
Git 分散式版本控制

郭忠義

jykuo@ntut.edu.tw

臺北科技大學資訊工程系

版本控制簡介

- 版本控制機制
 - 多人共同合作時，不會把別人的檔案程式覆蓋掉。
 - 可隨時復原修改的內容，回到之前的版本。
 - 可得知今天寫的程式跟昨天的差異處。
 - 針對所發行軟體，可以分成維護版跟開發版管理。
 - 保留修改歷史記錄，包括修改者和理由，以供查詢。
- 儲存檔案庫(Repository)
 - 存放所有規格文件、設計文件、原始程式碼、測試報告等。
 - 暫存檔、紀錄檔、程式編譯後的檔案等可以不予存入。
 - 資訊放在.git資料夾。

Git簡介

- 可處理超大資料能力
 - 管理Linux 核心程式超過一千萬行。
- 採用分散式版本控制機制
 - 開發者本機端有完整程式，不需網路連線也能獨立作業。
 - 可製作不同開發分支，或合併分支。

主要基準線分支

發展測試分支

特殊功能分支

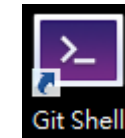


Lab. Git 安裝

- TortoiseGit 介面 + msysgit
- Eclips 內建 Plug-in 模組 Egit 
- GitHub for Windows
 - 下載 <https://github.com/>，GitHubSetup.exe
 - 安裝後，使用Git Shell命令列操作

Download GitHub Desktop for Mac

Download GitHub Desktop for Windows



◀ Welcome

Log in

Configure

Repositories

GitHub GitHub Enterprise

Username or email

Password

The best way to build and ship software. Go to github.com to sign up for an account

✓ Log in

✗ Skip setup

- git for Windows

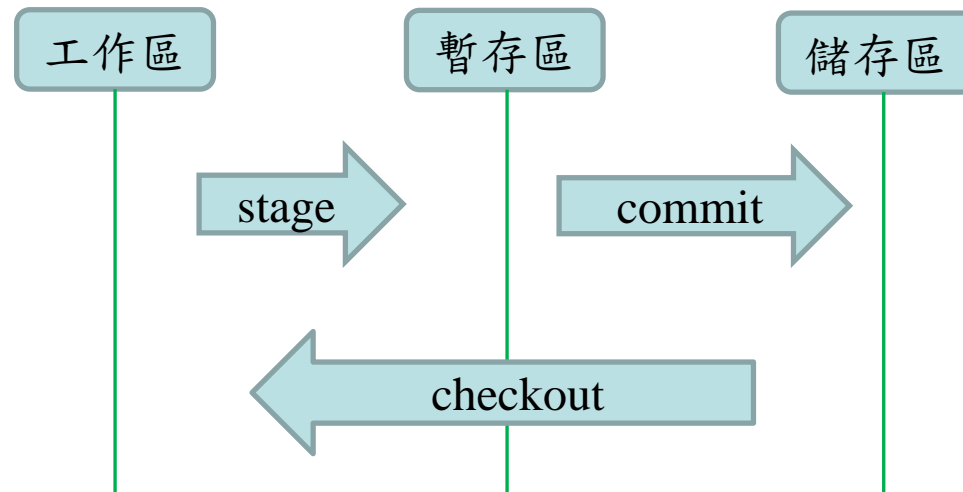
- <https://git-scm.com/downloads>

- <http://ithelp.ithome.com.tw/articles/10132333>

Downloads for Windows

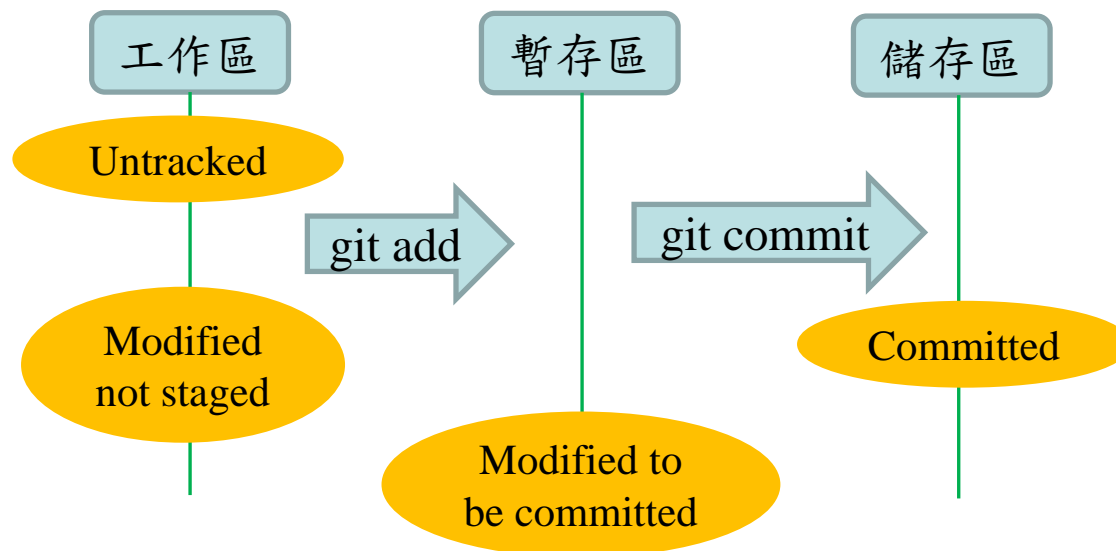
三種檔案區域

- 工作區(Working Directory)：尚未被追蹤控制的檔案區域。
- 暫存區(Staging Area)：被追蹤控制的檔案區域。
- 儲存區(Repository)：存放版控的檔案。



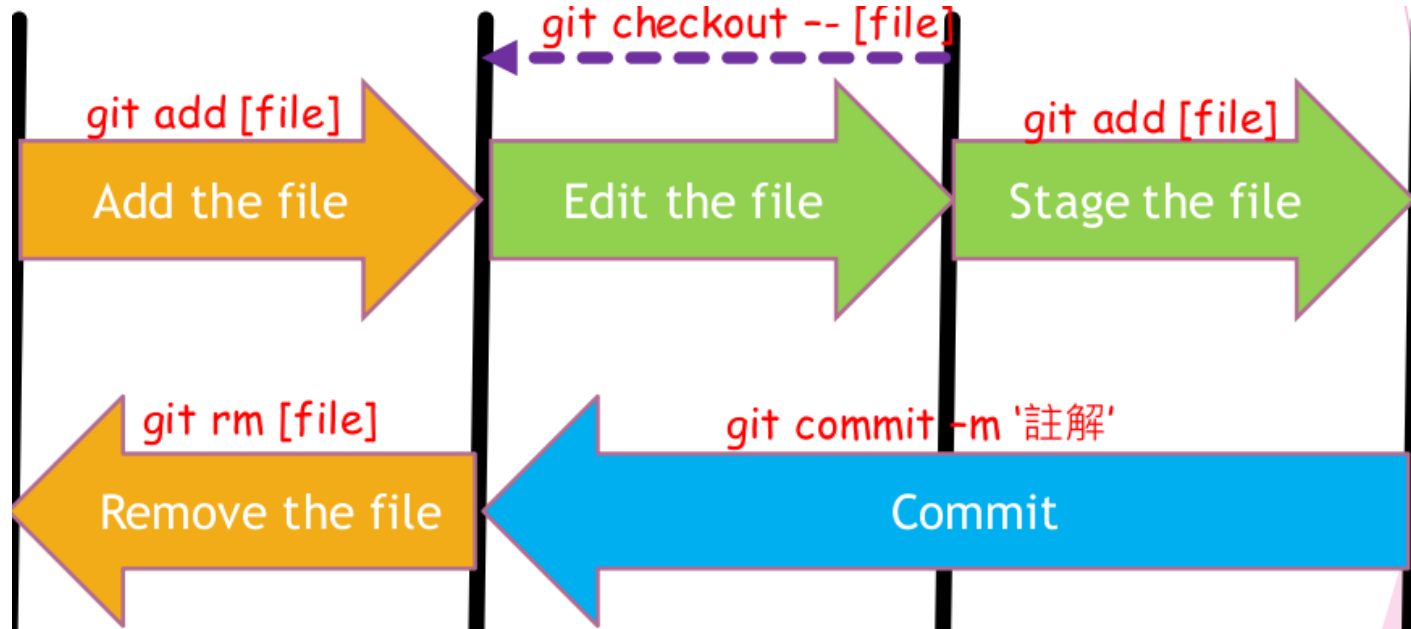
四種檔案狀態₁

- Untracked：沒有納入版本控制範圍內的檔案（紅色）。
- Modified not staged：沒變動。
- Modified to be committed：有變動尚未commit檔案。
- Committed：儲存區中的檔案。



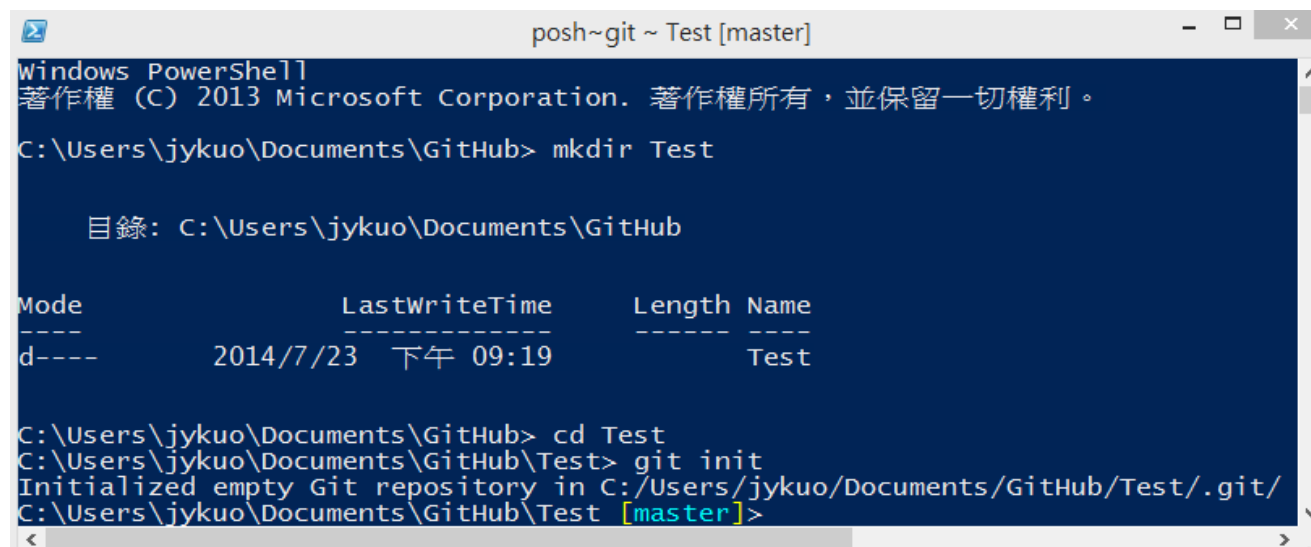
四種檔案狀態₂

Untracked Unmodified Modified staged



Lab 1. 建置工作目錄與儲存庫

- 執行Git Shell
- 建一個空目錄 Test，並切換到Test目錄
 - mkdir Test
 - cd Test
- 在目前目錄中建立儲存庫，檔案放置於.git目錄
 - git init



```
posh~git ~ Test [master]
Windows PowerShell
著作權 (C) 2013 Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\jykuo\Documents\GitHub> mkdir Test

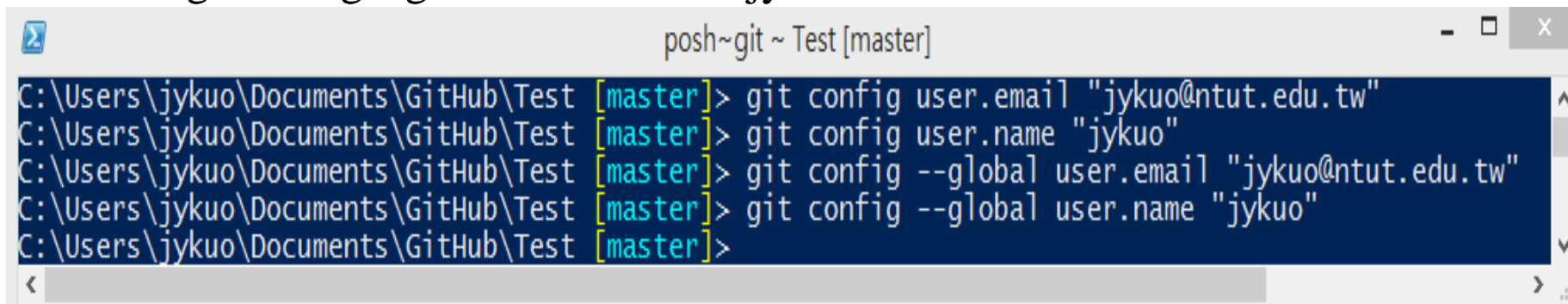
目錄: C:\Users\jykuo\Documents\GitHub

Mode                LastWriteTime         Length Name
----                -
d-----          2014/7/23 下午 09:19             Test

C:\Users\jykuo\Documents\GitHub> cd Test
C:\Users\jykuo\Documents\GitHub\Test> git init
Initialized empty Git repository in C:/Users/jykuo/Documents/GitHub/Test/.git/
C:\Users\jykuo\Documents\GitHub\Test [master]>
```


Lab 2. 組態設定

- 設定本地端姓名和電子郵件，指令為：
 - `git config user.email "jykuo@ntut.edu.tw"`
 - `git config user.name "jykuo"`
- 設定全域姓名和電子郵件，指令為：
 - `git config --global user.email "jykuo@ntut.edu.tw"`
 - `git config --global user.name "jykuo"`



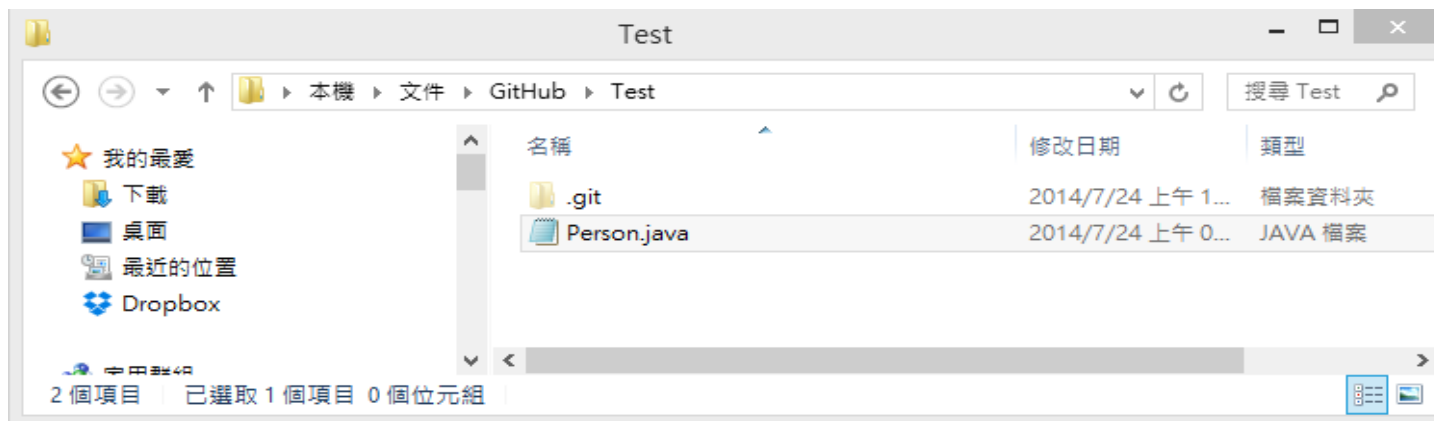
A screenshot of a Windows command prompt window. The title bar shows the path "posh~git ~ Test [master]". The command history is as follows:

```
C:\Users\jykuo\Documents\GitHub\Test [master]> git config user.email "jykuo@ntut.edu.tw"
C:\Users\jykuo\Documents\GitHub\Test [master]> git config user.name "jykuo"
C:\Users\jykuo\Documents\GitHub\Test [master]> git config --global user.email "jykuo@ntut.edu.tw"
C:\Users\jykuo\Documents\GitHub\Test [master]> git config --global user.name "jykuo"
C:\Users\jykuo\Documents\GitHub\Test [master]>
```

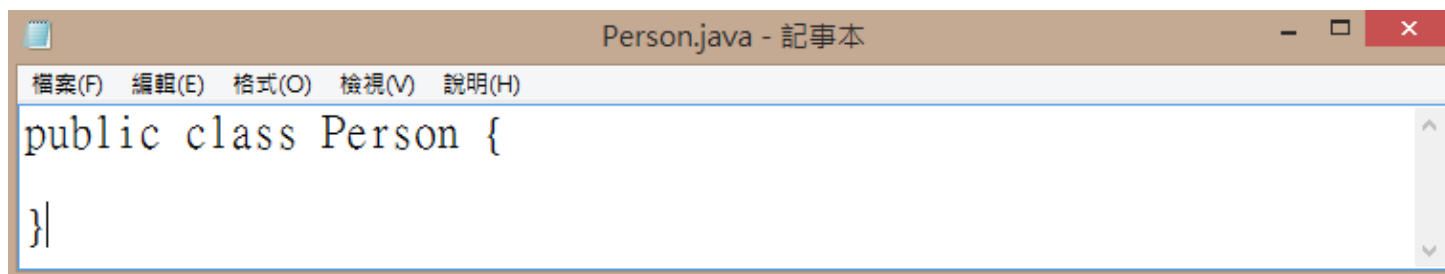
- 查看設定
 - `git config --list`

Lab 3.編輯工作區檔案

- 使用檔案總管在目前目錄，新增一個檔案，命名為 Person.java

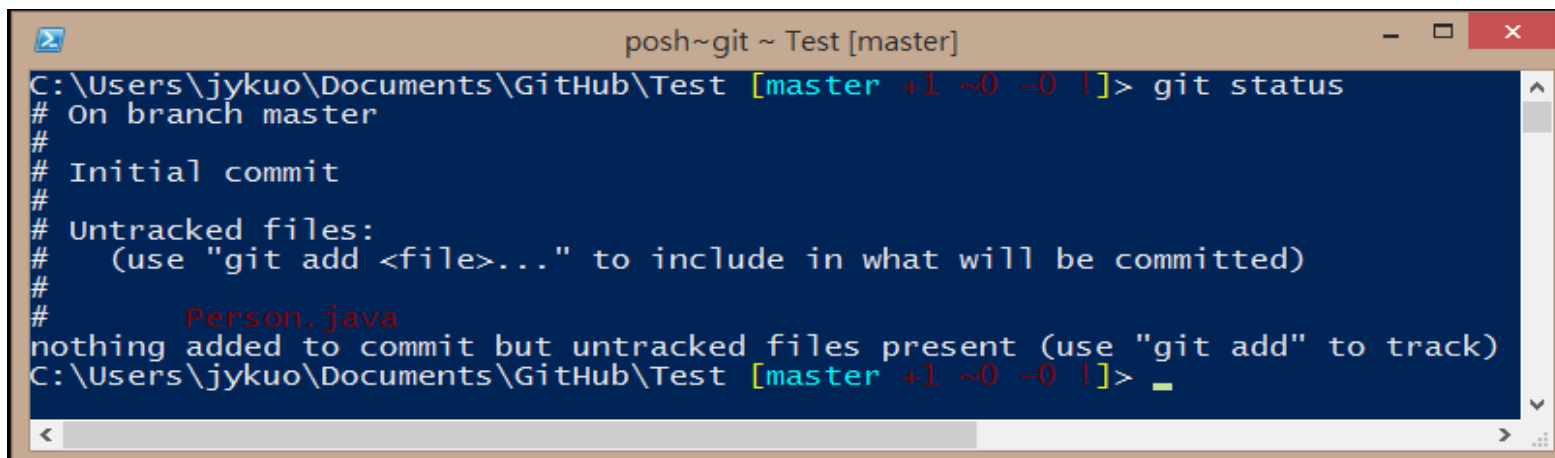


- 使用編輯器如記事本，編輯一份新的程式



Lab 4.檢視目前檔案狀態

- 檢視暫存區、工作區，目前檔案狀態，與儲存庫差異。
- 使用指令：git status 。

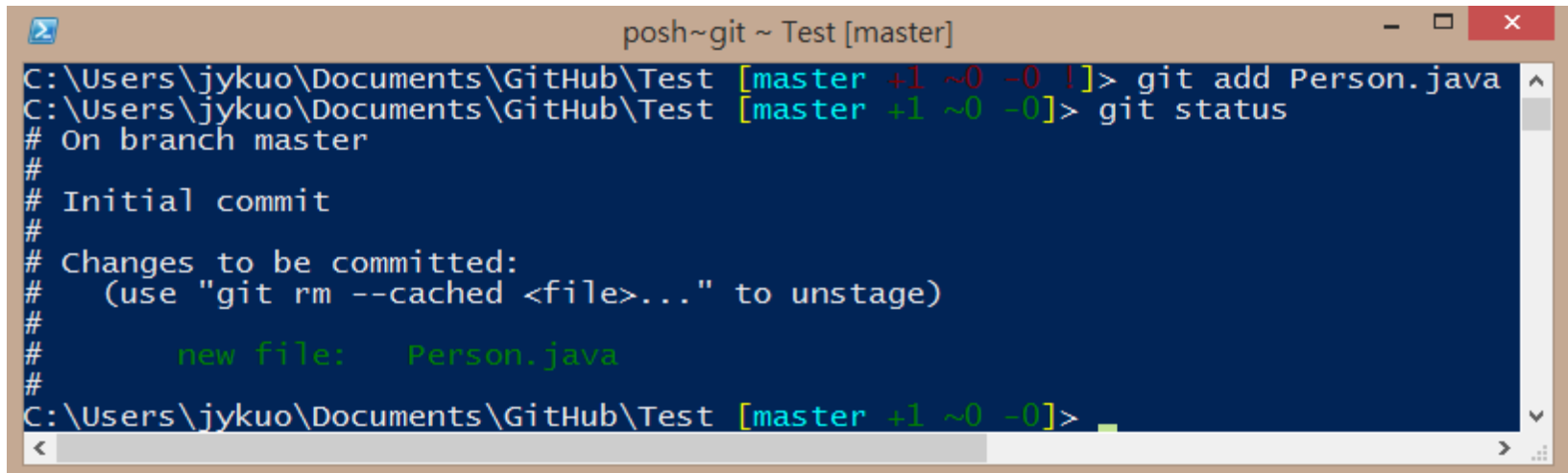


```
posh~git ~ Test [master]
C:\Users\jykuo\Documents\GitHub\Test [master +1 ~0 -0 !]> git status
# On branch master
#
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       Person.java
nothing added to commit but untracked files present (use "git add" to track)
C:\Users\jykuo\Documents\GitHub\Test [master +1 ~0 -0 !]> _
```

