of this branch as a safe "side road" from your main code. This allows for a few things. First, if you're trying to add a feature to your program and it just turns out to be an ugly mess, you can simply return to your master branch and delete your experiment; magic recovery!! Second, in the git system, while on a branch you can create what is called a **pull request** which is a request to collaborate with others on your code. This is the mechanism where you can ask me vexing questions about some problem in your lab or tell me that you're ready to have your lab scored!! Also, when we have group projects later this year, branching allows multiple people to work on the same project simultaneously. Once some collaboration has occurred a pull request can be **merged** back into the master branch.



- So now we'll make a branch. Press the branch button shown on page 3 and you'll see this:



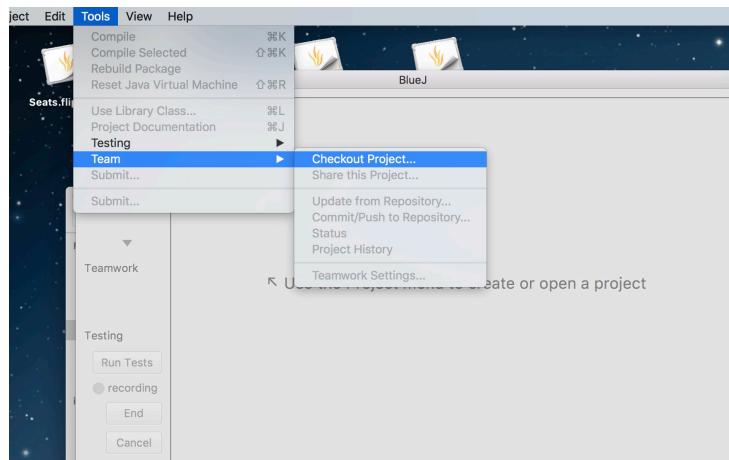
- Now, here is something a bit unfortunate. BlueJ's internal git interface always works on the "master" branch by default and there's no way to change this. So, the branch we'll create will preserve the initial starting place of the project. You'll do all of your atomic commits to the "master" branch and when you feel your lab is ready you'll make your **pull request** to **merge** the two branches back together.



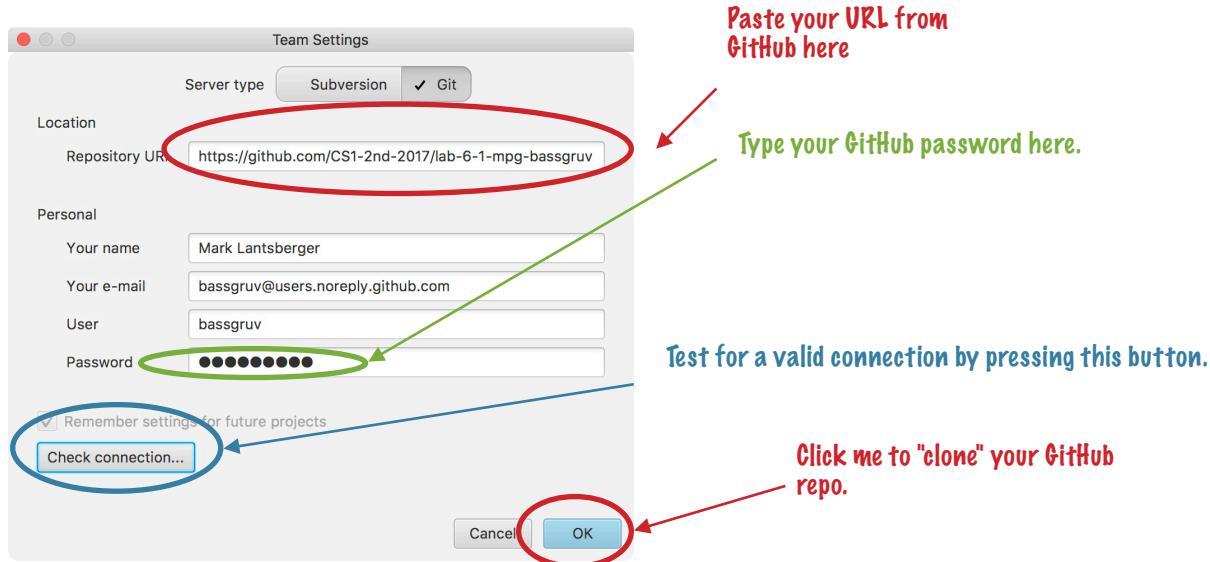
- Be sure to press the Branch button again and bring GitHub back to showing the "master" branch so you can view your progress. Copy the URL for this repo.

