

Open Source SW & Lab - Summer 2023

3. Git-Branching

Walid Abdullah Al

Computer and Electronic Systems Engineering
Hankuk University of Foreign Studies



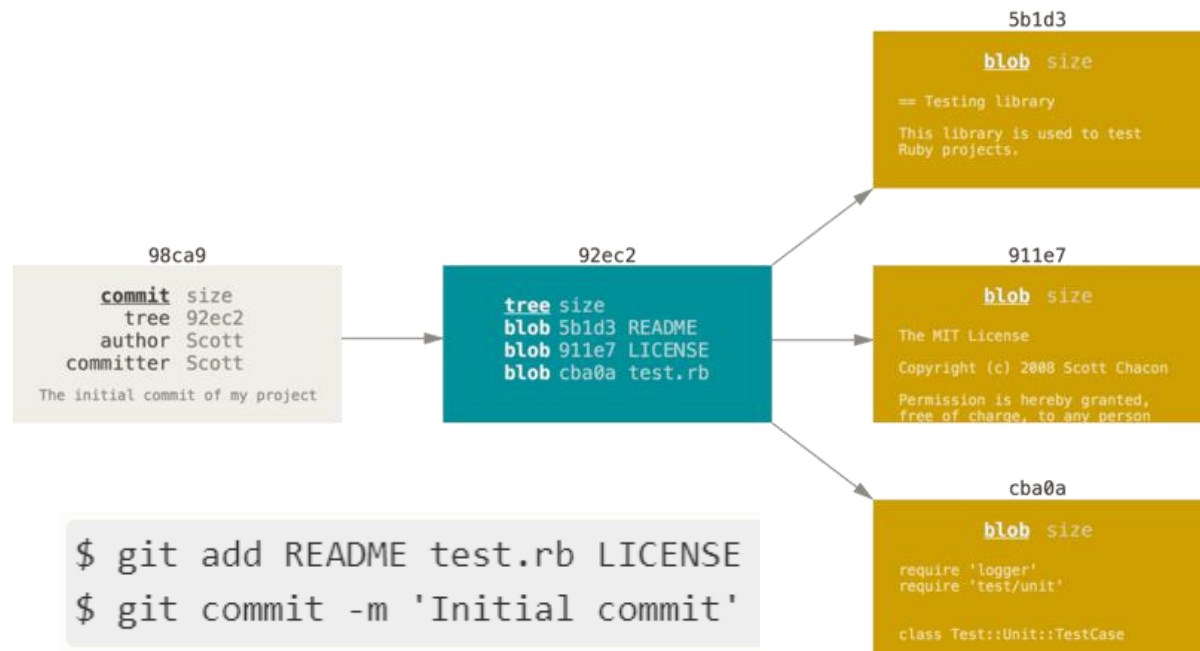
Computer Vision Lab
Hankuk University of Foreign Studies

Based on:

Pro Git (2022) by Scott Chacon, Ben Straub

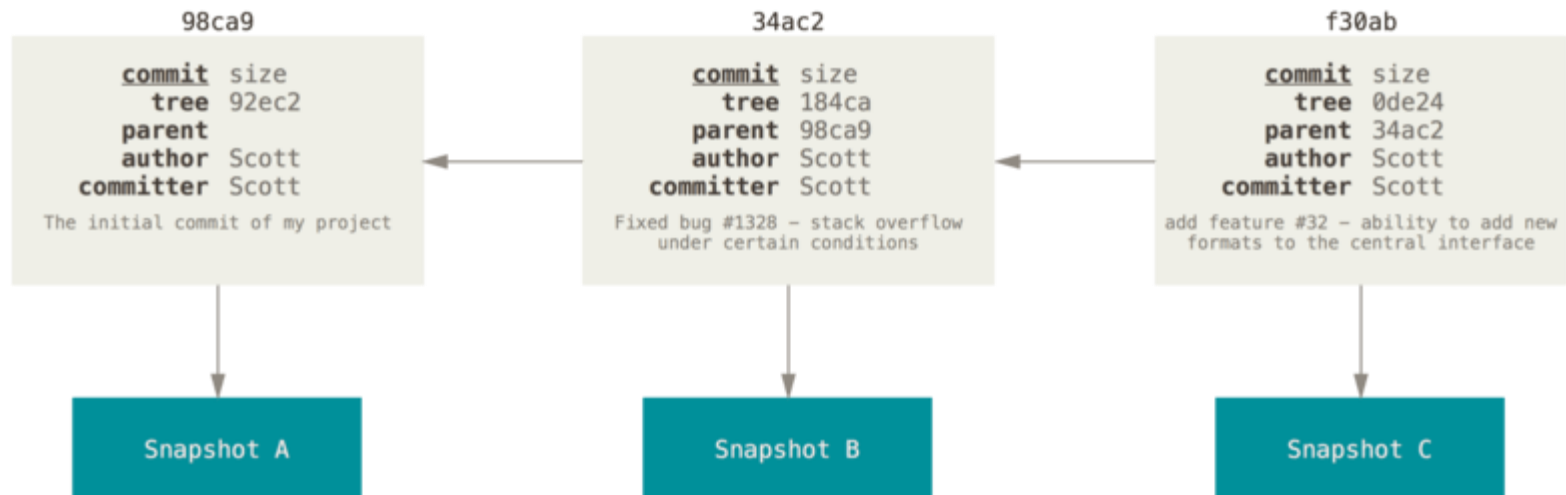
A commit and its tree

- **After a commit,**
 - A **blob** stores the file contents
 - A **tree** lists the blobs/contents in the directory
 - A **commit** points to the tree-root



Subsequent commits

- The next commit stores a pointer to the commit that came immediately before it.



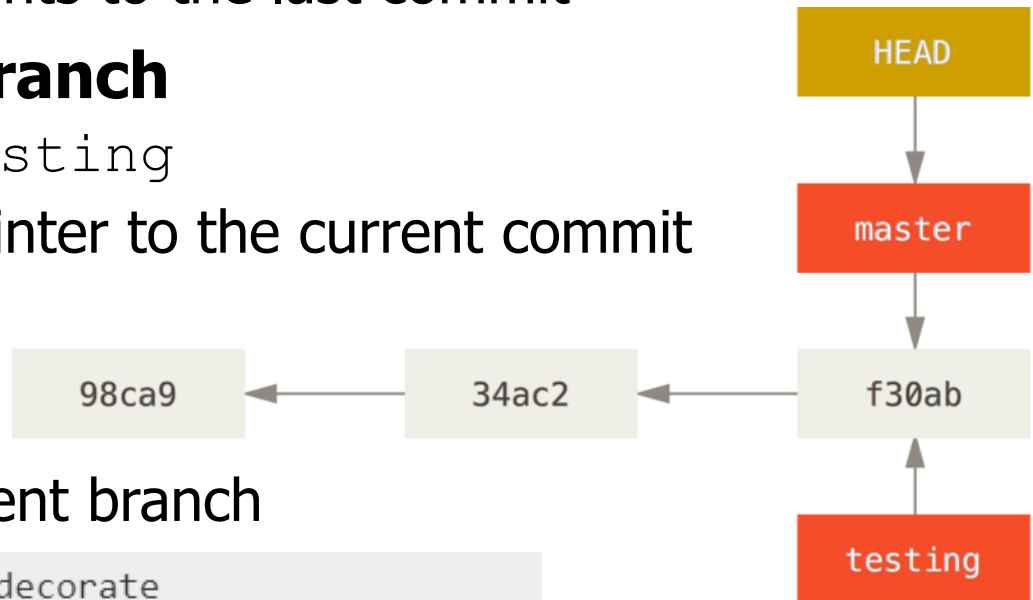
A Branch

- **A lightweight pointer to a commit**
- **Master branch**
 - Default branch in Git
 - Master branch points to the last commit
- **Creating a new branch**
 - `git branch testing`
 - Creates a new pointer to the current commit

- **HEAD**

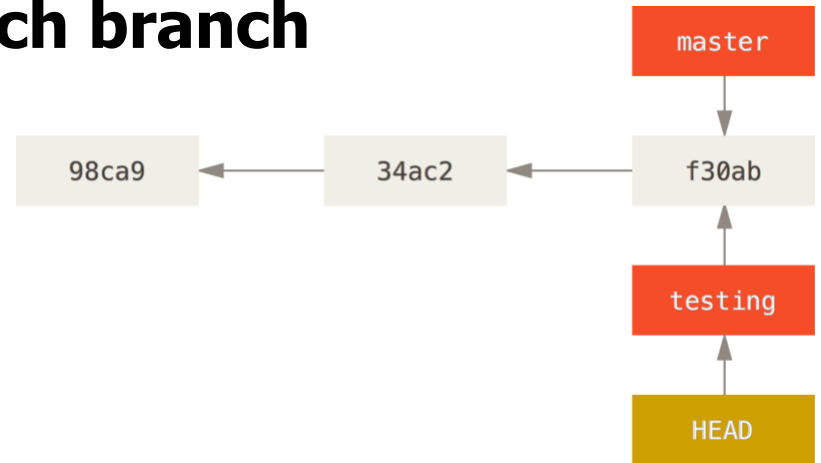
- Points to the current branch

```
$ git log --oneline --decorate
f30ab (HEAD -> master, testing) Add feature #32
34ac2 Fix bug #1328 - stack overflow under certa
98ca9 Initial commit
```