- 發現紅色字體 Person.java 為 untracked 檔案，顯示工作區檔案。

Lab 5.將新增檔案加入暫存區

- 把新增檔案加入暫存區。
- 指令為 `git add Person.java` 。

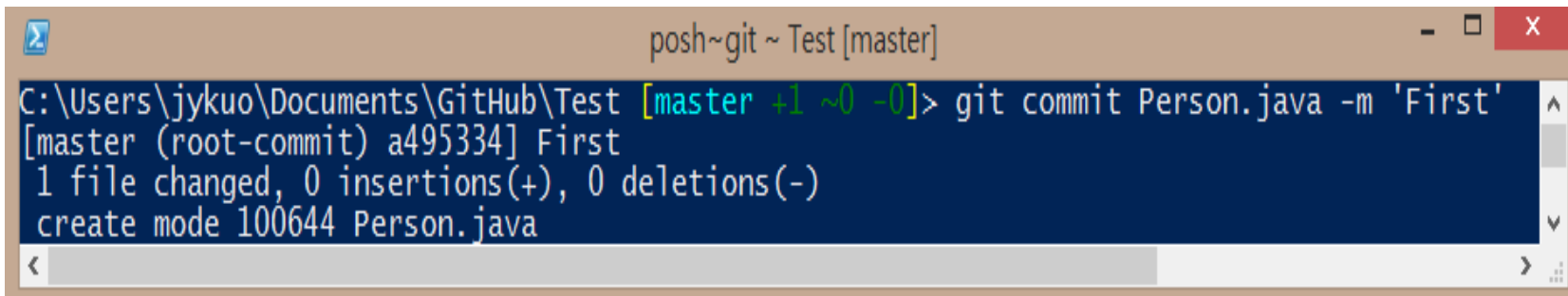


```
posh~git ~ Test [master]
C:\Users\jykuo\Documents\GitHub\Test [master +1 ~0 -0 !]> git add Person.java
C:\Users\jykuo\Documents\GitHub\Test [master +1 ~0 -0]> git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   Person.java
#
C:\Users\jykuo\Documents\GitHub\Test [master +1 ~0 -0]>
```

- 檢視目前檔案狀態，發現綠色字體Person.java加入暫存區。
- `git add .`：將所有在工作區的檔案加入暫存區。

Lab 6.將暫存區檔案簽入儲存庫區

- 確認檔案修改無誤，輸入說明訊息後簽入儲存庫。
- 使用指令：`git commit Person.java -m 'First'`。
- First為標籤(Tag)說明訊息。

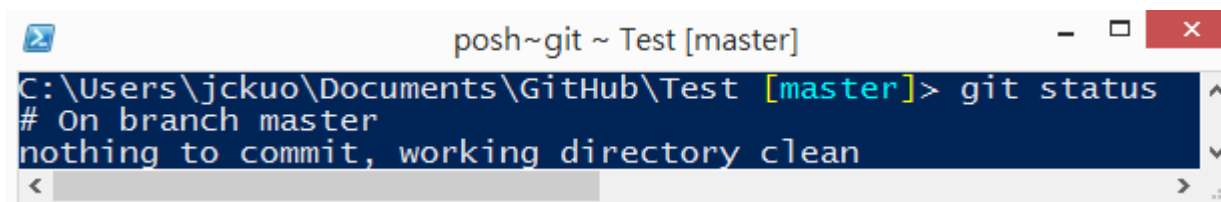
A screenshot of a Windows command prompt window. The title bar reads "posh~git ~ Test [master]". The command prompt shows the path "C:\Users\jykuo\Documents\GitHub\Test" followed by the command `[master +1 ~0 -0]> git commit Person.java -m 'First'`. The output shows the commit hash `[master (root-commit) a495334]`, the message `First`, and the file status `1 file changed, 0 insertions(+), 0 deletions(-)` and `create mode 100644 Person.java`.

```
posh~git ~ Test [master]  
C:\Users\jykuo\Documents\GitHub\Test [master +1 ~0 -0]> git commit Person.java -m 'First'  
[master (root-commit) a495334] First  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 Person.java
```

- 若要將暫存區的檔案全部簽入儲存庫，使用指令：
- `git commit -a -m 'message'`

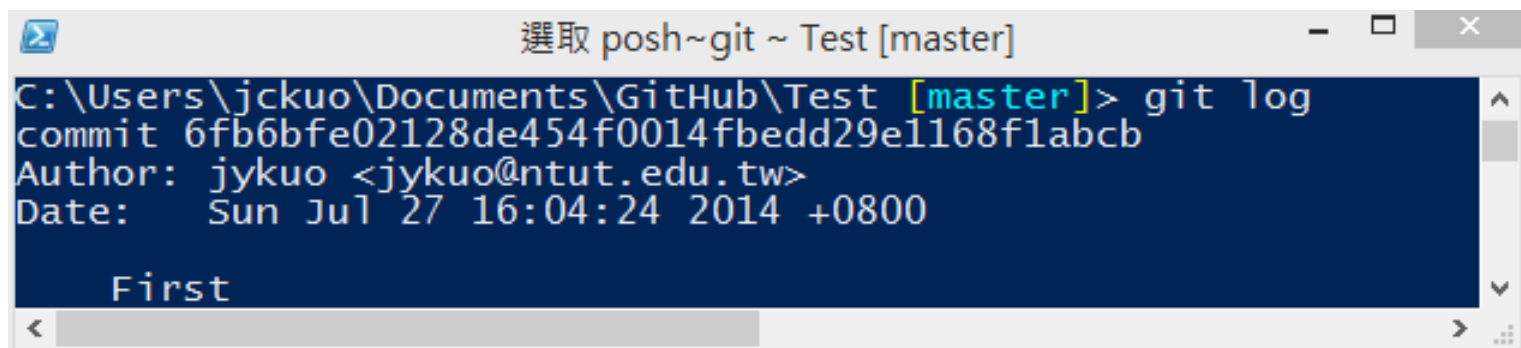
Lab 7. 查看簽入儲存庫區資訊 1

- 先檢視暫存區、工作區，目前檔案狀態，使用指令：git status，



```
posh~git ~ Test [master]  
C:\Users\jckuo\Documents\GitHub\Test [master]> git status  
# On branch master  
nothing to commit, working directory clean
```

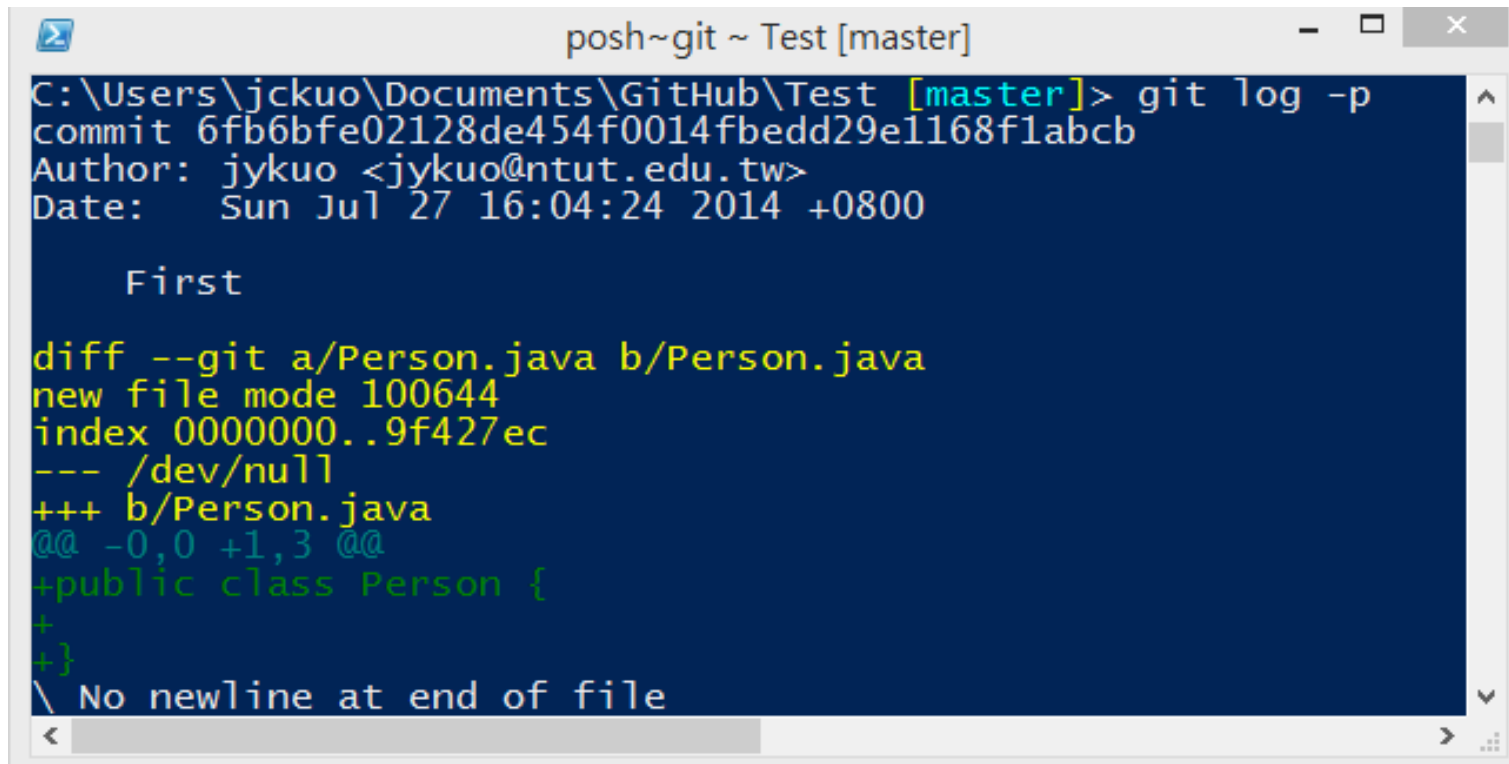
- 顯示工作區與儲存庫檔案一致，暫存區沒有檔案。
- 查看簽入歷史資料，使用指令：git log。
- 發現簽入檔案狀態：包括簽入流水編號、作者、時間、說明，若有簽入過則顯示之前版本資訊。



```
選取 posh~git ~ Test [master]  
C:\Users\jckuo\Documents\GitHub\Test [master]> git log  
commit 6fb6bfe02128de454f0014fbbedd29e1168f1abcb  
Author: jykuo <jykuo@ntut.edu.tw>  
Date: Sun Jul 27 16:04:24 2014 +0800  
  
First
```

Lab 7. 查看簽入儲存庫區資訊 2

- 查看詳細資訊，使用指令 `git log -p`，發現簽入檔案詳細資訊。



```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master]> git log -p
commit 6fb6bfe02128de454f0014fbedd29e1168f1abcb
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Sun Jul 27 16:04:24 2014 +0800

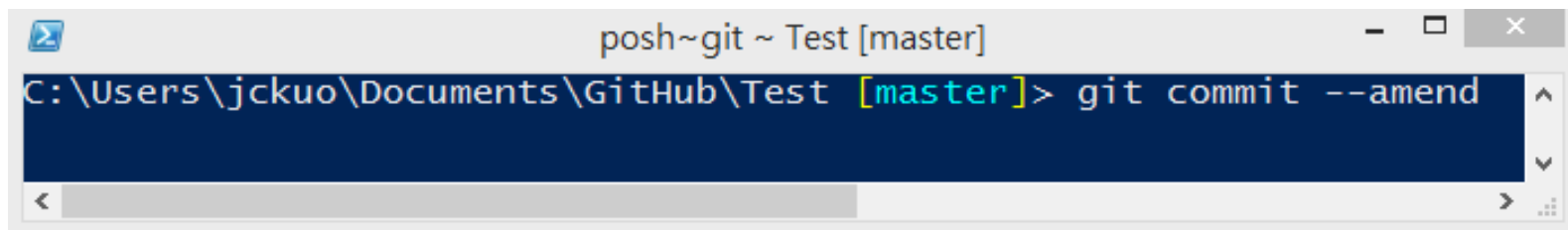
    First

diff --git a/Person.java b/Person.java
new file mode 100644
index 0000000..9f427ec
--- /dev/null
+++ b/Person.java
@@ -0,0 +1,3 @@
+public class Person {
+
+}
\ No newline at end of file
```

- `gitk --all` 圖形化介面顯示

Lab 8.修改簽入的說明₁

- 修改上次簽入說明，使用指令：`git commit --amend`



A terminal window titled "posh~git ~ Test [master]" showing the command `git commit --amend` being executed in a Windows command prompt. The path shown is `C:\Users\jckuo\Documents\GitHub\Test`.

- 顯示記事本，供重新編輯簽入說明

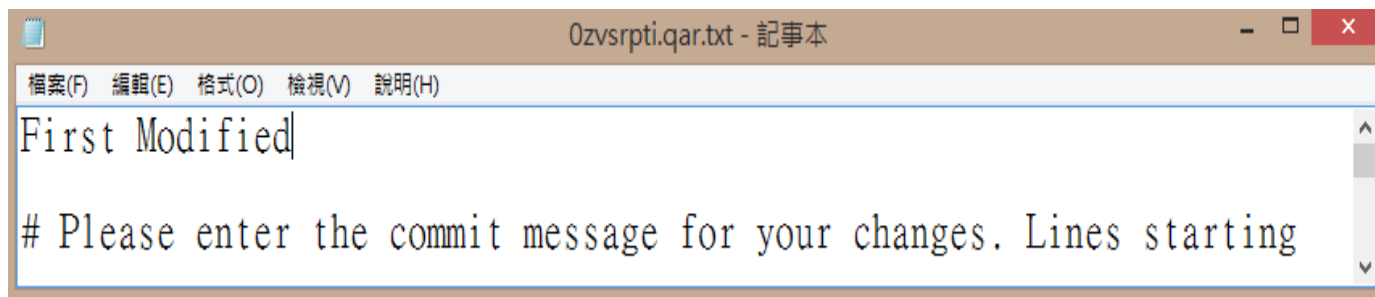


A Notepad window titled "0zvsrpti.qar.txt - 記事本" showing the commit message template. The text is as follows:

```
First
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   Person.java
#
```

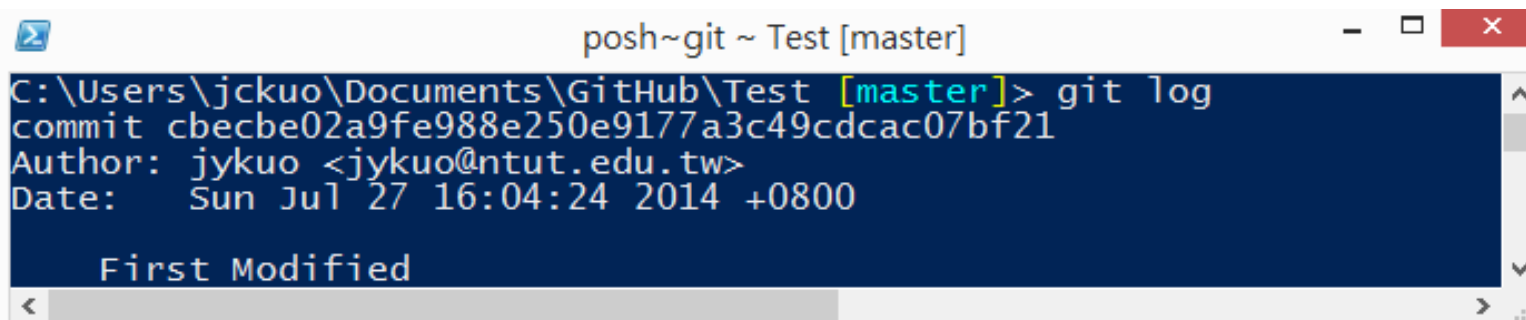

Lab 8.修改簽入的說明₂

- 修改上次簽入說明，'First Modified'



```
0zvsrpti.qar.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
First Modified
# Please enter the commit message for your changes. Lines starting
```

- 結束筆記本編輯，儲存檔案變更。
- 顯示簽入歷史，使用指令：git log，觀察到簽入說明已變更

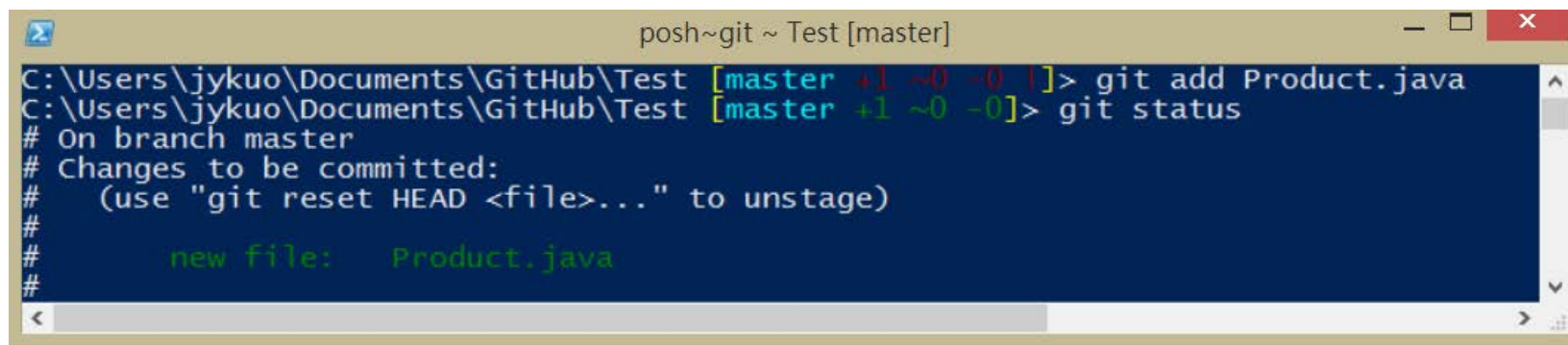


```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master]> git log
commit cbecbe02a9fe988e250e9177a3c49cdcac07bf21
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 16:04:24 2014 +0800

    First Modified
```

Lab 9. 檔案退回工作區 1

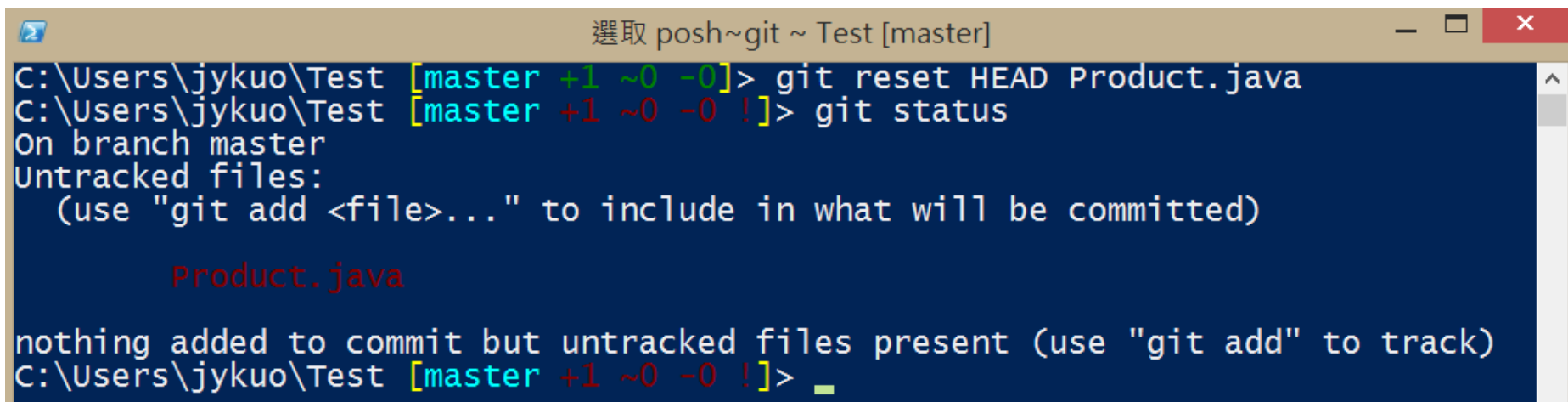
- 依照 Lab 2，編輯一個新檔案 Product.java
- 依照 Lab 4，加入暫存區 `git add Product.java`
- 依照 Lab 3，查看檔案狀態 `git status`