Checking Out with BlueJ

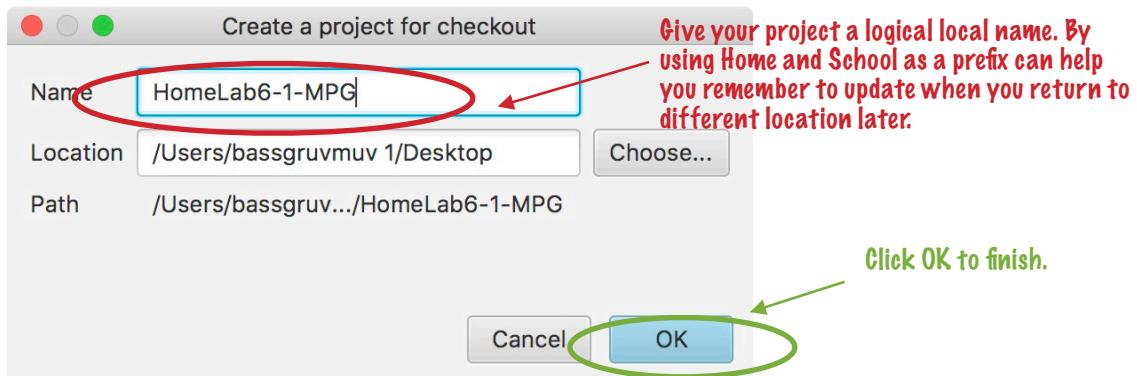
- You may have noticed a difference from how you've practiced this before. Your starting repo on GitHub already has some files in it! This makes creating a "local" repo on your computer super easy. It's known as the Checkout process.
- Launch BlueJ and close any other projects which may auto-open. Then choose Tools, followed by Team, followed by Checkout Project.



- Which brings you to the Checkout dialog



- Next, you'll need to name your "local" repo. I'm just putting this one on the Desktop, but be sure to create it in a logical location in your folder hierarchy!



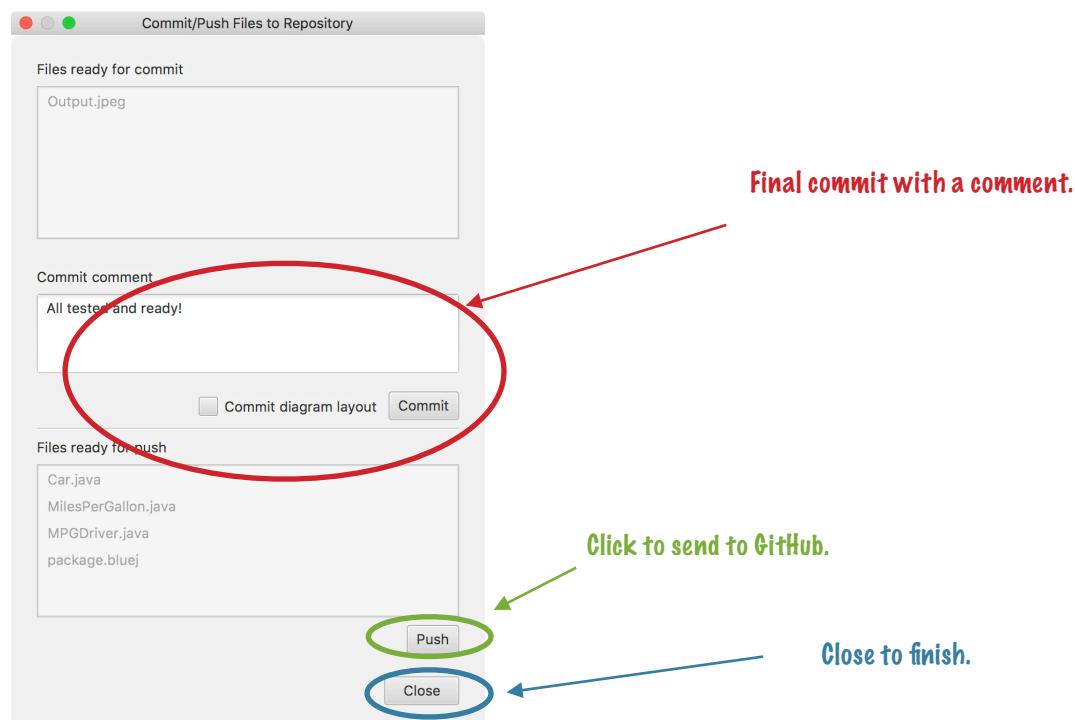
- From this point you may proceed as you've already done before. Be sure you commit often with comments along the way. This is the correct professional habit to have!! My expectation is to see many commits in your project. Remember, you don't have to push your commits to GitHub until the end of your work session.