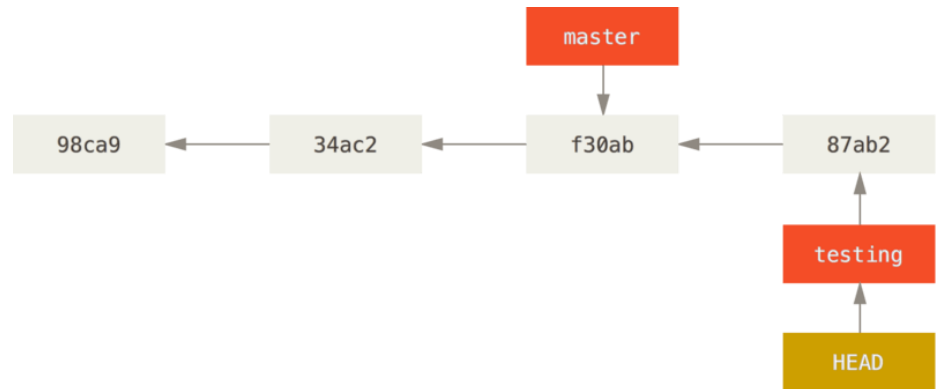
Switching branch

- `git branch` does not switch branch
- Use `$ git checkout testing`



- After a commit

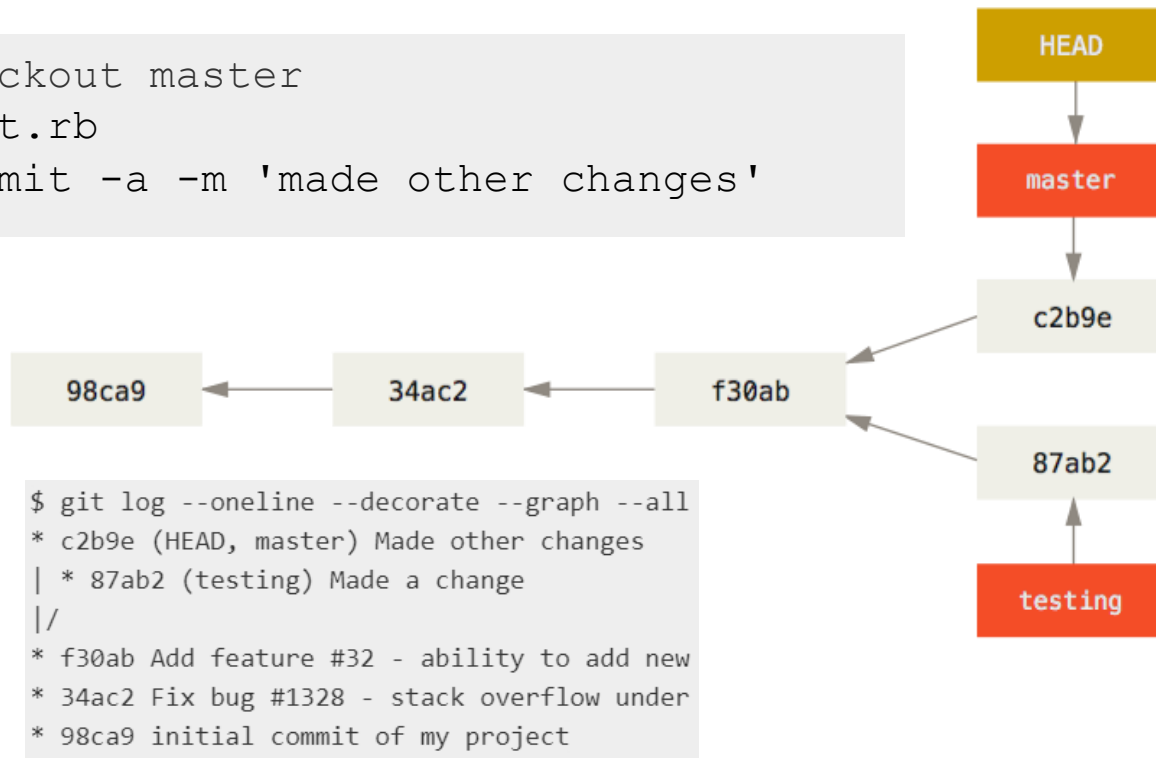
```
$ vim test.rb  
$ git commit -a -m 'made a change'
```