A screenshot of a terminal window with a gold title bar. The title bar text is 'posh~git ~ Test [master]'. The terminal has a dark blue background with white and green text. The commands and output are as follows:

```
C:\Users\jykuo\Documents\GitHub\Test [master +1 ~0 -0 1]> git add Product.java
C:\Users\jykuo\Documents\GitHub\Test [master +1 ~0 -0]> git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   Product.java
#
```

Lab 9. 檔案退回工作區 2

- 將新加入暫存區的檔案，退回工作區
 - 使用指令：git reset HEAD Product.java
 - 使用查看檔案狀態 git status，發現檔案變成紅色



```
選取 posh~git ~ Test [master]
C:\Users\jykuo\Test [master +1 ~0 -0]> git reset HEAD Product.java
C:\Users\jykuo\Test [master +1 ~0 -0 !]> git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        Product.java


nothing added to commit but untracked files present (use "git add" to track)
C:\Users\jykuo\Test [master +1 ~0 -0 !]> _
```

Lab 9. 檔案退回工作區 ³

- `git reset [-- soft | mixed | hard] [<commit tree version>]`
 - 重設目前branch的HEAD到<commit tree version>
 - `--soft` 沒有清除暫存區與工作區，若有修改則保留在暫存與工作區。
 - 保留暫存區和工作區的檔案，可以 `commit` 出不同的 branch 或 version。
 - `--mixed` 清除暫存區，沒有清除工作區，工作區的檔案若有改變將被保留。**預設的狀態選項。**
 - `--hard` 清除暫存區和工作區。
- `git reset HEAD file_name`
 - 清除暫存區內的內容/檔案，保留工作區內的檔案。
 - 到上一次 HEAD 的commit 狀態。

Lab 10. 清除暫存區與工作區檔案 1

- 將剛剛退回工作區的檔案，Product.java，加入暫存區。
 - `git add Product.java`

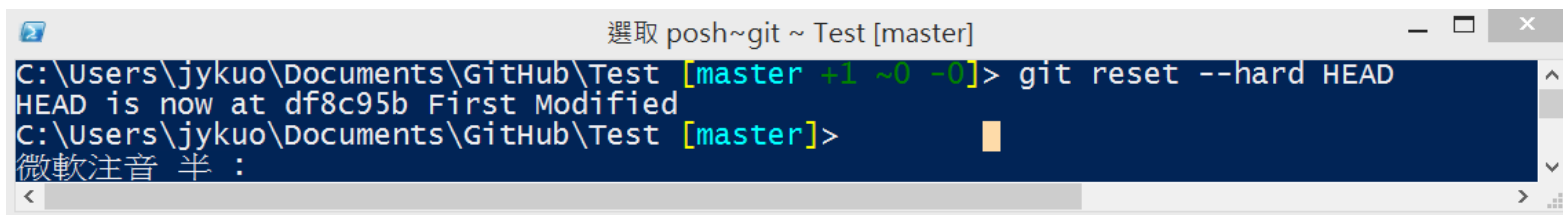


```
選取 posh~git ~ Test [master]
C:\Users\jykuo\Documents\GitHub\Test [master +1 ~0 -0 !]> git add Product.java
C:\Users\jykuo\Documents\GitHub\Test [master +1 ~0 -0]> git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   Product.java

C:\Users\jykuo\Documents\GitHub\Test [master +1 ~0 -0]> _
微軟注音 半 :
```

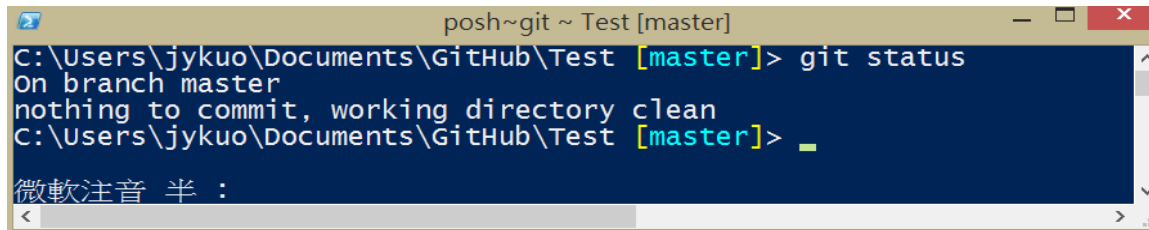
- `git reset --hard HEAD`
 - 清除/重設暫存區內的內容/檔案，清除/重設工作區內的檔案。
 - 恢復任何所做的改變，恢復到上一次的commit。



```
選取 posh~git ~ Test [master]
C:\Users\jykuo\Documents\GitHub\Test [master +1 ~0 -0]> git reset --hard HEAD
HEAD is now at df8c95b First Modified
C:\Users\jykuo\Documents\GitHub\Test [master]>
微軟注音 半 :
```

Lab 10. 清除暫存區與工作區檔案²

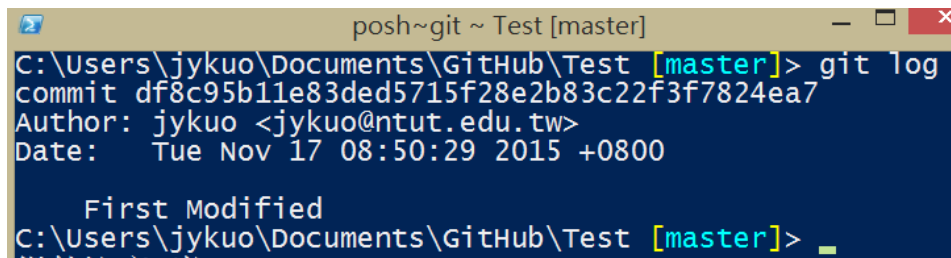
- git status 暫存區與工作區都被重設



```
posh~git ~ Test [master]
C:\Users\jykuo\Documents\GitHub\Test [master]> git status
On branch master
nothing to commit, working directory clean
C:\Users\jykuo\Documents\GitHub\Test [master]> _
```

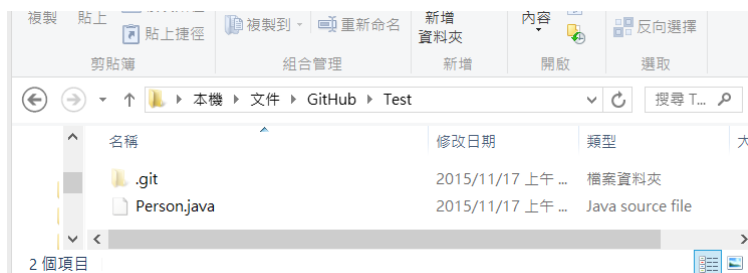
微軟注音 半 :

- git log -p 簽入的版本不變，Product.java在檔案總管中消失。



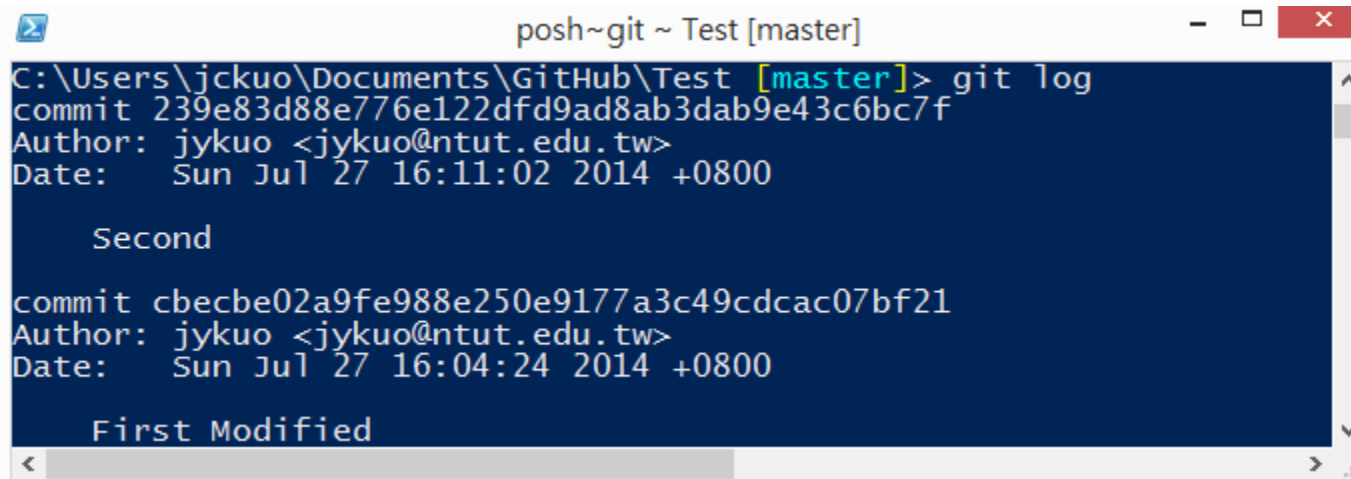
```
posh~git ~ Test [master]
C:\Users\jykuo\Documents\GitHub\Test [master]> git log
commit df8c95b11e83ded5715f28e2b83c22f3f7824ea7
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 08:50:29 2015 +0800

    First Modified
C:\Users\jykuo\Documents\GitHub\Test [master]> _
```



Lab 11. 簽入第二版本

- 用檔案總管，重新new一個Product.java
- 將檔案重新加入暫存區 `git add Product.java`
- 將檔案簽入儲存庫 `git commit Product.java -m 'Second'`
- 查看簽入版本歷史資料 `git log`，發現



```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master]> git log
commit 239e83d88e776e122dfd9ad8ab3dab9e43c6bc7f
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Sun Jul 27 16:11:02 2014 +0800

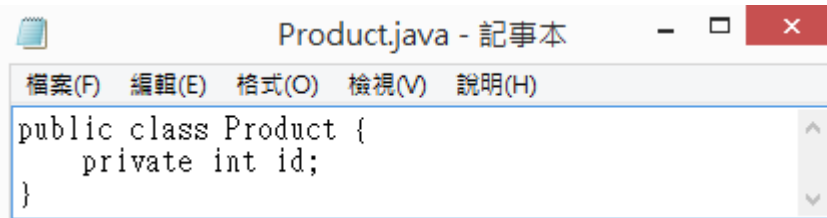
    Second

commit cbece02a9fe988e250e9177a3c49cdcac07bf21
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Sun Jul 27 16:04:24 2014 +0800

    First Modified
```

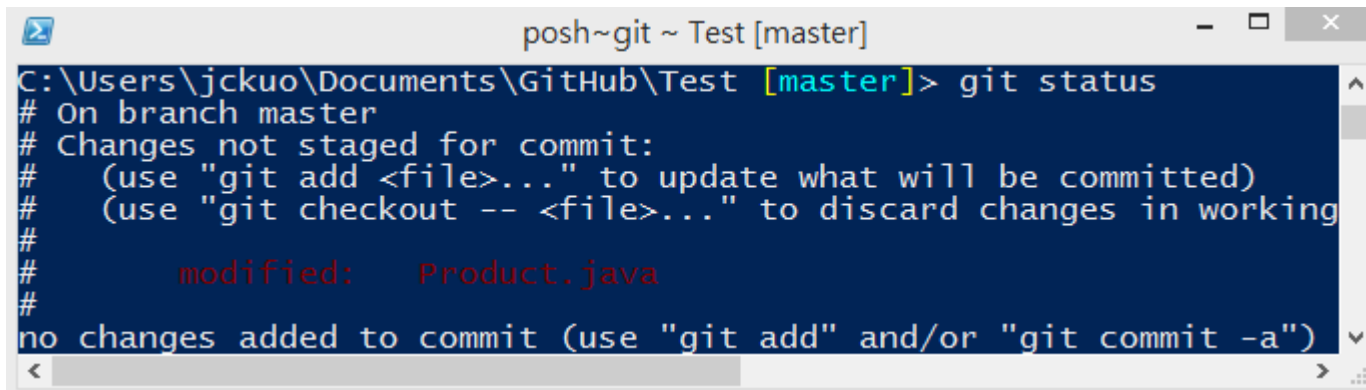
Lab 12. 簽出上一版本檔案 1

- 已經commit的檔案，加以修改，增加 private int id;



```
Product.java - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
public class Product {
    private int id;
}
```

- git status

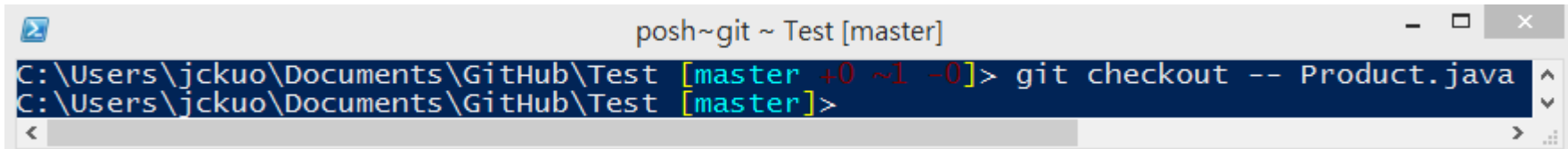


```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master]> git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
#
#       modified:   Product.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

- 顯示紅色，暫存區為modified修改過檔案。

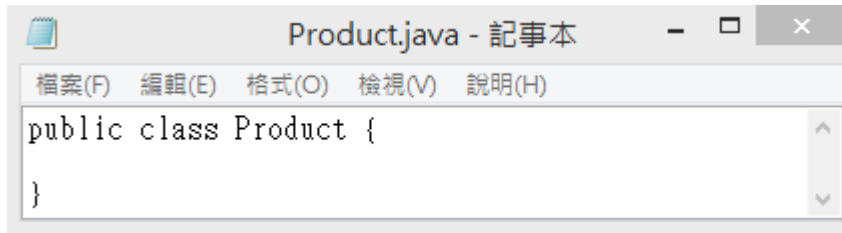
Lab 12. 簽出上一版本檔案 2

- `git checkout -- Product.java`

A terminal window titled "posh~git ~ Test [master]" showing the execution of the command `git checkout -- Product.java`. The prompt is `C:\Users\jckuo\Documents\GitHub\Test [master +0 ~1 -0]>`. The command is entered, and the prompt changes to `C:\Users\jckuo\Documents\GitHub\Test [master]>`.

```
posh~git ~ Test [master]  
C:\Users\jckuo\Documents\GitHub\Test [master +0 ~1 -0]> git checkout -- Product.java  
C:\Users\jckuo\Documents\GitHub\Test [master]>
```

- 開啟Product.java查看

A Notepad window titled "Product.java - 記事本" showing the content of the file. The menu bar includes "檔案(F)", "編輯(E)", "格式(O)", "檢視(V)", and "說明(H)". The text in the editor is `public class Product {` followed by a blank line and a closing brace `}`.

```
Product.java - 記事本  
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)  
public class Product {  
  
}
```

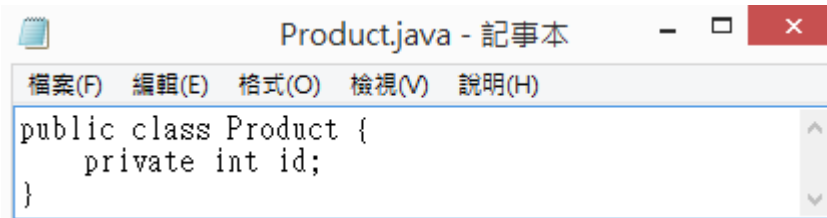
- 恢復檔案為上一本，未修改的檔案狀態。
- 若暫存區有相同檔案，則無法執行此命令。

Lab 12. 簽出上一版本檔案 ³

- 切換到任一快照版本(流水編號)
 - `git checkout` 流水編號
- 從儲存區，取出資料
 - `git checkout -f`
 - 會影響暫存區檔案，不會影響工作區檔案

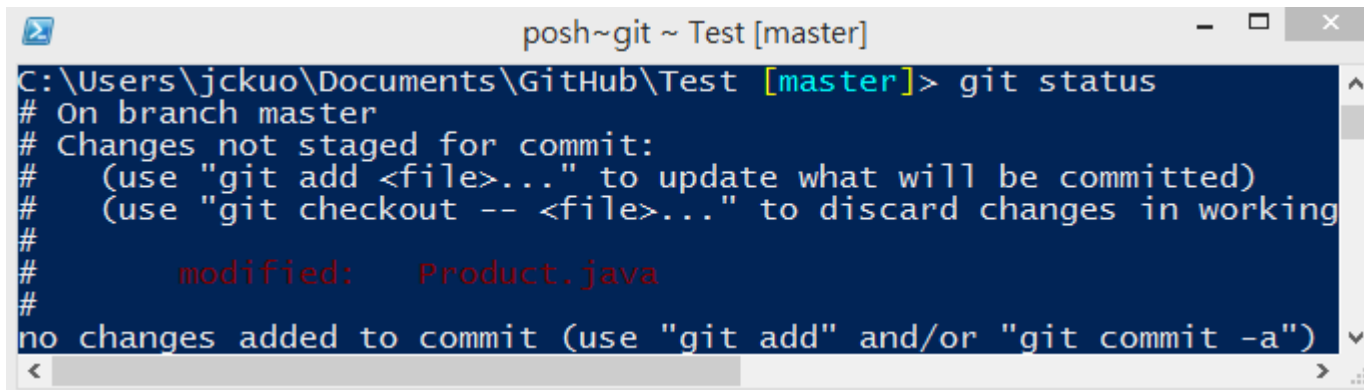
Lab 13. 簽入修改檔案 1

- 已經commit的檔案，加以修改，增加 private int id;



```
Product.java - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
public class Product {
    private int id;
}
```

- git status

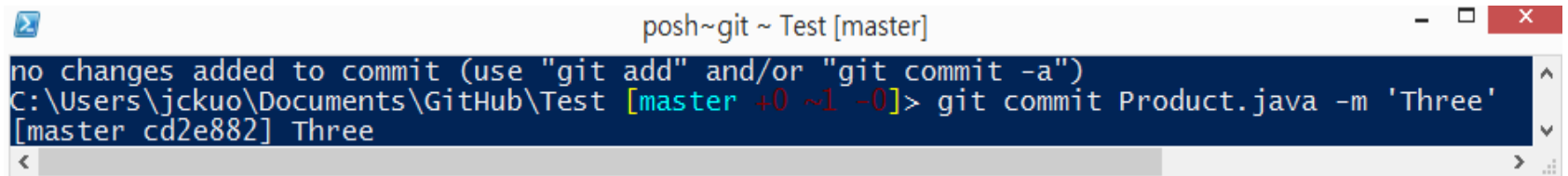


```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master]> git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working
#
#       modified:   Product.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

- 顯示紅色modified，表示暫存區有修改過的檔案。

Lab 13. 簽入修改檔案 2

- 使用指令：git commit Product.java -m 'Three'
 - 直接簽入新的版本

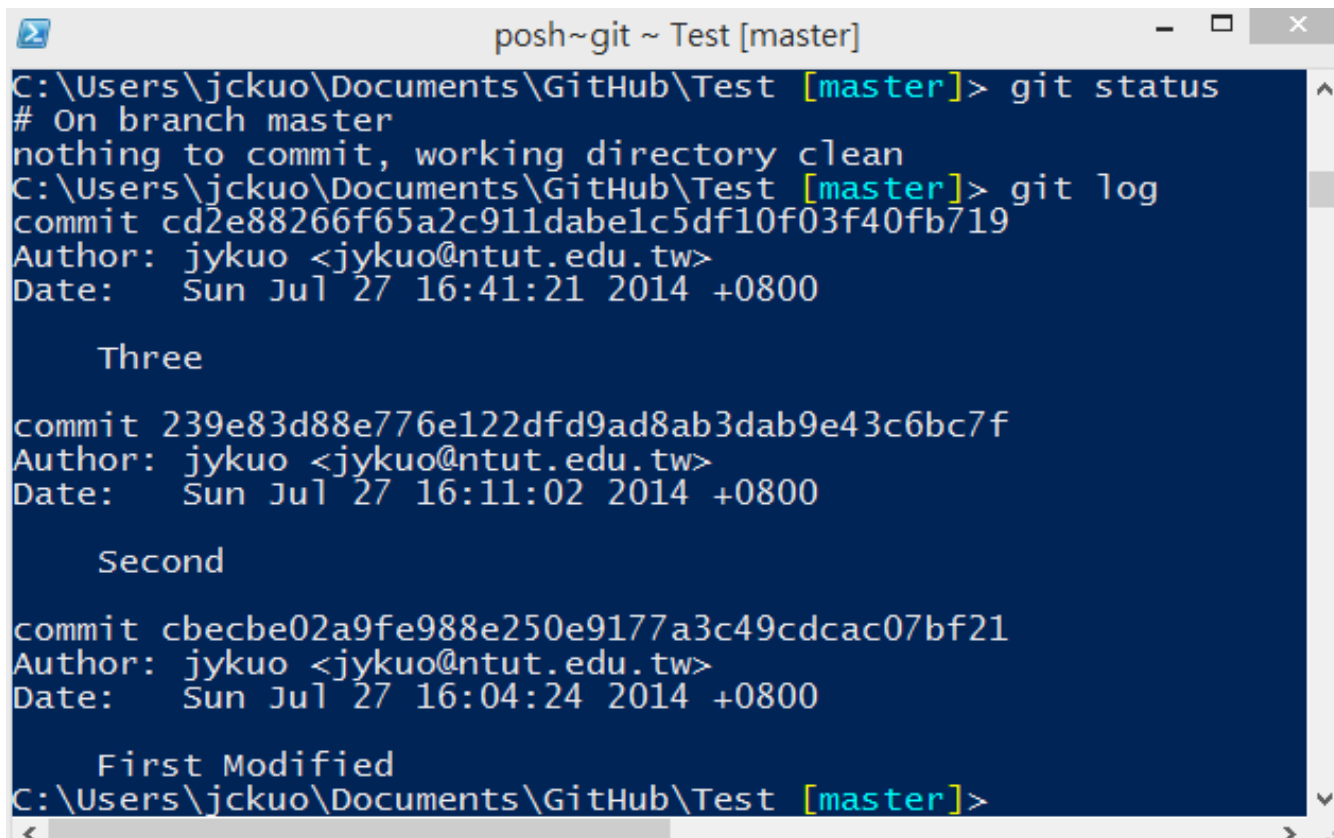
A screenshot of a Windows command prompt window titled "posh~git ~ Test [master]". The window has a dark blue background with white text. The text shows the output of a git command: "no changes added to commit (use 'git add' and/or 'git commit -a')", followed by the command "C:\Users\jckuo\Documents\GitHub\Test [master +0 ~1 -0]> git commit Product.java -m 'Three'", and the result "[master cd2e882] Three".

```
posh~git ~ Test [master]
no changes added to commit (use "git add" and/or "git commit -a")
C:\Users\jckuo\Documents\GitHub\Test [master +0 ~1 -0]> git commit Product.java -m 'Three'
[master cd2e882] Three
```