Finishing your project and submitting on GitHub

- So, you've written your code (with lots of commits along the way) and tested it and you're feeling like you're ready to submit it. At this point, double check the lab specification (spec) .pdf file and see if there is any test data presented to you. You'll want to run your program using this test data and take a screenshot of your output window:

The screenshot shows a BlueJ terminal window titled "BlueJ Terminal Window - HomeLab6-1-MPS". It displays the following text:
New Car odometer reading: 15
Filling Station Visit
odometer reading: 250
gallons to fill tank: 10
Miles per gallon: 23.50
Filling Station Visit
odometer reading: 455
gallons to fill tank: 12.5
Miles per gallon: 16.40
At the bottom, a status bar says "Can only enter input while your programming is running".

- Once you have this screenshot, save it directly into your project folder. BlueJ will notice this and include it in your push. Now make your final push in BlueJ.



- Return to GitHub, we can now open a ***pull request*** to initiate communication.

Click this button to open a pull request.

The screenshot shows a GitHub repository page for 'lab-6-1-mpg-bassgruv'. At the top, there's a red circle and arrow pointing to the 'New pull request' button, which is highlighted in green. Below the button, a list of files is shown with their commit history. The README file contains a lab exercise about designing classes for a car. The bottom part of the screenshot shows the 'LAB EXERCISE MPG' section of the README.

- You'll notice there's nothing to compare!! It's because GitHub is comparing the "master" branch to itself. We want to compare the "master" branch to the "initial" branch we created earlier.

Click this button to alter "master" to "initial".

The screenshot shows a GitHub repository page for 'Comparing master...master'. A red circle and arrow point to the 'base: master' dropdown menu, which is highlighted. Below it, another dropdown menu shows 'compare: master'. The main content area displays a message: 'There isn't anything to compare. You'll need to use two different branch names to get a valid comparison. Check out some of these sample comparisons.' It lists two example comparisons: 'initial' vs 'initial' (an hour ago) and 'master@{1day}...master' vs 'master' (24 hours ago).

- The Open a **pull request** page appears.

Notice the difference here.

Type a title for this pull request similar to the one you see here. Details are helpful.

Please include your output window here as well. It's very helpful to me. Start by typing the first line you see here. Then click on the link indicated here. This link will allow you to navigate to your output file and attach it to your request.

Click me to complete the pull request!!

- Once complete, here are the results you'll see. At this point there is nothing left for you to do but wait. I'll get to scoring your lab as soon as I'm able.

Hey Mr L. I'm ready to score Lab 6.1 MPG #1

bassgruv wants to merge 3 commits into initial from master

Conversation 0 Commits 3 Checks 0 Files changed 5

bassgruv commented just now

Here is my output window

```
New Car odometer reading: 15
Filling Station Visit
odometer reading: 208
gallons to fill tank: 18
Miles per gallon: 23.56
Filling Station Visit
odometer reading: 455
gallons to fill tank: 12.5
Miles per gallon: 16.40
```

+106 -34

Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications
You're receiving notifications because you authored the thread.

1 participant

Click me to complete the pull request!!

Checking your Lab scores!!

- So now you've waited a while and maybe even gotten an email notification that your pull request has been *merged*. This is an indication your lab has been scored.
- So log into your GitHub account and choose the repo for the lab in question. You'll see this:

The screenshot shows a GitHub repository page for 'CS1-2nd-2017 / lab-6-1-mpg-bassgruv'. The 'Pull requests' tab is highlighted with a red circle. Below it, the 'Issues' tab is also circled in red. A green arrow points from the text 'Click this tab. But you'll notice there are 0 pull requests!!' to the 'Pull requests' tab. Another green arrow points from the text 'It's because your request was closed (and merged) when I scored your lab for you. Click on this location and you can view your "closed" pull request.' to the 'Issues' tab. The page displays 0 open pull requests and 1 closed pull request. A ProTip at the bottom suggests using filters to find issues.

It's because your request was closed (and merged) when I scored your lab for you. Click on this location and you can view your "closed" pull request.

Click this tab. But you'll notice there are 0 pull requests!!

There aren't any open pull requests.

Use the links above to find what you're looking for, or try a new search query. The Filters menu is also super helpful for quickly finding issues most relevant to you.