Divergent history

- **Switch back to the main branch and do some changes**
 - Isolated changes in separate branches
 - Called “divergent history”

```
$ git checkout master
$ vim test.rb
$ git commit -a -m 'made other changes'
```



Basic branching and merging

- Your usual flow
 - Do some work on a website
 - Create a branch for a new user story you're working on
 - Do some work in that branch
- Receive an issue and need a hotfix
 - Switch to your production branch
 - Create a branch to add the hotfix
 - After it's tested, merge the hotfix branch, and push to production
 - Switch back to your original user story and continue working

Basic branching

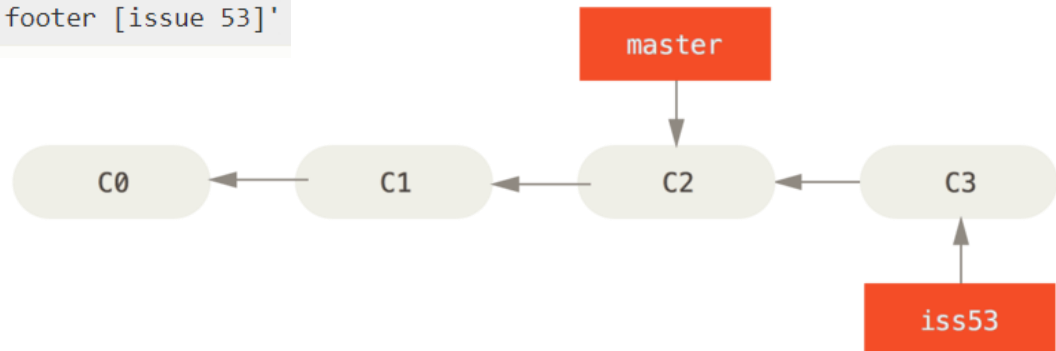
- You have decided to work on an issue (issue #53)

```
$ git checkout -b iss53  
Switched to a new branch "iss53"
```

This is shorthand for:

```
$ git branch iss53  
$ git checkout iss53
```

```
$ vim index.html  
$ git commit -a -m 'Create new footer [issue 53]'
```