- 也可使用git add, git commit兩個步驟簽入新的版本。

Lab 13. 簽入修改檔案 3

- 使用 git status, git log 查看暫存區狀態，以及版本歷史資訊。



```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master]> git status
# On branch master
nothing to commit, working directory clean
C:\Users\jckuo\Documents\GitHub\Test [master]> git log
commit cd2e88266f65a2c911dabelc5df10f03f40fb719
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Sun Jul 27 16:41:21 2014 +0800

    Three

commit 239e83d88e776e122dfd9ad8ab3dab9e43c6bc7f
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Sun Jul 27 16:11:02 2014 +0800

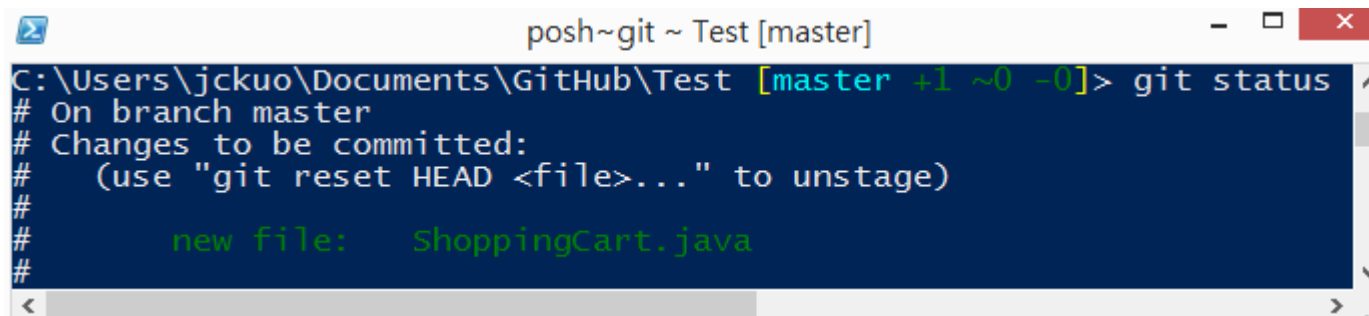
    Second

commit cbecbe02a9fe988e250e9177a3c49cdcac07bf21
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Sun Jul 27 16:04:24 2014 +0800

    First Modified
C:\Users\jckuo\Documents\GitHub\Test [master]>
```

Lab 14. 移除暫存區檔案 1

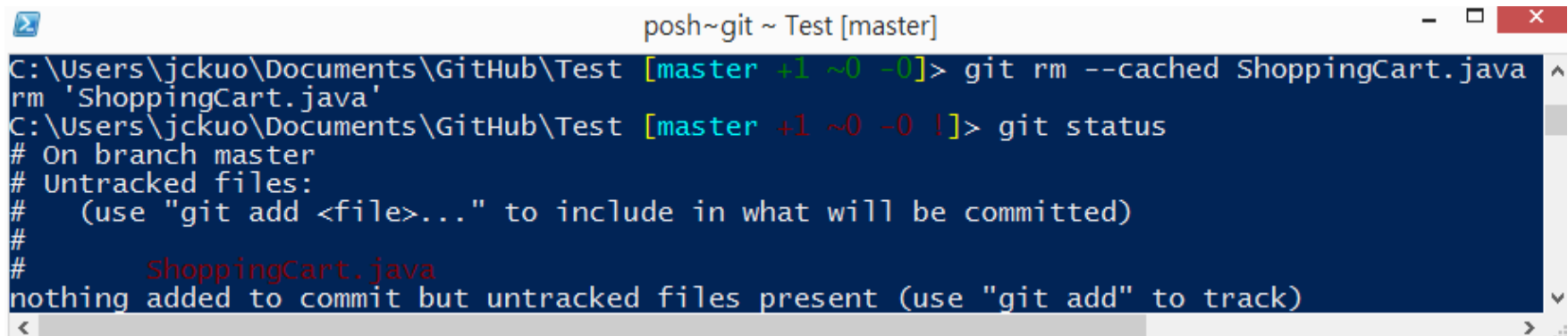
- 依照 Lab 3，編輯一個新檔案 ShoppingCart.java
- 依照 Lab 5，加入暫存區 `git add ShoppingCart.java`
- 依照 Lab 4，查看檔案狀態 `git status`

A screenshot of a Windows command prompt window titled "posh~git ~ Test [master]". The command prompt shows the execution of "git status" in a directory "C:\Users\jckuo\Documents\GitHub\Test". The output indicates the current branch is master and lists "new file: ShoppingCart.java" as a change to be committed. The prompt also includes instructions on how to unstage files using "git reset HEAD <file>...".

```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master +1 ~0 -0]> git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   ShoppingCart.java
#
```

Lab 14. 移除暫存區檔案 2

- 將檔案從暫存區中移除，不刪除工作區原來檔案。
- 指令：git rm --cached ShoppingCart.java

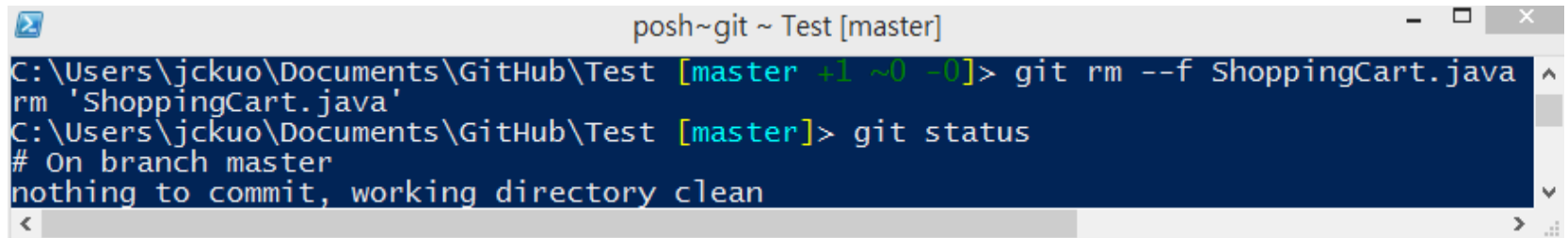


```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master +1 ~0 -0]> git rm --cached ShoppingCart.java
rm 'ShoppingCart.java'
C:\Users\jckuo\Documents\GitHub\Test [master +1 ~0 -0 !]> git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       ShoppingCart.java
nothing added to commit but untracked files present (use "git add" to track)
```

- 使用git status查看，暫存區檔案已經移除到工作區。
- 如果檔案從上次commit後有更改，則效果跟以下指令相同。
 - git reset HEAD Product.java

Lab 14. 移除暫存區檔案 3

- 再將檔案加入暫存區：git add ShoppingCart.java
- 將檔案從暫存區與工作區中強制刪除。
- 指令為：git rm --f ShoppingCart.java

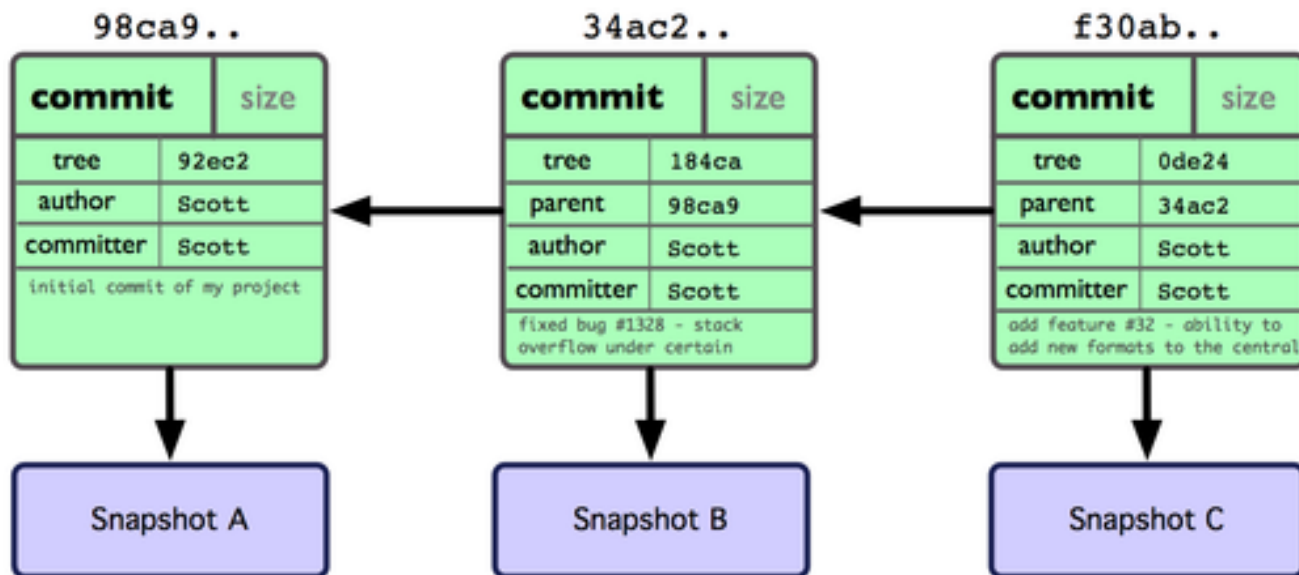
A screenshot of a Windows command prompt window titled "posh~git ~ Test [master]". The window has a dark blue background with white text. The command history shows: "C:\Users\jckuo\Documents\GitHub\Test [master +1 ~0 -0]> git rm --f ShoppingCart.java", "rm 'ShoppingCart.java'", "C:\Users\jckuo\Documents\GitHub\Test [master]> git status", and the output: "# On branch master", "nothing to commit, working directory clean".

```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master +1 ~0 -0]> git rm --f ShoppingCart.java
rm 'ShoppingCart.java'
C:\Users\jckuo\Documents\GitHub\Test [master]> git status
# On branch master
nothing to commit, working directory clean
```

- 查看檔案總管，裡面已經沒有ShoppingCart.java

樹狀歷史管理 1

- Git用樹狀結構紀錄簽入版本的繼承關係
 - 以串列(Linked List)結構，在每個簽入版本記錄其前一版本。
 - 允許簽入版本產生分支以及予以合併。
 - 針對每個檔案版本記錄，稱為快照(Snapshot)，而非只記錄之間差異。



樹狀歷史管理 2

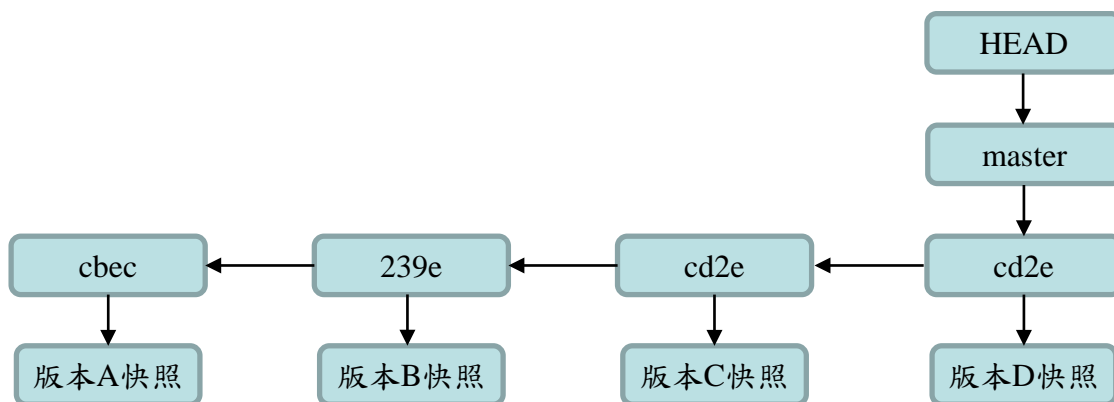
- 標頭(HEAD)指標
 - 記錄目前使用的分支(Branch)。
 - 可隨時新增、切換和刪除分支。
 - 分支可透過 Merge 和 Rebase 指令合併。
- 分支(Branch)指標
 - 指向某特定簽入版本的指標。
 - 使用中的分支，在新簽入版本後自動移動。
 - 每個儲存庫都有一個預設分支，稱為master。
- 說明(Tag)指標
 - 指向某特定簽入版本指標，一種標記說明，不會移動。
- 流水編號
 - 指向某一個快照，本地端快照不會刪除。

樹狀歷史管理 3

- 分支處理機制
 - 每個分支都分開處理一個開發線，例如一位特別客戶需求版本。
 - 將測試中和穩定的版本分開為不同的分支。
 - 將需長時間才能完成的獨立項目拆成另一個分支。
- 若還沒推(Push)到遠端儲存庫，可直接把整個樹退回前次簽入版本
 - `git reset HEAD^`
- 若已 Push 則需使用以下指令
 - `git revert HEAD`

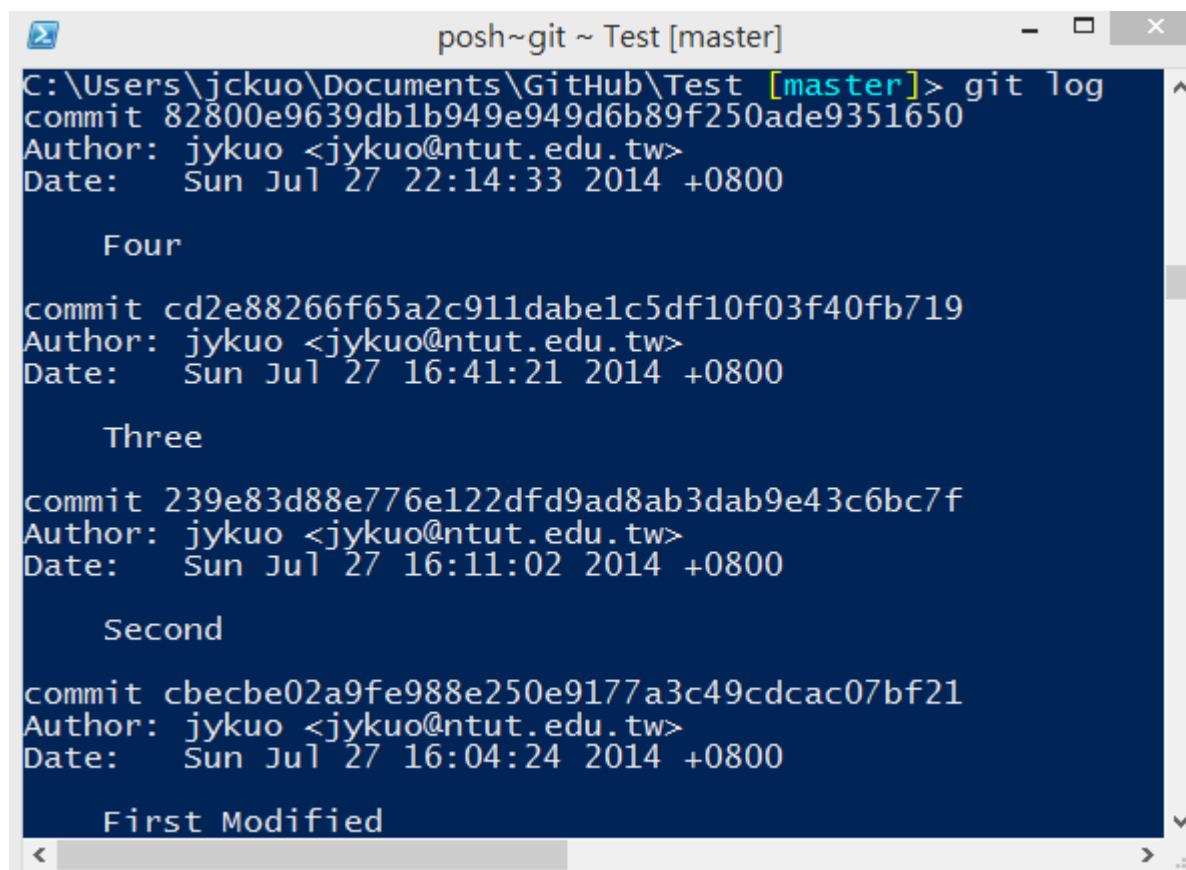
Lab 15. 退回前一版本 1

- 前置條件：在master分支，造出三個簽入儲存區版本。
 - 版本A：空的Person.java
 - 版本B：空的Product.java
 - 版本C：Product.java 內含一行 `private int id;`
 - 版本D：空的ShoppingCart.java



Lab 15. 退回前一版本 2

- 新增檔案：編輯檔案：git add, git commit -m 'message'



```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master]> git log
commit 82800e9639db1b949e949d6b89f250ade9351650
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 22:14:33 2014 +0800

    Four

commit cd2e88266f65a2c911dabe1c5df10f03f40fb719
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 16:41:21 2014 +0800

    Three

commit 239e83d88e776e122dfd9ad8ab3dab9e43c6bc7f
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 16:11:02 2014 +0800

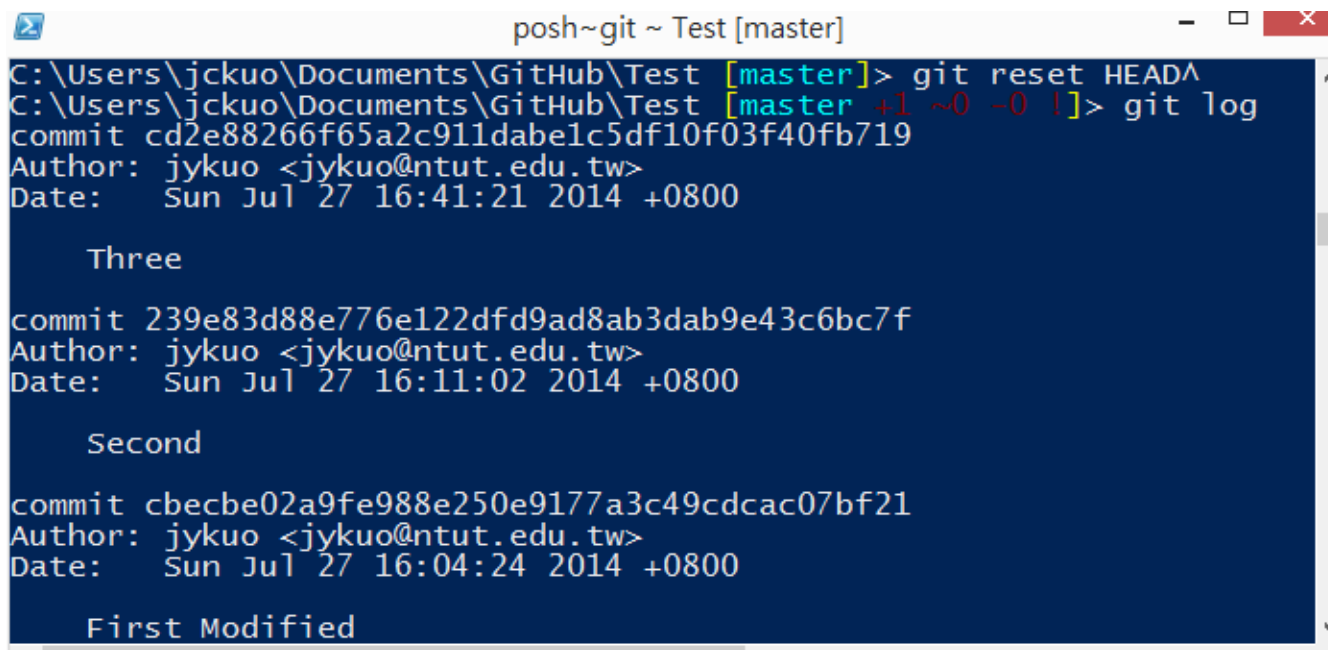
    Second

commit cbecbe02a9fe988e250e9177a3c49cdcac07bf21
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 16:04:24 2014 +0800

    First Modified
```