© 2018 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

- Now you can view your closed pull request. Click on the title of the closed pull request:

Your recently pushed branches:

initial (3 minutes ago) Compare & pull request

is:pr is:closed Labels Milestones New pull request

Hey Mr L. I'm ready to score Lab 6! MPG
#1 by bassgruv was merged 3 minutes ago 5

toTip! Click a checkbox on the left to edit multiple issues at once.

© 2018 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#) Contact GitHub [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

Click this title and the details of the pull request will be shown.

- This view can be scrolled down to view individual comments I may have left about your code. Towards the bottom you'll find your lab score.

Car.java

```

48 +     */
49 +     double calculateMPG()
50 +
51 +     return (myEndMiles - myStartMiles) / myGallonsUsed;

```

bassgruv 5 minutes ago
Yay!! You got the Math right!

bassgruv reviewed 4 minutes ago View changes

MilesPerGallon.java

```

18 +     * @param args The command line arguments (not used)
19 +
20 +     */
21 +    public static void main(String[] args)
22 +
23 +

```

bassgruv 4 minutes ago
Yes! You needed local variables in the Driver as well! This driver is great by the way!

bassgruv merged commit [478cb2b](#) into initial a minute ago Revert

bassgruv commented a minute ago ...

This looks great Mark. 50/50

- If there is no score, it may be that you've made some errors you didn't recognize and you'll need to go back, modify your code and resubmit. At that point you'll want to continue working back in BlueJ and make another pull request at a later time.
- IF you received a score, we just want to tidy up your repo so there are no remaining loose ends.

The screenshot shows a GitHub repository page for 'CS1-2nd-2017 / lab-6-1-mpg-bassgruv'. The 'Code' tab is highlighted with a red circle and an arrow pointing to it from the left. The 'Branches' tab is highlighted with a green circle and an arrow pointing to it from the right. The repository has 5 commits, 2 branches, and 0 releases. A yellow box highlights the 'Initial' commit. The 'Branch: master' dropdown is set to 'master'. The 'Clone or download' button is visible. The README.txt file contains a lab exercise about MPG with two questions.

Click on the Code tab to make it active.

Click on the branches tab to reveal the branches present in this project.

```

LAB EXERCISE
MPG
Background:
1. Professional programmers carefully design the classes they need before any coding is done. With well-designed classes, programming is much easier and the program has fewer bugs. Object-oriented design consists of deciding what classes are needed, what data they will hold, and how they will behave. All these decisions are documented (written up) and then examined. If something doesn't look right, it is fixed before any programming is done.
2. The specifications of a class that models the fuel efficiency of a car would be:

```

- Click to delete the initial branch. This is the final step once you know you won't be modifying this code any longer and your Lab is totally complete! And that's it!!!

The screenshot shows a GitHub repository page for 'CS1-2nd-2017/lab-6-1-mpg-bassgruv'. The 'Overview' tab is selected. In the 'Your branches' section, there is a table with one row:

Branch	Last Updated	Action Buttons
<code>initial</code>	Updated 8 minutes ago by bassgruv	0 1 New pull request trash can icon (circled in red)

A red arrow points from the text 'Click this button to delete this branch.' to the trash can icon in the 'Action Buttons' column of the 'initial' branch row.

Click this button to delete this branch.

A Recap of the GitHub Assignment Process (From L03B)

Starting a new "Assignment" Project

- Follow the link posted on Edmodo and accept Assignment.
- Create the "initial" branch on GitHub
- Use the "Checkout" process in BlueJ to create a "local" repo
- Write code and test!!!
 - **commit** often!!! Don't forget Git comments.
 - **push** to GitHub at least once at the end of your work session.

Working In A New Location

- In BlueJ, make a **clone** of the desired GitHub repo using "Checkout".
- Write code and test!!!
 - **commit** often!!! Don't forget Git comments.
 - **push** to GitHub at least once at the end of your work session.

Resuming Work In A Previous Location

- Launch BlueJ using the package icon in your "local" repo.
- Click the "Update" button (this performs a **pull**).
- Write code and test!!!
 - **commit** often!!! Don't forget Git comments.
 - **push** to GitHub at least once at the end of your work session.

Submitting Work for Grading

- Test your code. Take a screenshot of final run and save to project folder.
- Create a **pull request** on GitHub. Be sure to alter the branches!
- Make a title and upload your output window in your **pull request**. Submit.
- Wait for your scoring to be completed.
- Check your score and comments to you. Revise code if needed.
- If scored, delete the "initial" branch.