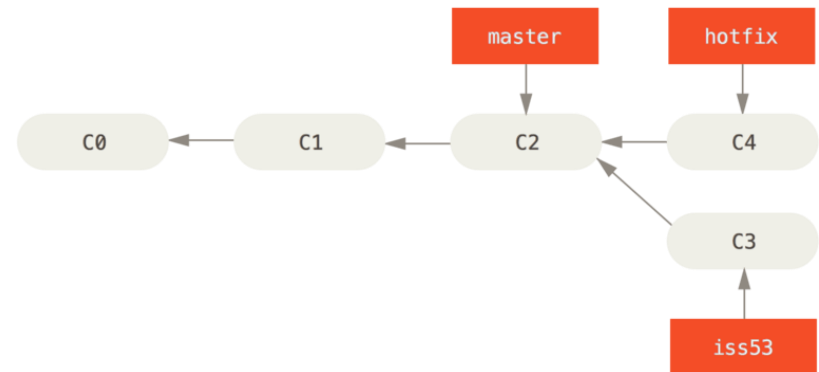


Another urgent issue

- **hotfix branch based on master**

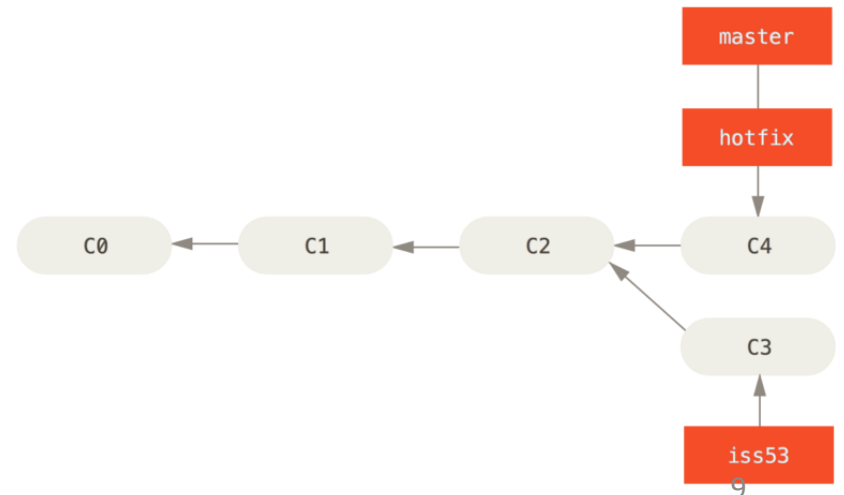
```
$ ???
```

```
$ git checkout -b hotfix  
Switched to a new branch 'hotfix'  
$ vim index.html  
$ git commit -a -m 'Fix broken email address'  
[hotfix 1fb7853] Fix broken email address  
1 file changed, 2 insertions(+)
```



- **Merge hotfix into master ("fast-forward" merge)**

```
$ git checkout master  
$ git merge hotfix  
Updating f42c576..3a0874c  
Fast-forward  
index.html | 2 ++  
1 file changed, 2 insertions(+)
```



- **Delete hotfix branch**

```
$ git branch -d hotfix  
Deleted branch hotfix (3a0874c).
```

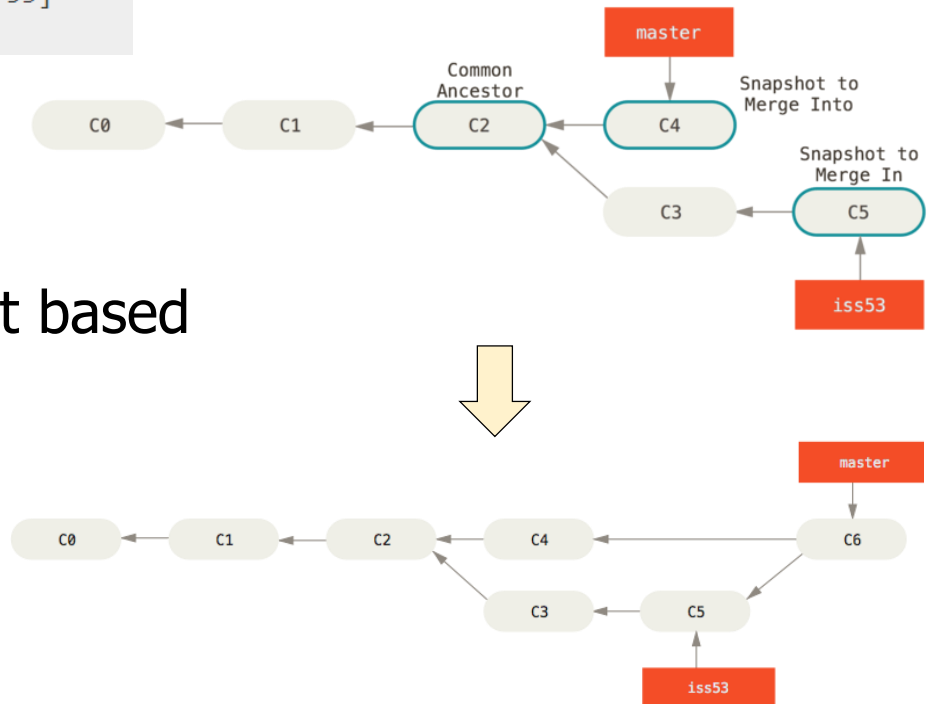
Basic merging: 3-way merge

- **Issue #53 is done**

```
$ git checkout iss53
Switched to branch "iss53"
$ vim index.html
$ git commit -a -m 'Finish the new footer [issue 53]'
[iss53 ad82d7a] Finish the new footer [issue 53]
1 file changed, 1 insertion(+)
```

- **Three-way merge**

- Creates a new snapshot based on three snapshots
- And a new commit



Basic merge conflicts

- **When the same part of a file is modified in the divergent branches**

```
$ git merge iss53
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

```
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")