Lab 15. 退回前一版本³

- 退回前一版本，到版本C。
- 使用指令：`git reset HEAD^`
 - `master`指向前一個版本快照



```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master]> git reset HEAD^
C:\Users\jckuo\Documents\GitHub\Test [master +1 ~0 -0 !]> git log
commit cd2e88266f65a2c911dabe1c5df10f03f40fb719
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 16:41:21 2014 +0800

    Three

commit 239e83d88e776e122dfd9ad8ab3dab9e43c6bc7f
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 16:11:02 2014 +0800

    Second

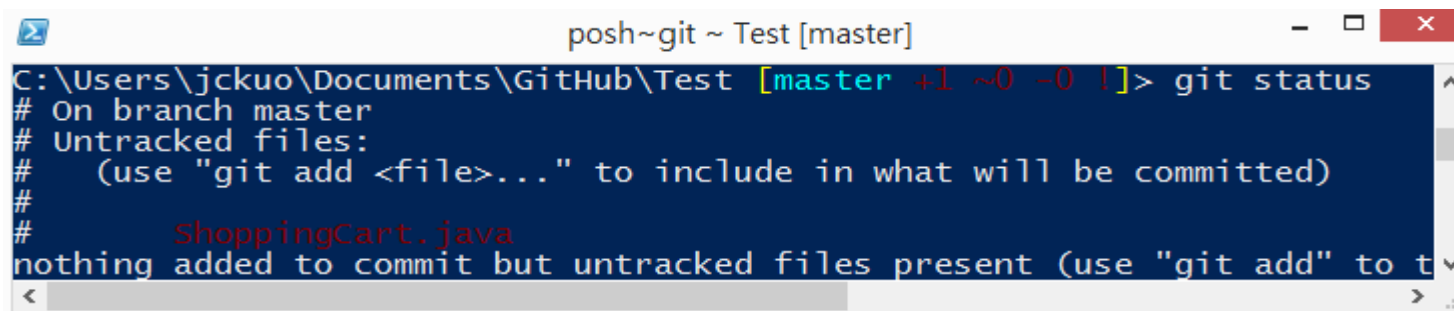
commit cbecbe02a9fe988e250e9177a3c49cdcac07bf21
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 16:04:24 2014 +0800

    First Modified
```

- 使用`git log`指令查看，觀察到剩下三個版本。

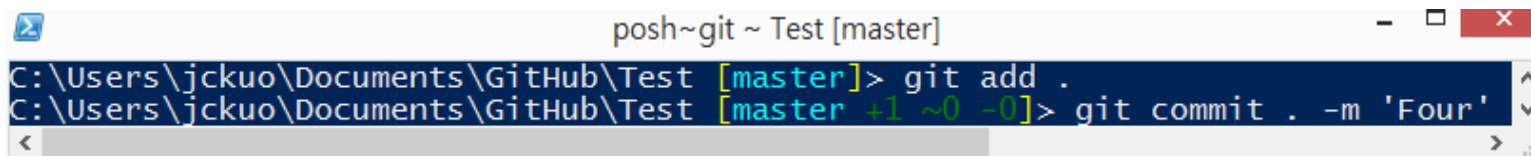
Lab 16. 建立退回前一版本的版本 1

- 接續上一Lab，查看目前暫存區與工作區 git status。



```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master +1 ~0 -0 !]> git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       ShoppingCart.java
nothing added to commit but untracked files present (use "git add" to t
```

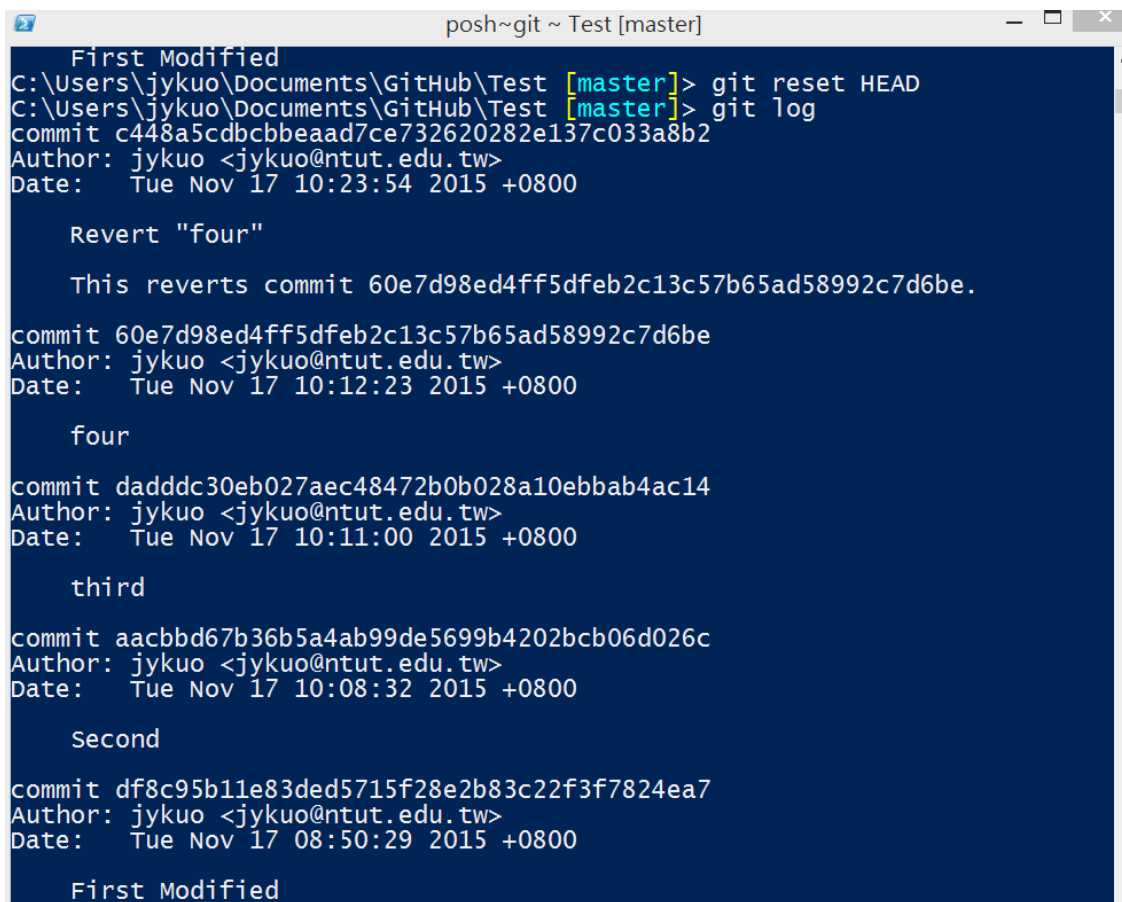
- 重新簽入版本D，git add . 以及 git commit -a -m 'Four'



```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master]> git add .
C:\Users\jckuo\Documents\GitHub\Test [master +1 ~0 -0]> git commit . -m 'Four'
```

Lab 16. 建立退回前一版本的版本 2

- git revert HEAD，出現記事本編輯說明標籤，編輯後關閉。
- git log，製造出第五版，為第四版的前一版(第三版)，SoppingCart.java 消失。



```
posh~git ~ Test [master]
First Modified
C:\Users\jykuo\Documents\GitHub\Test [master]> git reset HEAD
C:\Users\jykuo\Documents\GitHub\Test [master]> git log
commit c448a5cdbcbbbead7ce732620282e137c033a8b2
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 10:23:54 2015 +0800

    Revert "four"

    This reverts commit 60e7d98ed4ff5df52c13c57b65ad58992c7d6be.

commit 60e7d98ed4ff5df52c13c57b65ad58992c7d6be
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 10:12:23 2015 +0800

    four

commit dadddc30eb027aec48472b0b028a10ebbab4ac14
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 10:11:00 2015 +0800

    third

commit aacbbd67b36b5a4ab99de5699b4202bcb06d026c
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 10:08:32 2015 +0800

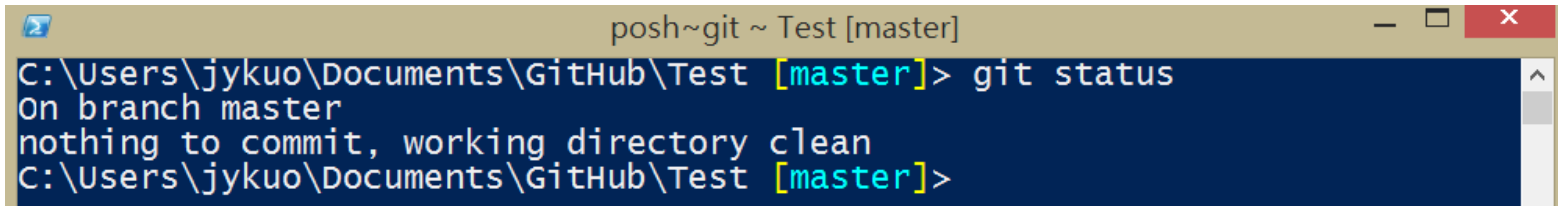
    Second

commit df8c95b11e83ded5715f28e2b83c22f3f7824ea7
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 08:50:29 2015 +0800

    First Modified
```

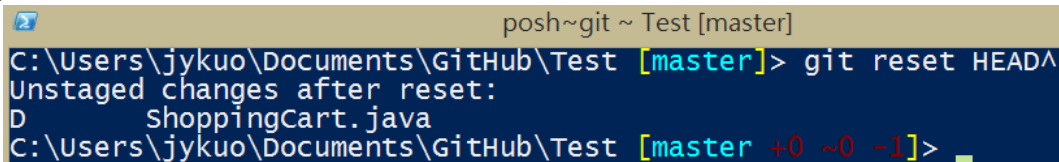

Lab 16. 建立退回前一版本的版本 3

- git status，暫存區與工作區是乾淨的。



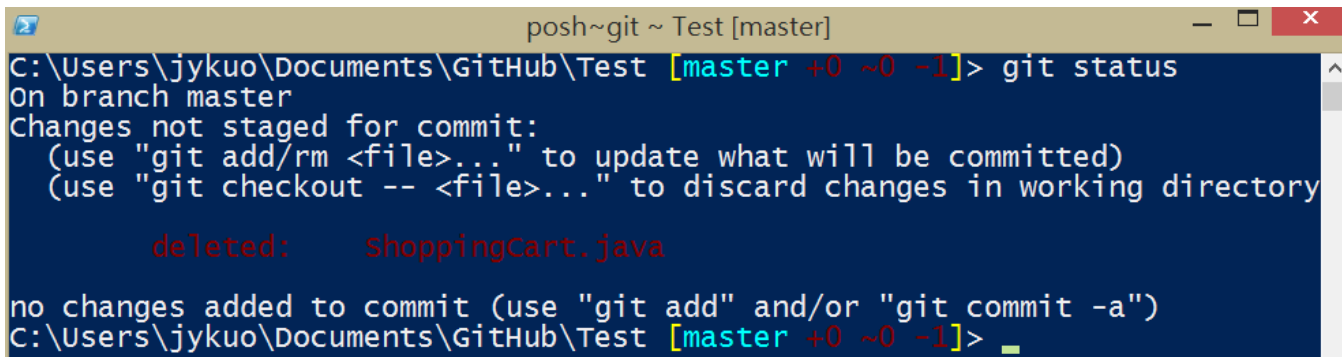
```
posh~git ~ Test [master]
C:\Users\jykuo\Documents\GitHub\Test [master]> git status
On branch master
nothing to commit, working directory clean
C:\Users\jykuo\Documents\GitHub\Test [master]>
```

- git reset HEAD^，退回到前一版本，且工作區是紅色。



```
posh~git ~ Test [master]
C:\Users\jykuo\Documents\GitHub\Test [master]> git reset HEAD^
Unstaged changes after reset:
D   ShoppingCart.java
C:\Users\jykuo\Documents\GitHub\Test [master +0 ~0 -1]> _
```

- git status，工作區顯示刪除ShoppingCart.java，檔案總管沒有ShoppingCart.java。



```
posh~git ~ Test [master]
C:\Users\jykuo\Documents\GitHub\Test [master +0 ~0 -1]> git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

       deleted:    ShoppingCart.java

no changes added to commit (use "git add" and/or "git commit -a")
C:\Users\jykuo\Documents\GitHub\Test [master +0 ~0 -1]> _
```

Lab 16. 建立退回前一版本的版本 4

- git log

```
posh~git ~ Test [master]
C:\Users\jykuo\Documents\GitHub\Test [master +0 ~0 -1]> git log
commit 60e7d98ed4ff5df52c13c57b65ad58992c7d6be
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 10:12:23 2015 +0800

    four

commit dadddc30eb027aec48472b0b028a10ebbab4ac14
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 10:11:00 2015 +0800

    third

commit aacbbd67b36b5a4ab99de5699b4202bcb06d026c
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 10:08:32 2015 +0800

    Second

commit df8c95b11e83ded5715f28e2b83c22f3f7824ea7
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 08:50:29 2015 +0800

    First Modified
C:\Users\jykuo\Documents\GitHub\Test [master +0 ~0 -1]>
```

Lab 16. 建立退回前一版本的版本 5

- 再做一次 git revert HEAD，git log 查看有五個版本。

```
posh~git ~ Test [master]
C:\Users\jykuo\Documents\GitHub\Test [master +0 ~0 ~1]> git revert HEAD
[master a704e7d] Revert "four"
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 ShoppingCart.java
C:\Users\jykuo\Documents\GitHub\Test [master]> git log
commit a704e7dc31ad3b0b5c48e5adec1ffbd31233ffc5
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 10:51:18 2015 +0800

    Revert "four"

    This reverts commit 60e7d98ed4ff5dfef2c13c57b65ad58992c7d6be.

commit 60e7d98ed4ff5dfef2c13c57b65ad58992c7d6be
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 10:12:23 2015 +0800

    four

commit dadddc30eb027aec48472b0b028a10ebbab4ac14
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 10:11:00 2015 +0800

    third

commit aacbbd67b36b5a4ab99de5699b4202bcb06d026c
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 10:08:32 2015 +0800

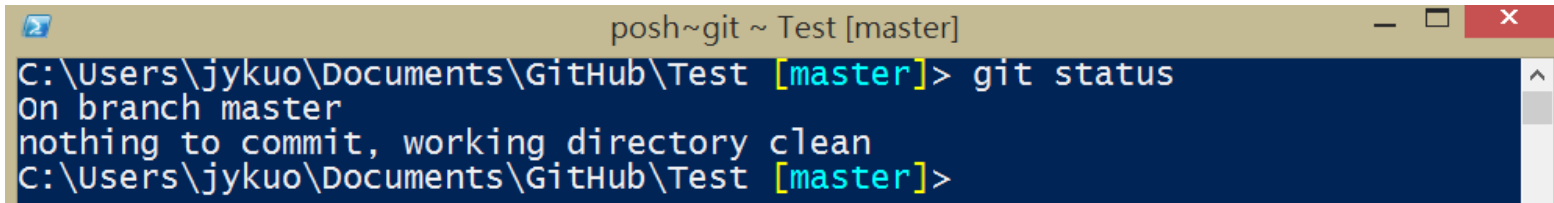
    Second

commit df8c95b11e83ded5715f28e2b83c22f3f7824ea7
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 08:50:29 2015 +0800

    First Modified
```

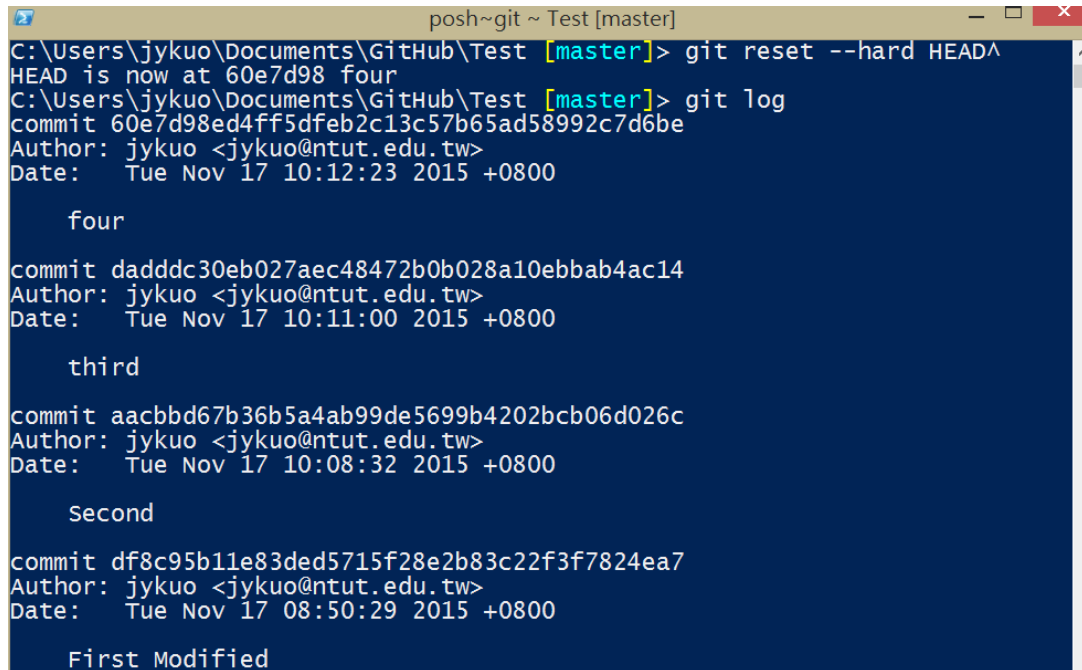
Lab 16. 建立退回前一版本的版本 6

- git status，暫存區與工作區是乾淨的。



```
posh~git ~ Test [master]
C:\Users\jykuo\Documents\GitHub\Test [master]> git status
On branch master
nothing to commit, working directory clean
C:\Users\jykuo\Documents\GitHub\Test [master]>
```

- git reset --hard HEAD^，退回到前一版本，且工作區與暫存區是乾淨的。



```
posh~git ~ Test [master]
C:\Users\jykuo\Documents\GitHub\Test [master]> git reset --hard HEAD^
HEAD is now at 60e7d98 four
C:\Users\jykuo\Documents\GitHub\Test [master]> git log
commit 60e7d98ed4ff5dfef2c13c57b65ad58992c7d6be
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 10:12:23 2015 +0800

    four

commit dadddc30eb027aec48472b0b028a10ebbab4ac14
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 10:11:00 2015 +0800

    third

commit aacbbd67b36b5a4ab99de5699b4202bcb06d026c
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 10:08:32 2015 +0800

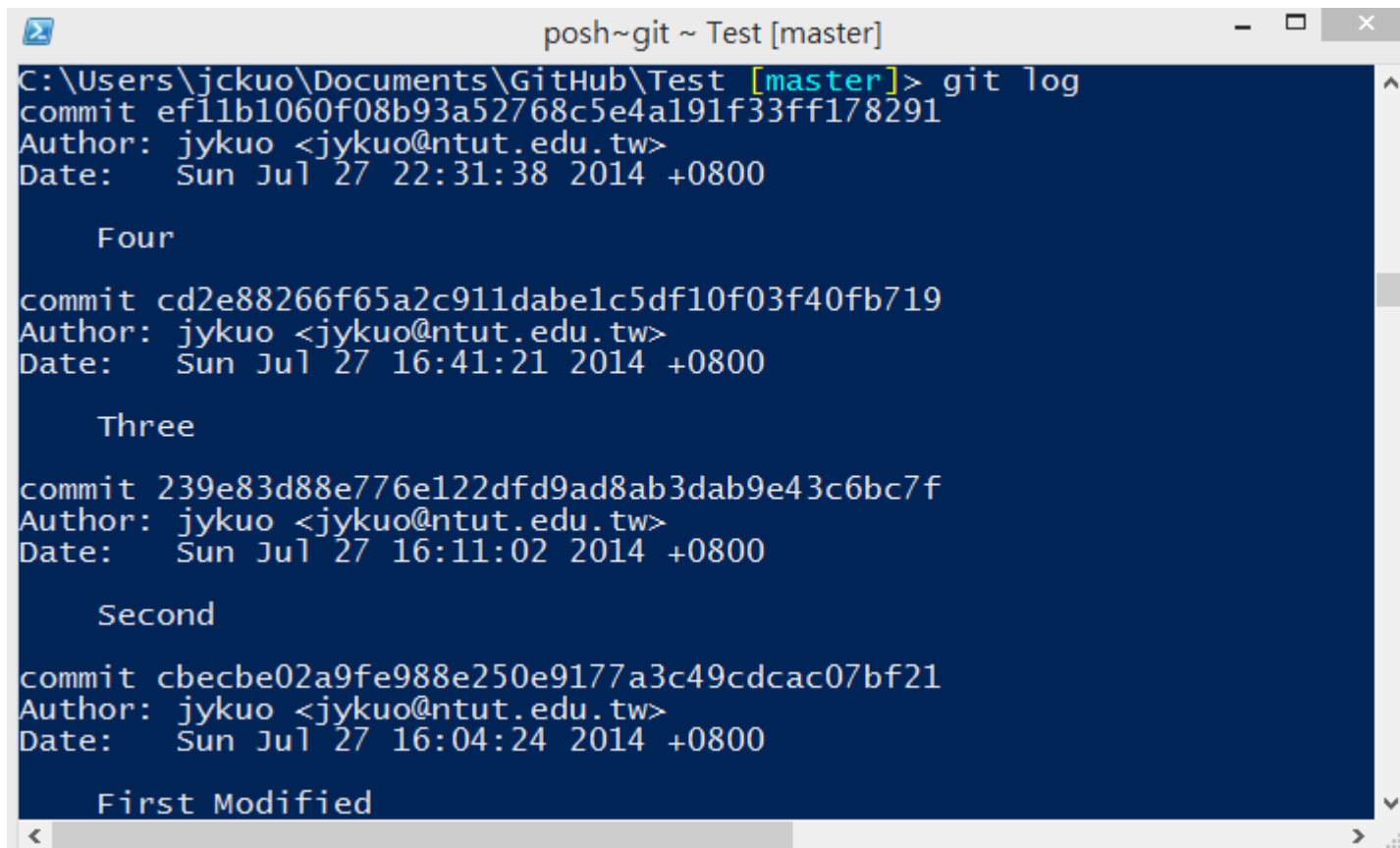
    second

commit df8c95b11e83ded5715f28e2b83c22f3f7824ea7
Author: jykuo <jykuo@ntut.edu.tw>
Date: Tue Nov 17 08:50:29 2015 +0800

    First Modified
```