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

   both modified:   index.html

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

```
<<<<<< HEAD:index.html
<div id="footer">contact : email.support@github.com
=====
<div id="footer">
  please contact us at support@github.com
</div>
>>>>>> iss53:index.html
```

Basic merge conflicts: resolve

- **Open the file manually and resolve**
 - Git adds resolution-markers to the file

```
<<<<<<< HEAD:index.html
<div id="footer">contact : email.support@github.com</div>
=====
<div id="footer">
  please contact us at support@github.com
</div>
>>>>>> iss53:index.html
```

- **Resolution**
 - Choose one or merge yourself

```
<div id="footer">
  please contact us at email.support@github.com
</div>
```

- **Run `git add` to mark this as resolved**

Branch management

- **List all the branches**

```
$ git branch  
  iss53  
* master  
  testing
```

- **See last commit on each branch**

```
$ git branch -v  
  iss53    93b412c Fix javascript issue  
* master   7a98805 Merge branch 'iss53'  
  testing  782fd34 Add scott to the author list in the readme
```

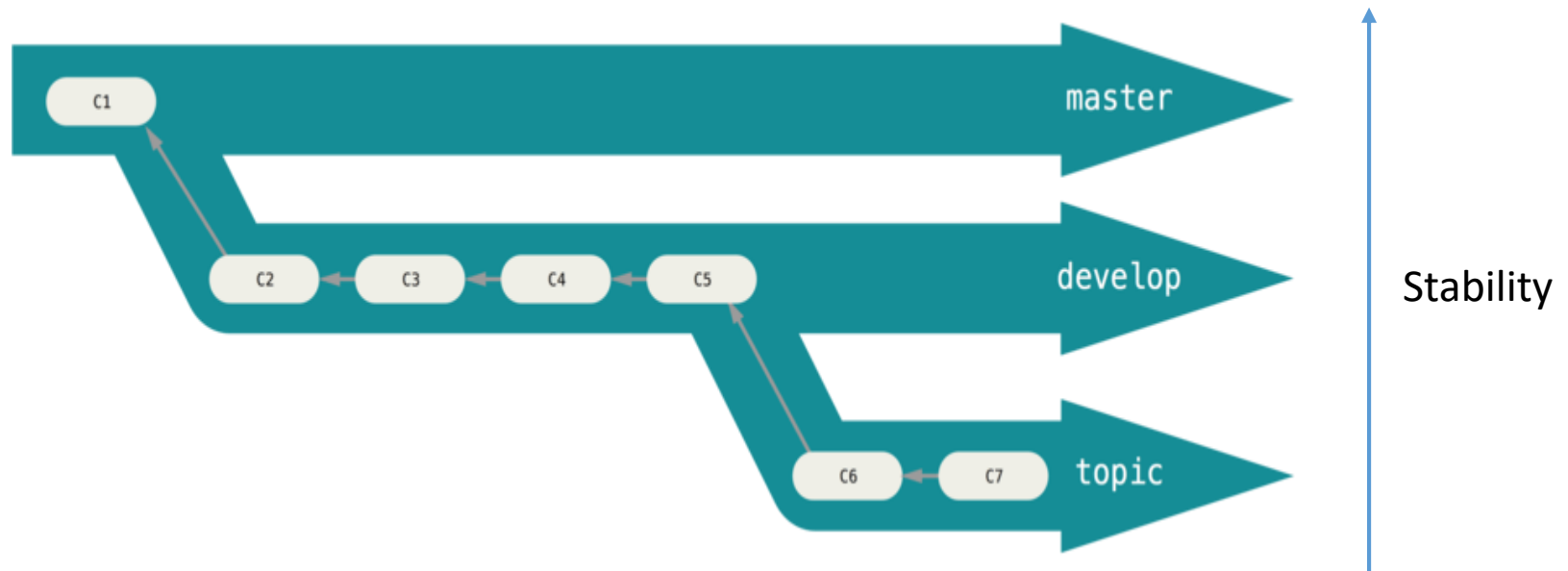
- **Branches merged into the current branch**

```
$ git branch --merged  
  iss53  
* master
```

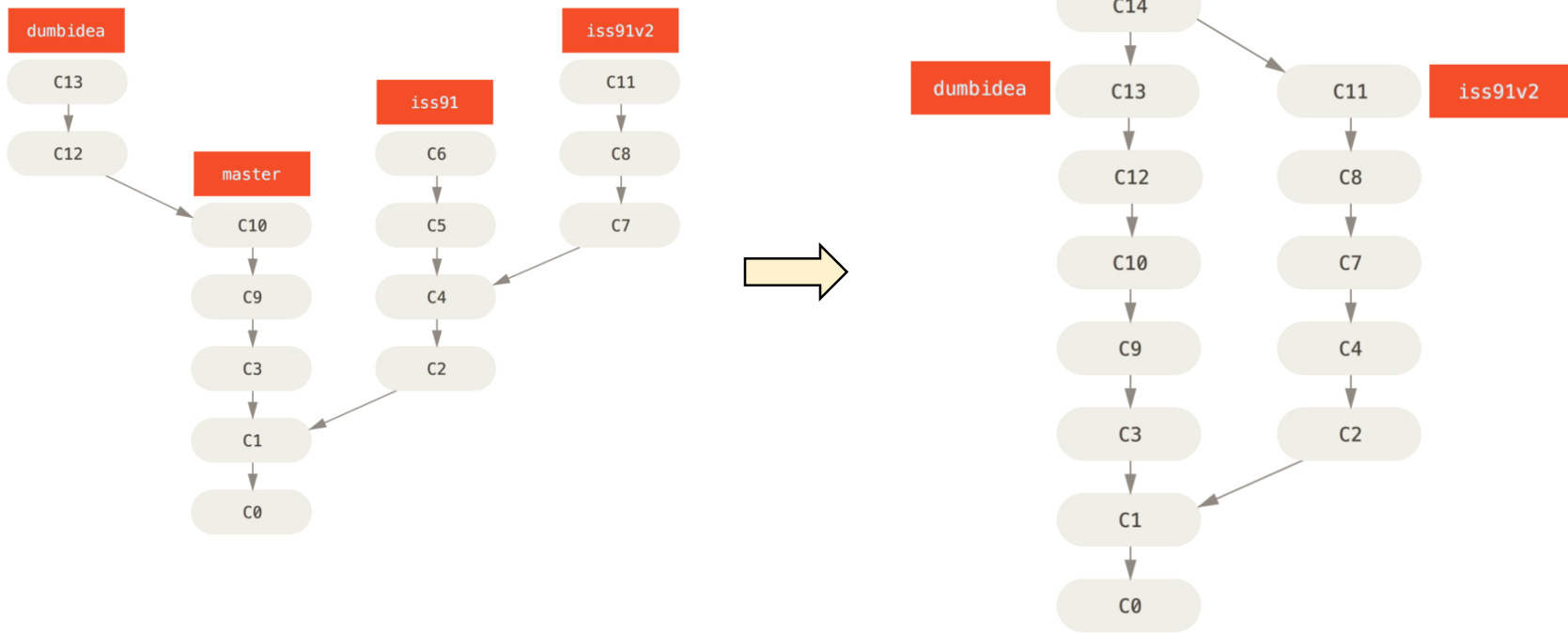
```
$ git branch --no-merged  
  testing
```

Branching workflow: progressive-stability branching

- **Multiple long running branches**

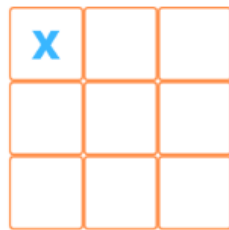


Topic branches

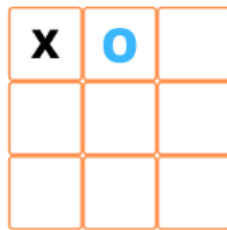


Tic-tac-toe: branching practice

Tic Tac Toe - Gameplay



1 - X Turn



2 - O Turn



3 - X Turn



4 - O Turn



5 - X Turn



6 - O Turn



7 - X Turn

X wins

```
$ python tic_tac_toe.py
Player X turn
- - -
- - -
- - -
Enter row and column numbers to fix spot: 1 1

Player O turn
X - -
- - -
- - -
Enter row and column numbers to fix spot: 2 1

Player X turn
X - -
O - -
- - -
Enter row and column numbers to fix spot: 1 2

Player O turn
X X -
O - -
- - -
```