Lab 17. 退回前兩版本 1

- 使用git log指令查看，觀察到四個版本。



```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master]> git log
commit ef11b1060f08b93a52768c5e4a191f33ff178291
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 22:31:38 2014 +0800

    Four

commit cd2e88266f65a2c911dabe1c5df10f03f40fb719
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 16:41:21 2014 +0800

    Three

commit 239e83d88e776e122dfd9ad8ab3dab9e43c6bc7f
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 16:11:02 2014 +0800

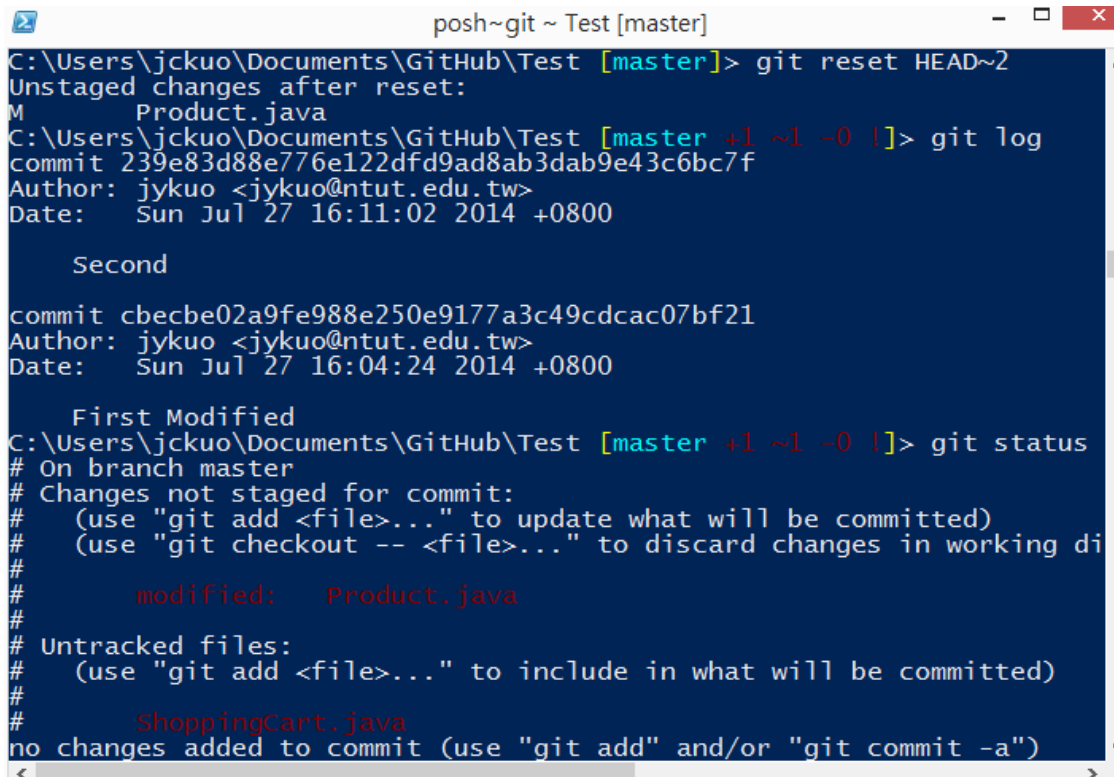
    Second

commit cbecbe02a9fe988e250e9177a3c49cdcac07bf21
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 16:04:24 2014 +0800

    First Modified
```

Lab 17. 退回前兩版本 2

- 使用 `git reset HEAD~2`，退回前兩版本。
- 使用 `git log`，查看簽入版本歷史資訊，觀察到剩版本A, B。
- 使用 `git status` 查看目前工作區、暫存區與版本B差異。



```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master]> git reset HEAD~2
Unstaged changes after reset:
M   Product.java
C:\Users\jckuo\Documents\GitHub\Test [master +1 ~1 -0 !]> git log
commit 239e83d88e776e122dfd9ad8ab3dab9e43c6bc7f
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Sun Jul 27 16:11:02 2014 +0800

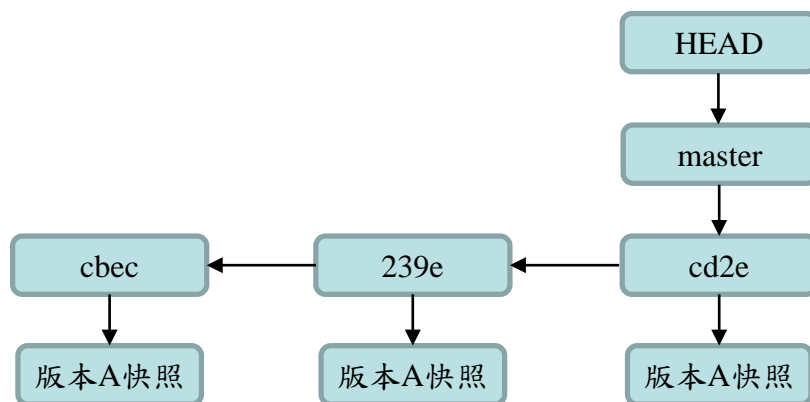
    Second

commit cbecbe02a9fe988e250e9177a3c49cdcac07bf21
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Sun Jul 27 16:04:24 2014 +0800

    First Modified
C:\Users\jckuo\Documents\GitHub\Test [master +1 ~1 -0 !]> git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working di
#
#       modified:   Product.java
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       ShoppingCart.java
no changes added to commit (use "git add" and/or "git commit -a")
```

Lab 18. 建立分支 1

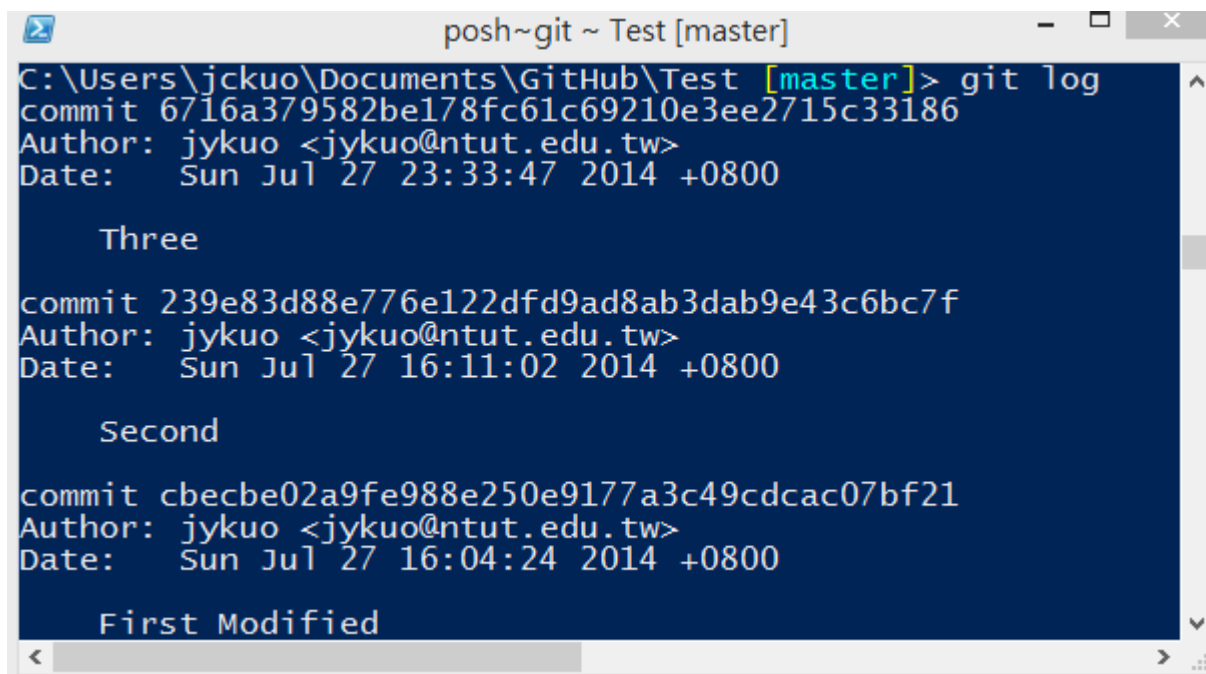
- 前置條件：在master分支，造出三個簽入儲存區版本。
 - 版本一：空的Person.java
 - 版本二：空的Product.java
 - 版本三：Product.java 內含一行 `private int id;`



- 可接續上一Lab.
 - 使用檔案總管刪除ShoppingCart.java
 - 使用指令：`git commit -a -m 'Three'`

Lab 18. 建立分支 2

- 使用指令 `git log` 查看目前簽入版本資訊



```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master]> git log
commit 6716a379582be178fc61c69210e3ee2715c33186
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 23:33:47 2014 +0800

    Three

commit 239e83d88e776e122dfd9ad8ab3dab9e43c6bc7f
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 16:11:02 2014 +0800

    Second

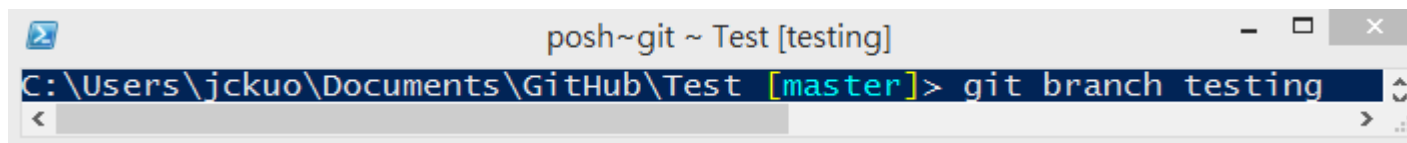
commit cbecbe02a9fe988e250e9177a3c49cdcac07bf21
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 16:04:24 2014 +0800

    First Modified
```

- 分別為First Modified, Second, Three

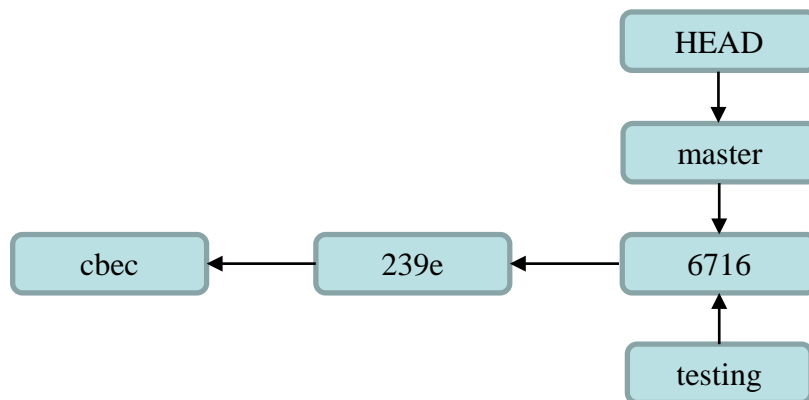
Lab 18. 建立分支 ³

- 建立一個新分支命名為 testing，指向目前簽入版本。
 - 使用指令：git branch testing



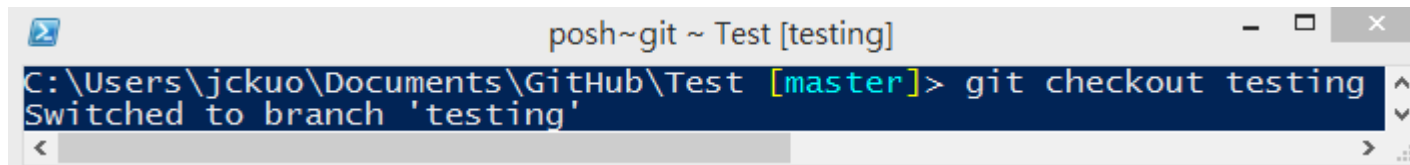
```
posh~git ~ Test [testing]  
C:\Users\jckuo\Documents\GitHub\Test [master]> git branch testing
```

- 簽入版本結構如下圖



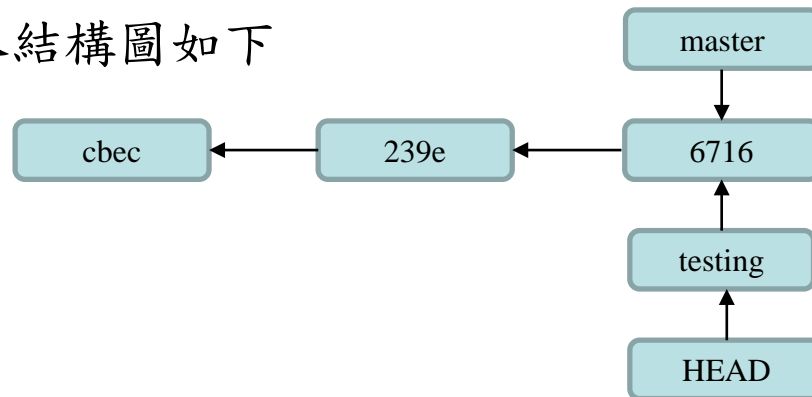
Lab 19. 切換分支

- 接續Lab 15. 切換到新造的分支testing，讓 HEAD 指向 testing。
 - 使用指令：git checkout testing



```
posh~git ~ Test [testing]  
C:\Users\jckuo\Documents\GitHub\Test [master]> git checkout testing  
Switched to branch 'testing'
```

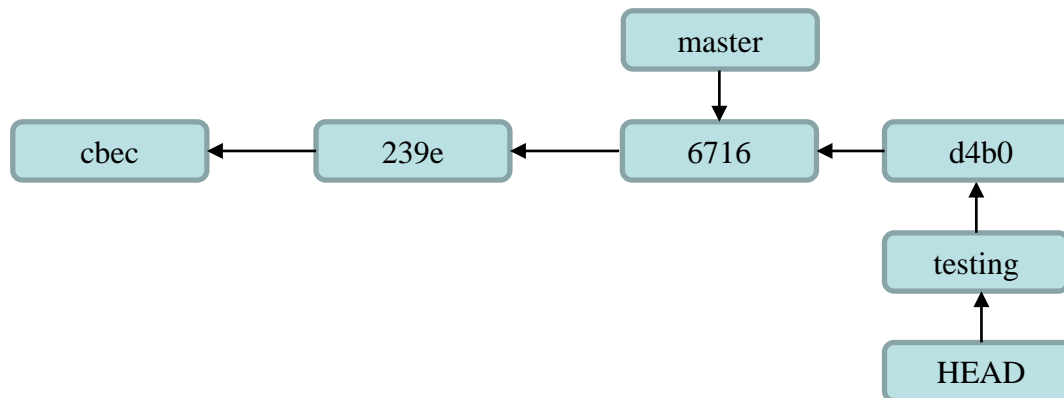
- 此時簽入版本結構圖如下



- 建立新分支testing，並切換到新分支
 - 使用指令：git checkout -b testing

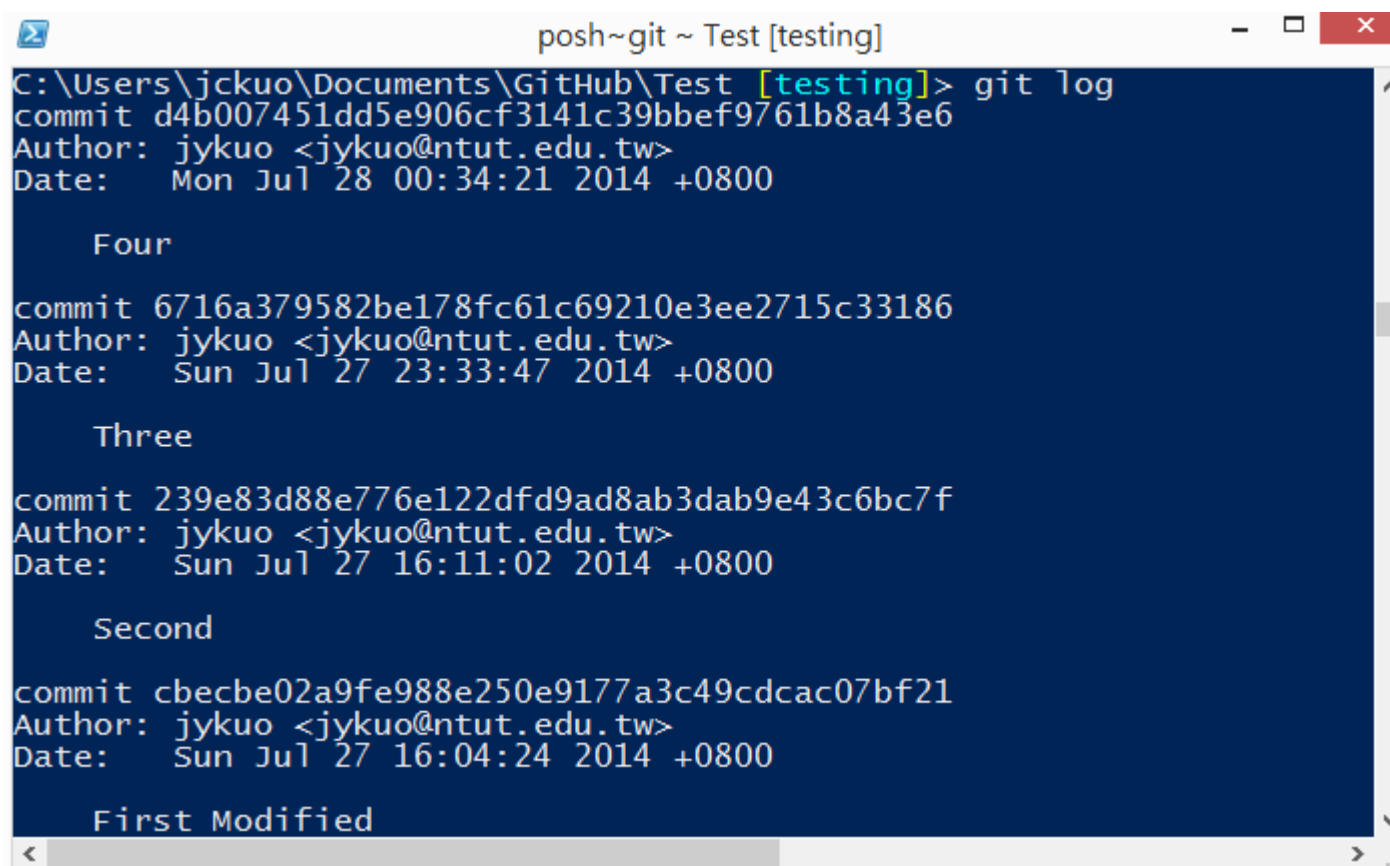
Lab 20. 創造分支版本 1

- 只有 HEAD 指到的 Branch 可以簽入新版本
 - 此時增加新簽入版本，會移動 testing 分支指標指向的位置，而不會移動 master 分支指標。
- 接續 Lab 16.
 - 使用檔案總管，新建一個空的檔案 ShoppingCart.java
 - 簽入新的版本 `git add .` 和 `git commit -a -m 'Four'`
 - 此時簽入版本結構圖如下



Lab 20. 創造分支版本 2

- 使用指令 `git log` 查看目前簽入版本資訊



```
posh~git ~ Test [testing]
C:\Users\jckuo\Documents\GitHub\Test [testing]> git log
commit d4b007451dd5e906cf3141c39bbef9761b8a43e6
Author: jykuo <jykuo@ntut.edu.tw>
Date: Mon Jul 28 00:34:21 2014 +0800

    Four

commit 6716a379582be178fc61c69210e3ee2715c33186
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 23:33:47 2014 +0800

    Three

commit 239e83d88e776e122dfd9ad8ab3dab9e43c6bc7f
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 16:11:02 2014 +0800

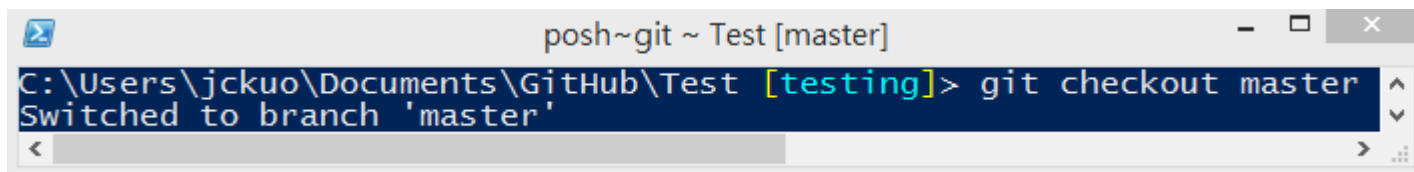
    Second

commit cbecbe02a9fe988e250e9177a3c49cdcac07bf21
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sun Jul 27 16:04:24 2014 +0800

    First Modified
```

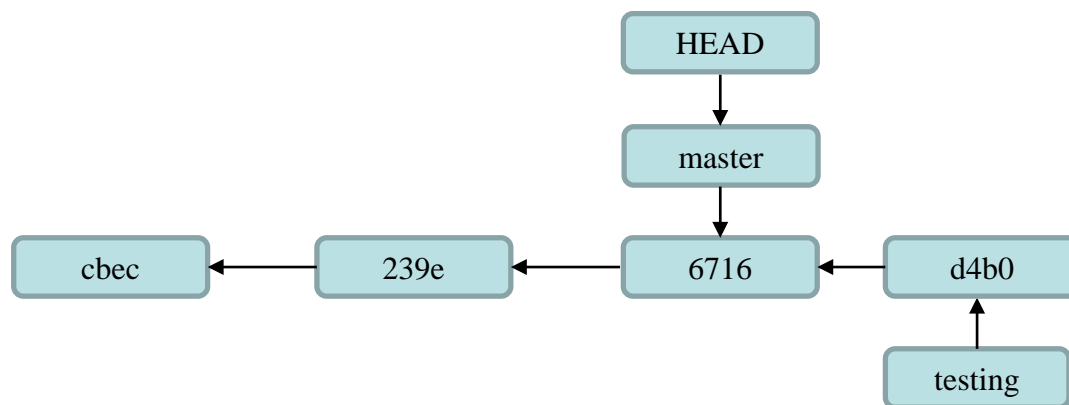
Lab 20. 創造分支版本 3

- 切換到master分支
 - 使用指令：git checkout master



```
posh~git ~ Test [master]  
C:\Users\jckuo\Documents\GitHub\Test [testing]> git checkout master  
Switched to branch 'master'
```

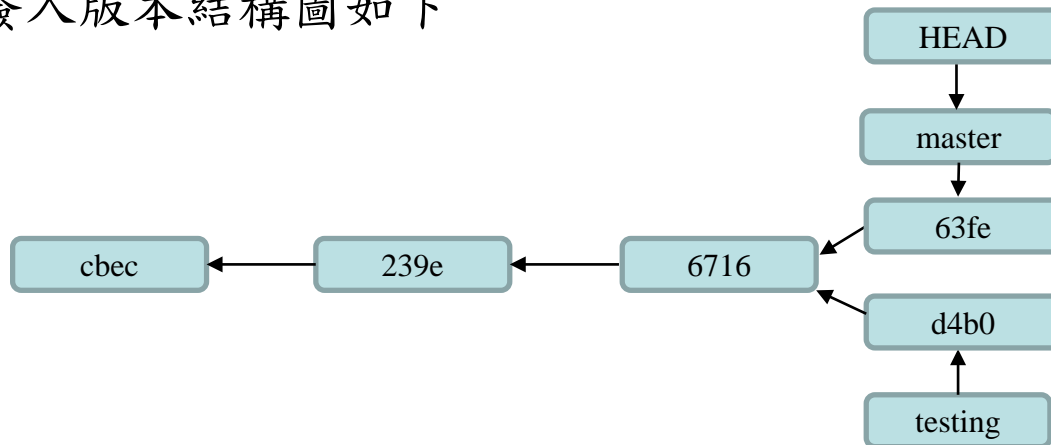
- 此時簽入版本結構圖如下



Lab 20. 創造分支版本 4

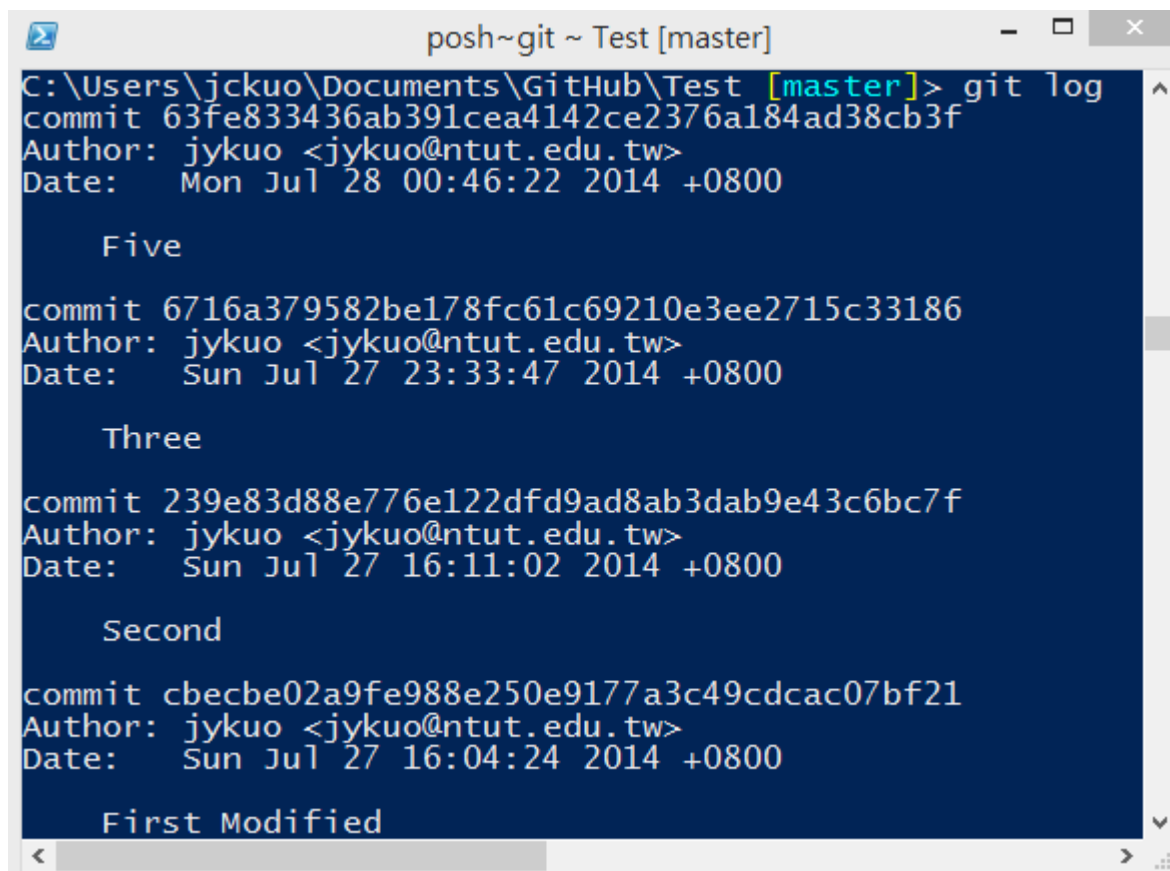
- 在 master 分支執行簽入新版本，產生分岔
 - 在檔案總管，新建一個空的檔案，NotFoundException.java
 - 使用指令：`git add .` 和 `git commit -a -m 'Five'`

- 此時簽入版本結構圖如下



Lab 20. 創造分支版本 5

- 使用指令 `git log` 查看目前簽入版本資訊



```
posh~git ~ Test [master]
C:\Users\jckuo\Documents\GitHub\Test [master]> git log
commit 63fe833436ab391cea4142ce2376a184ad38cb3f
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Mon Jul 28 00:46:22 2014 +0800

    Five

commit 6716a379582be178fc61c69210e3ee2715c33186
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Sun Jul 27 23:33:47 2014 +0800

    Three

commit 239e83d88e776e122dfd9ad8ab3dab9e43c6bc7f
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Sun Jul 27 16:11:02 2014 +0800

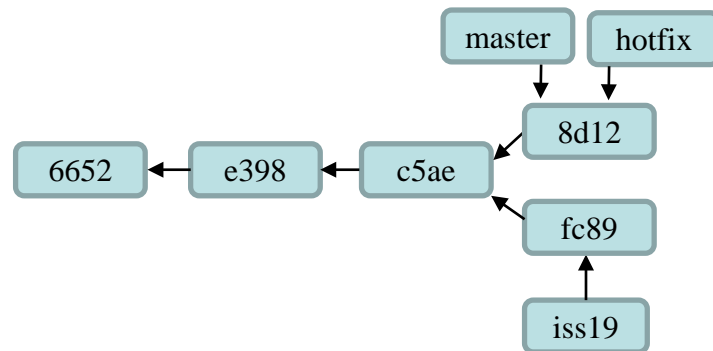
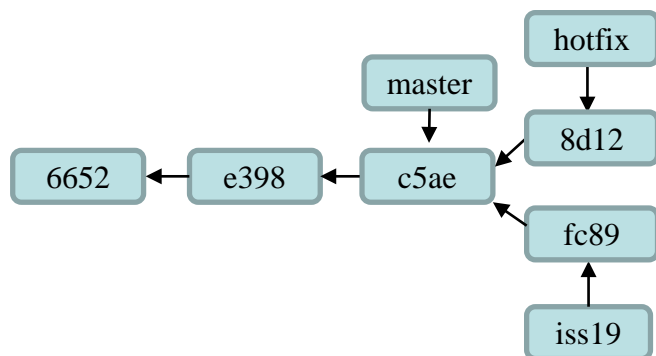
    Second

commit cbecbe02a9fe988e250e9177a3c49cdcac07bf21
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Sun Jul 27 16:04:24 2014 +0800

    First Modified
```

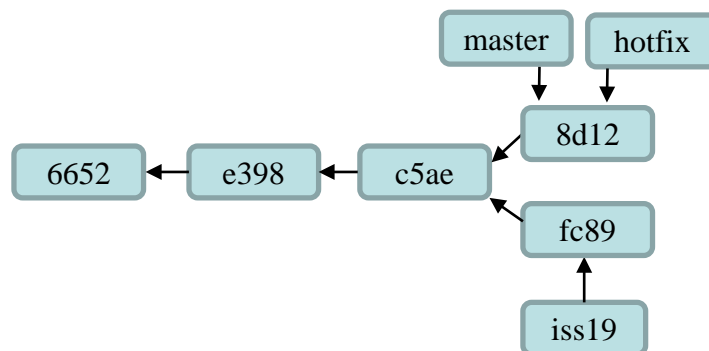
Lab 21. 快速往前合併 1

- 目前發現一個問題，編號為Issue 19
 - iss19 版本用來修正 Issue 19，但尚未寫完。
 - hotfix 版本是為解決現在master迫切問題，已寫好要合併進master。
- 快速往前合併(Fast-forward merge)
 - 若合併目標是master分支往前推的幾個簽入版本，只要把分支往前合併。
- 完成後，把不需要的分支hotfix 刪除。



Lab 21. 快速往前合併 2

- 製作出下面簽入版本結構
 - 重新建立一個目錄lab19，以及儲存庫。
 - 使用檔案總管新建一個空檔案，Person.java，簽入版本。
 - 使用檔案總管新建一個空檔案，Product.java，簽入版本。
 - 使用檔案總管新建一個空檔案，ShoppingCart.java，簽入版本。
 - 造出分支iss19，切換到iss19，修改Product.java，簽入iss19版本。
 - 切換到master。
 - 造出分支hotfix，切換到hotfix，修改Person.java，簽入hotfix版本。



Lab 21. 快速往前合併 3

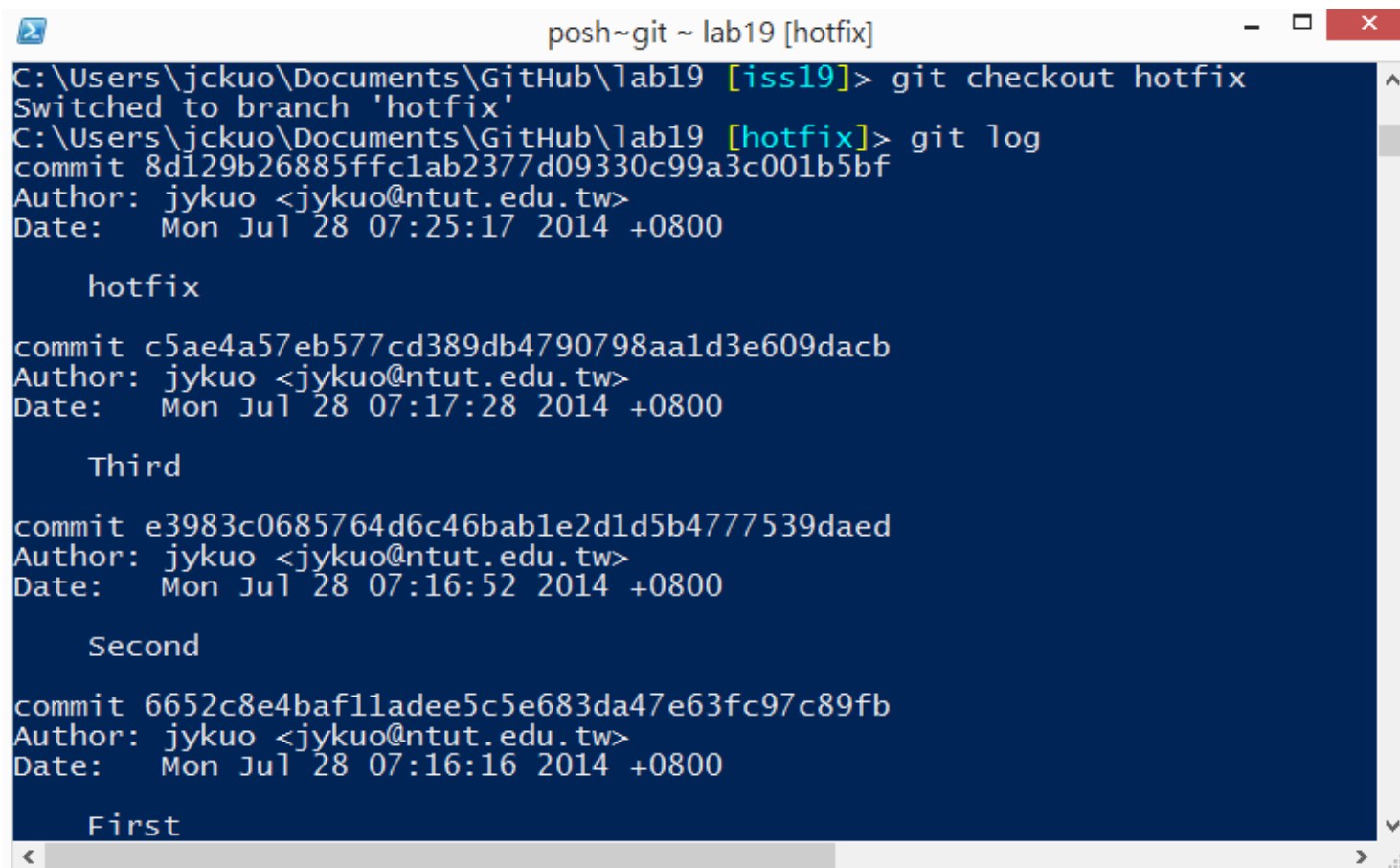
- 重新建立一個目錄lab19，以及儲存庫。
 - `mkdir lab19`
 - `cd lab19`
 - `git config --global user.email "jykuo@ntut.edu.tw"`
 - `git config --global user.name "jykuo"`
 - `git init`
- 使用檔案總管新建一個空檔案，Person.java。
 - 使用 `git add .` 和 `git commit -a -m 'First'`，簽入版本。
- 使用檔案總管新建一個空檔案，Product.java，簽入版本。
 - 使用 `git add .` 和 `git commit -a -m 'Second'`，簽入版本。
- 使用檔案總管新建一個空檔案，ShoppingCart.java，簽入版本。
 - 使用 `git add .` 和 `git commit -a -m 'Third'`，簽入版本。

Lab 21. 快速往前合併 4

- 造出分支iss19，切換到iss19，修改Product.java，簽入iss19版本。
 - `git checkout -b iss19`
 - 修改Product.java，增加private int id;
 - `git commit -a -m 'iss19'`
- 切換到master：`git checkout master`
- 造出分支hotfix，切換到hotfix，修改Person.java，簽入hotfix版本。
 - `git checkout -b hotfix`
 - 修改Person.java，增加private String name;
 - `git commit -a -m 'hotfix'`

Lab 21. 快速往前合併 5

- git checkout hotfix
- git log



```
posh~git ~ lab19 [hotfix]
C:\Users\jckuo\Documents\GitHub\lab19 [iss19]> git checkout hotfix
Switched to branch 'hotfix'
C:\Users\jckuo\Documents\GitHub\lab19 [hotfix]> git log
commit 8d129b26885ffclab2377d09330c99a3c001b5bf
Author: jykuo <jykuo@ntut.edu.tw>
Date: Mon Jul 28 07:25:17 2014 +0800

    hotfix

commit c5ae4a57eb577cd389db4790798aa1d3e609dacb
Author: jykuo <jykuo@ntut.edu.tw>
Date: Mon Jul 28 07:17:28 2014 +0800

    Third

commit e3983c0685764d6c46bab1e2d1d5b4777539daed
Author: jykuo <jykuo@ntut.edu.tw>
Date: Mon Jul 28 07:16:52 2014 +0800

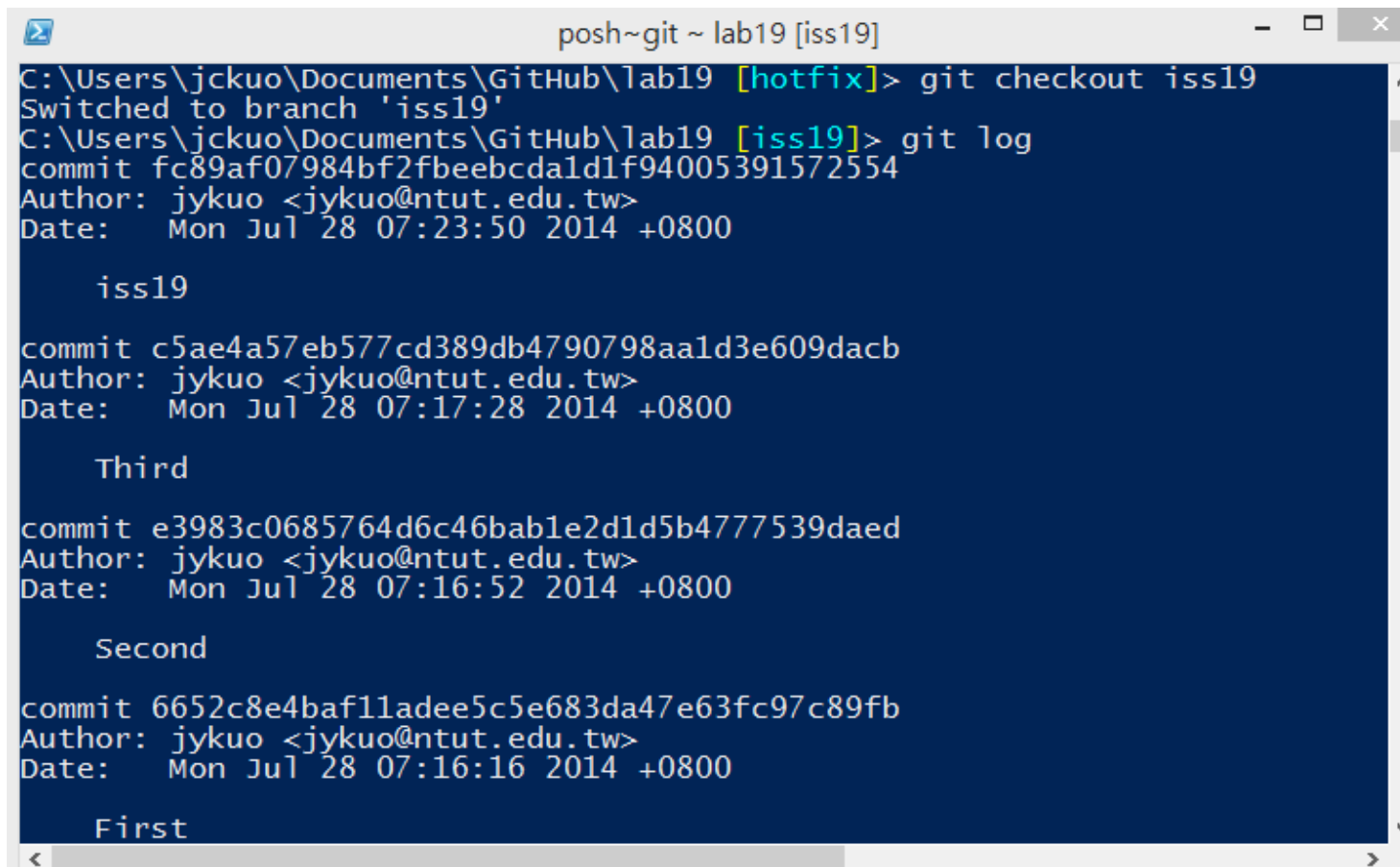
    Second

commit 6652c8e4baf11adee5c5e683da47e63fc97c89fb
Author: jykuo <jykuo@ntut.edu.tw>
Date: Mon Jul 28 07:16:16 2014 +0800

    First
```

Lab 21. 快速往前合併 6

- git checkout iss19
- git log



```
posh~git ~ lab19 [iss19]
C:\Users\jckuo\Documents\GitHub\lab19 [hotfix]> git checkout iss19
Switched to branch 'iss19'
C:\Users\jckuo\Documents\GitHub\lab19 [iss19]> git log
commit fc89af07984bf2fbeebebd1d1f94005391572554
Author: jykuo <jykuo@ntut.edu.tw>
Date: Mon Jul 28 07:23:50 2014 +0800

    iss19

commit c5ae4a57eb577cd389db4790798aa1d3e609dacb
Author: jykuo <jykuo@ntut.edu.tw>
Date: Mon Jul 28 07:17:28 2014 +0800

    Third

commit e3983c0685764d6c46bab1e2d1d5b4777539daed
Author: jykuo <jykuo@ntut.edu.tw>
Date: Mon Jul 28 07:16:52 2014 +0800

    Second

commit 6652c8e4baf11adee5c5e683da47e63fc97c89fb
Author: jykuo <jykuo@ntut.edu.tw>
Date: Mon Jul 28 07:16:16 2014 +0800

    First
```