Download the starter code

- **Download from Eclass**

- **Current version**

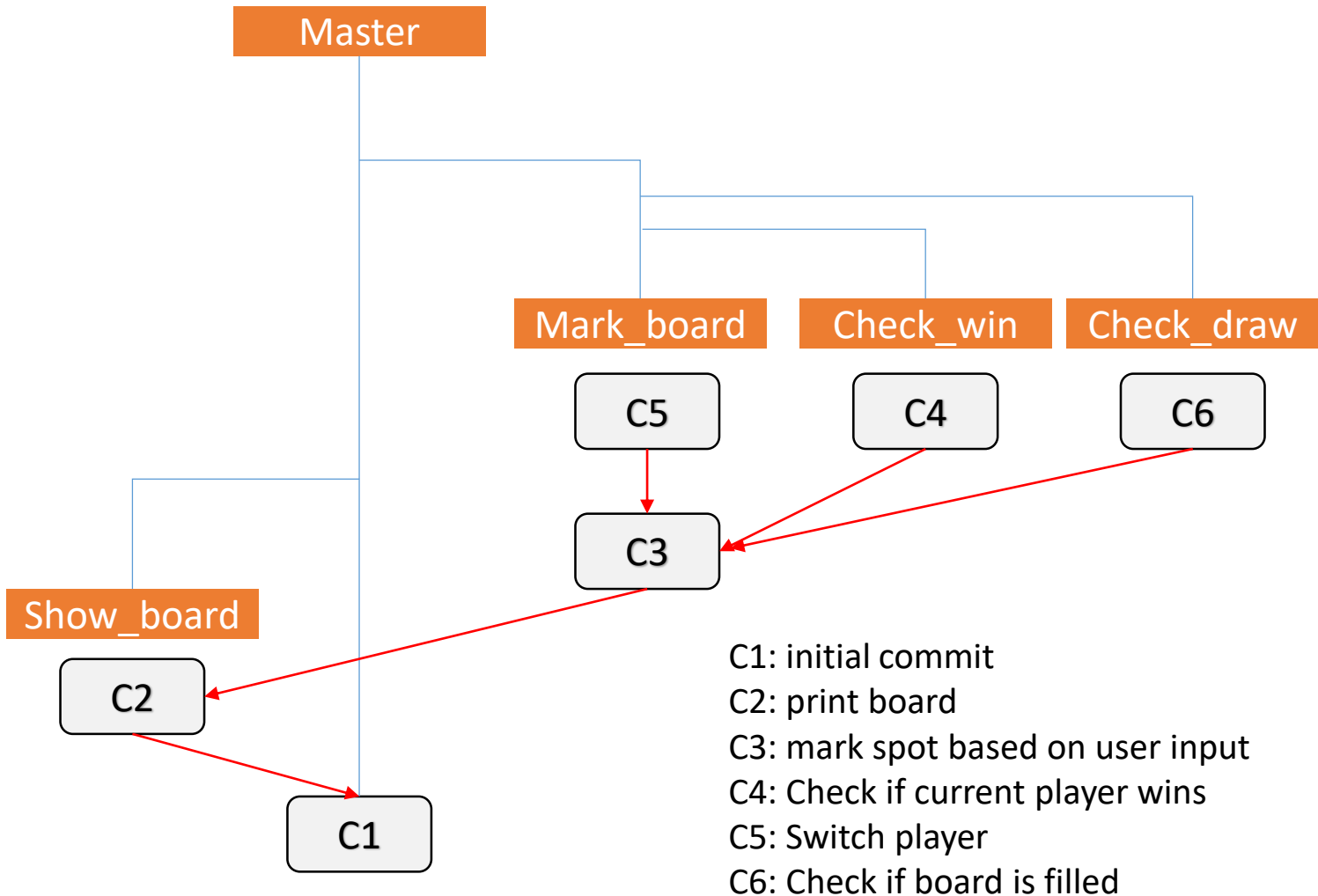
- Board structure
- Inputs from player

```
class TicTacToe:
    def __init__(self):
        self.board=[]

    def init_board(self, empty_sign='-'):
        for i in range(3):
            rows=[]
            for j in range(3):
                rows.append(empty_sign)
            self.board.append(rows)

    def gameplay(self):
        self.init_board()
        player = '0' # can be '0' or 'X'
        while True:
            print("player {}'s turn:".format(player), end="")
            i, j = [int(x) for x in input().split()]
```

Branch-workflow



What you need to submit

- Do a graph check **before** and **after** merging the mark_board, check_win, check_draw

```
git log --oneline --decorate --graph --all
```

- **Take screenshots of each and submit**

Done for today!

- Please, ask me to check your lab attendance