Lab 21. 快速往前合併 8

- git checkout master
- git log

```
C:\Users\USER\Documents\GitHub\lab21 [master]> git checkout master
Already on 'master'
C:\Users\USER\Documents\GitHub\lab21 [master]> git log
commit 8058f69d91c3c101f5ea87663340f3bfba7c76a1 (HEAD -> master)
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sat Sep 28 16:20:18 2019 +0800

    Three

commit 9a1fcd08bc92d2a492d997d280a218474342cd70
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sat Sep 28 16:19:53 2019 +0800

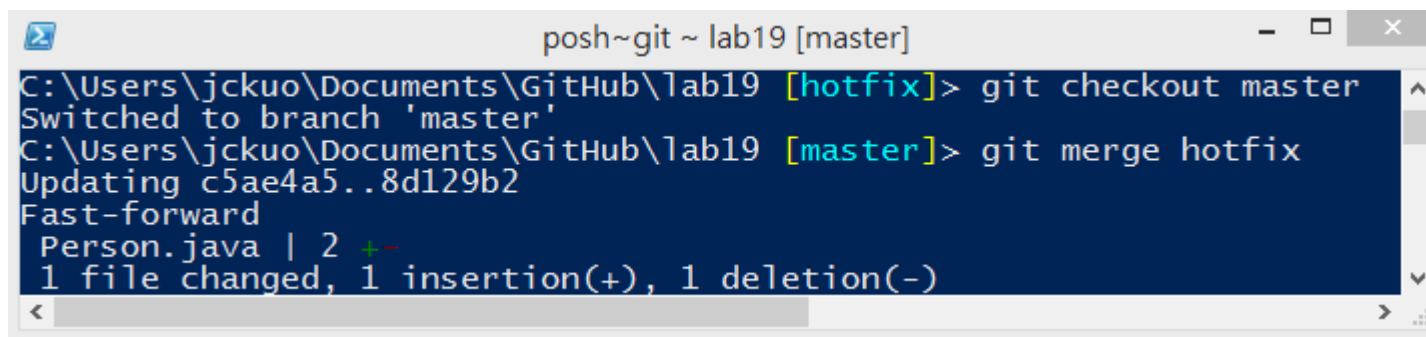
    Second

commit 3412e03efdaa5700d89bb85db7c5a9c4d1625df2
Author: jykuo <jykuo@ntut.edu.tw>
Date: Sat Sep 28 16:19:19 2019 +0800

    First
C:\Users\USER\Documents\GitHub\lab21 [master]>
```

Lab 21. 快速往前合併 9

- 切回 master 後，把hotfix 的變更合併
 - git checkout master
 - git merge hotfix

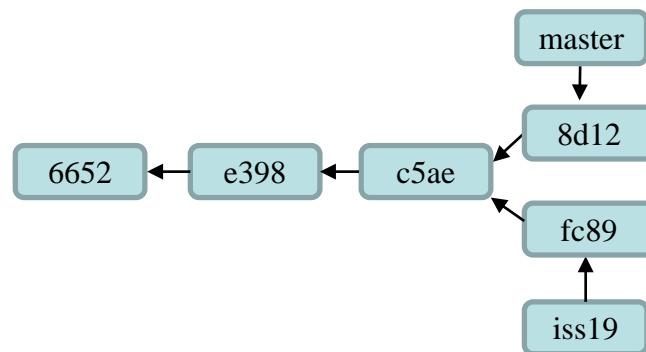
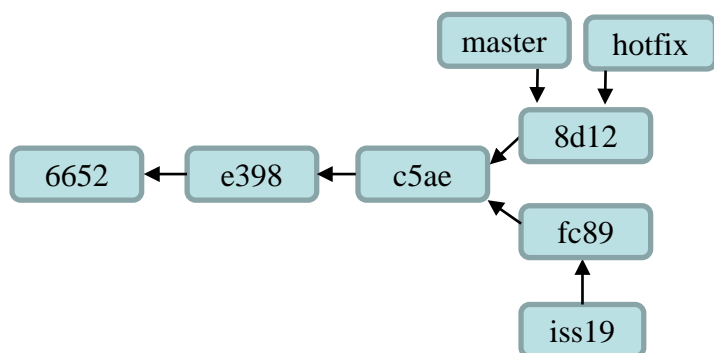
A screenshot of a Windows command prompt window titled "posh~git ~ lab19 [master]". The window shows the execution of two git commands. The first command is "git checkout master", which results in "Switched to branch 'master'". The second command is "git merge hotfix", which results in "Updating c5ae4a5..8d129b2" and "Fast-forward". Below this, a diff is shown for "Person.java" with 2 lines of changes (1 insertion and 1 deletion). The diff shows a line with a green plus sign and a red minus sign. The terminal output is as follows:

```
posh~git ~ lab19 [master]  
C:\Users\jckuo\Documents\GitHub\lab19 [hotfix]> git checkout master  
Switched to branch 'master'  
C:\Users\jckuo\Documents\GitHub\lab19 [master]> git merge hotfix  
Updating c5ae4a5..8d129b2  
Fast-forward  
 Person.java | 2 +-  
 1 file changed, 1 insertion(+), 1 deletion(-)
```

- 因合併對象是往前幾個簽入版本，可透過快速往前合併，把分支往後推以解決問題。

Lab 21. 快速往前合併 10

- 完成後，把不需要的分支刪除。
 - `git branch -d hotfix`

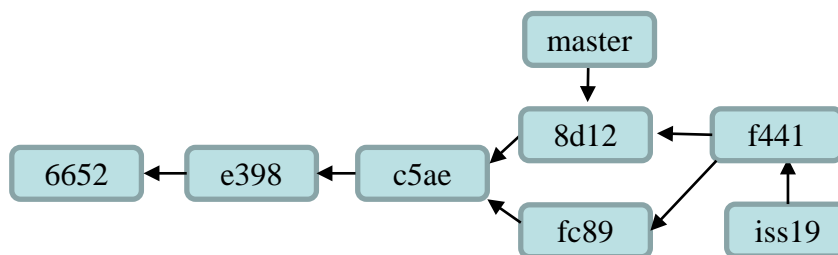


- 觀看目前所有分支，以及目前所在分支。

```
posh~git ~ lab19 [master]
C:\Users\jckuo\Documents\GitHub\lab19 [master]> git branch
  iss19
* master
```


Lab 22. 三向合併修剪 1

- 接續Lab. 19，這時iss53 寫好了，要合併回 master 分支。
- 使用三向合併(3-way Merge) – 修剪(rebase)。
 - 比較祖先和合併對象間的變動，將這些變動合併。
- 合併結果產生新簽入版本。

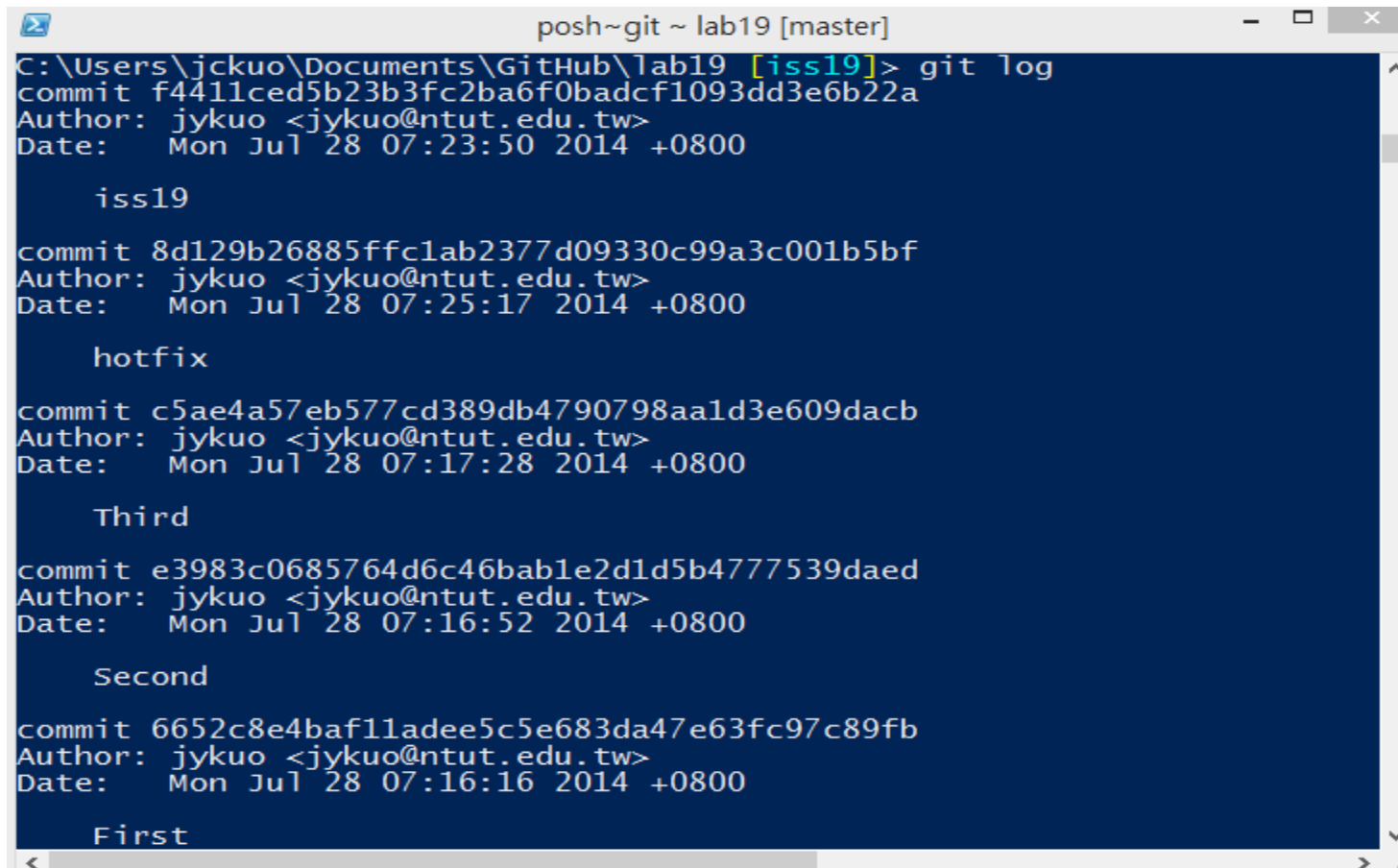


- 使用指令：
 - git checkout iss19
 - git rebase master

```
posh~git ~ lab19 [iss19]
C:\Users\jckuo\Documents\GitHub\lab19 [master]> git checkout iss19
Switched to branch 'iss19'
C:\Users\jckuo\Documents\GitHub\lab19 [iss19]> git rebase master
First, rewinding head to replay your work on top of it...
Applying: iss19
```

Lab 22. 三向合併修剪 2

- 切換到iss19，git checkout iss19，查看iss19合併master結果，git log



```
posh~git ~ lab19 [master]
C:\Users\jckuo\Documents\GitHub\lab19 [iss19]> git log
commit f4411ced5b23b3fc2ba6f0badcf1093dd3e6b22a
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Mon Jul 28 07:23:50 2014 +0800

    iss19

commit 8d129b26885ffc1ab2377d09330c99a3c001b5bf
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Mon Jul 28 07:25:17 2014 +0800

    hotfix

commit c5ae4a57eb577cd389db4790798aa1d3e609dacb
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Mon Jul 28 07:17:28 2014 +0800

    Third

commit e3983c0685764d6c46bab1e2d1d5b4777539daed
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Mon Jul 28 07:16:52 2014 +0800

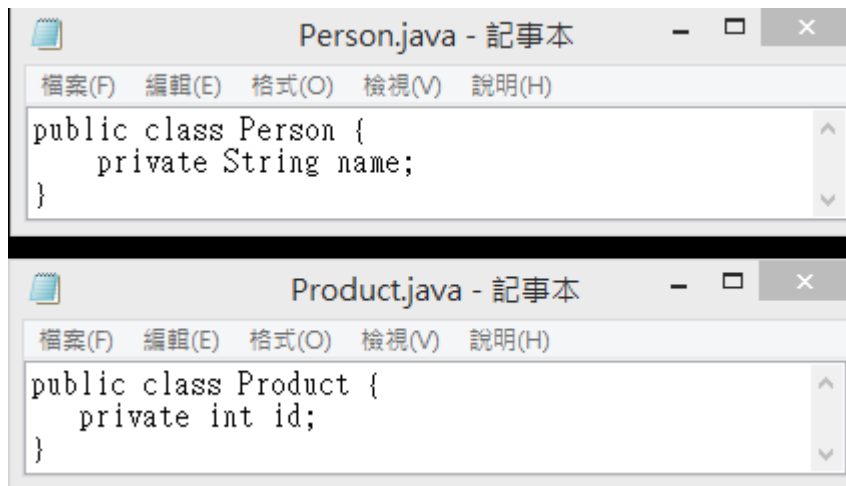
    Second

commit 6652c8e4baf11adee5c5e683da47e63fc97c89fb
Author: jykuo <jykuo@ntut.edu.tw>
Date:   Mon Jul 28 07:16:16 2014 +0800

    First
```

Lab 22. 三向合併修剪 ³

- 切換到issue，使用檔案總管，查看iss19合併master結果



The image shows two Notepad windows stacked vertically. The top window is titled 'Person.java - 記事本' and contains the following code:

```
public class Person {  
    private String name;  
}
```

The bottom window is titled 'Product.java - 記事本' and contains the following code:

```
public class Product {  
    private int id;  
}
```

- 觀察到兩個檔案都修改了。

Lab 22. 三向合併修剪 4

- 三向合併時，若雙方更動同一檔案內容，可能發生衝突。
- Git 遇到衝突會停下來，把未衝突的地方放入暫存區，衝突的地方標示起來。
- 透過 `git status` 找出衝突檔案(unmerged)。
- 打開衝突檔案，手動把衝突處修正後，重新簽入新版本以解決衝突
- rebase 指令
 - pick = 要此 commit，什麼都不改
 - reword = 要此commit，但要改 commit message
 - edit = 要此commit，但要改 commit 的內容
 - squash = 要此 commit，且跟前面那條合併，保留這條的 messages
 - fixup = squash + 使用前面 commit 的 message，捨棄這條 message
 - exec = 執行一條指令

Lab 23. 多 commit 合併修剪 1

- 在一個branch上開發一段時間，commits多又亂
 - 整理commits，將多個commits合併成1個
 - git rebase -i
- 將原本git log
 - a
 - b1
 - b2
- 合併成
 - a
 - b
- 將兩個commits，b1和b2，合併成為b。

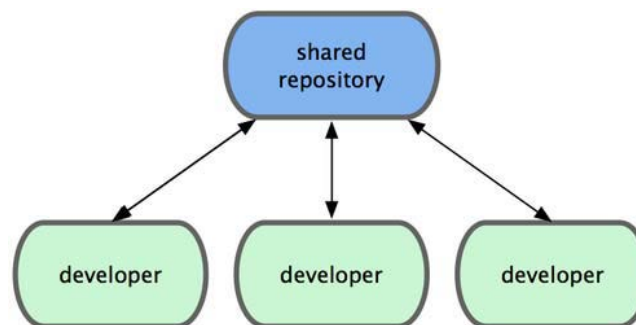
Lab 23. 多 commit 合併修剪 2

- `git rebase -i <commit的流水編號 SHA-1>`
 - 假設 a 的 SHA-1 (用 `git log` 查) 49687a，b1 是 033beb，b2 是 d426a8
 - `git rebase -i 49687a`，按 enter 進到編輯模式，編輯以下指令
 - pick 033beb4 b1
 - squash d426a8a b2
 - 編輯完成，存檔退出，上述指令為，將 b2 合併到 b1。
 - 跳出 commit 標籤說明的編輯視窗，輸入此 commit 的標籤說明，存檔離開。

```
C:\Users\jykuo\Documents\GitHub\test [a]> git rebase -i 49687a
[detached HEAD 14a7115] b rebase b1, b2
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 b1.java
create mode 100644 b2.java
Successfully rebased and updated refs/heads/a.
```

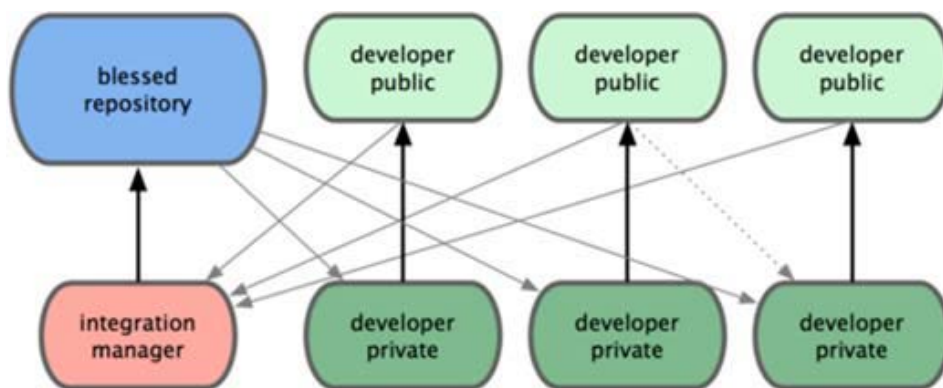
遠端儲存區 1

- 將本地端的儲存庫同步到其他伺服器
 - 多人協作時，需藉此整合。
 - Git 允許多個遠端，做複雜的同步工作。
- 多人協作：單一中心式
 - 類似中心版本控制系統，所有人跟一個共同遠端儲存庫同步。
 - 跟中心式相比，遇到衝突時，可透過 Merge 和 Rebase 指令解決，不會有卡住情形。
 - 適合人數與規模較小的情況



遠端儲存區 2

- 多人協作：整合管理員式
 - 有一個「正統(Blessed)」儲存庫存放最終版本。
 - 每個開發者個別整理後，傳到自己公開儲存庫。
 - 此模式為 GitHub 或 BitBucket 等系統的 Fork 與 Pull Request 機制。
 - 開發者可把正統版 Fork 下來，改好後送 Pull Request 給管理員整合。



遠端伺服器安裝

- 使用現有網路服務：GitHub、Bitbucket
- 安裝 Linux 主機，使用 Gitis 或 Gitolite 架設 Remote Server
- 透過 NFS 檔案系統，或者 Dropbox 作為 Remote。
- Git 提供四種 Remote 協定
 - Local：以本機路徑作為 Remote 目標，適用於架設 NFS 時。
 - /paths/to/remote.git 或 file:///opt/git/project.git
 - SSH：可讀寫，透過 SSH Key 進行認證。
 - username@server.tld:project.git 通常使用 SSH，方便安全。
 - Git：連接網路的協定，唯讀，但速度快，
git://server.tld/project.git
 - HTTP(S)：可讀寫，速度較慢，適用公司內有防火牆，只有 HTTP(S) Port 對外開放的環境，<http://server.tld/project.git>

遠端儲存庫複製

- 將程式碼從指定點下載到本地端儲存庫
 - 使用指令：`git clone git@github.com:littlebtc/myproject`。
 - 遠端儲存庫若有分支，前面加上 遠端伺服器儲存庫分支名字識別。
 - `origin/master` 指 `origin` 這個遠端儲存庫中的 `master` 分支。

相關指令 1

- `git branch`：顯示目前所有的 Branch，用星號標示目前所在
- `git branch -v`同時顯示各 Branch 所在的Commit
- `git branch --merged`：顯示已合併到目前Branch 其他 Branch
 - 把沒用到的 Branch 找出來刪掉
- `git branch --no-merged`：顯示未合併到目前Branch 的其他 Branch
- `git tag v1.0`：把現在的 Commit 加上名為 v1.0的 Tag
- `git tag v1.0 abcde`：對代碼為 abcde 的Commit 加上 v1.0 Tag
- `git tag -d v1.0`：刪除名為 v1.0 的 Tag
- annotated Tags，可在 Tag 上加注作者資訊和認證機制。
- `git push --tags` 把 Tag 給一起Push 上去
- `git fetch`：將 Remote 更新到最新版本。
- `git pull`：進行 Fetch 後，將 Remote Branch 的變動合併進本機 Branch。
- `git push`：將本機的 Branch 變動合併到 Remote Branch。
- 在未設定 Remote 的 Repository 中，新增 Remote 後Push
 - `git remote add origin git@github.com:littlebtc/myproject.git`
 - `git push -u origin master`

相關指令 2

- 觀看目前跟上一版本差異 `git diff`
- 觀看stage跟上一版本差異 `git diff -cached`
- 查看歷史紀錄 `git log -stat`, `git log --pretty=format:"%h - %an, %ar:%s"`
- stage狀態還原到unstage(staged -> modified) `git reset HEAD <file>`
- modified -> unmodified `git checkout -- <file>`
- 修改最後一次 vommit log `git commit --amend`
- 新增遠端 Server
 - Format: `git remote add [short_name] [url]`
 - `git remote add orgin git@xxxxxx`
- 上傳變更檔案到 server
 - Format: `git push [short_name] [branch_name]`
 - `git push -u orgin master`
- 直接下載檔案Format: `git fetch [short_name]`
 - `git fetch origin`
- 下載檔案並且merge
 - `git pull [short_name] [branch_name]`, `git pull origin master`
 - `git pull = git fetch + git merge`, `git fetch origin + git merge origin/master`

相關指令 3

- 列出既有標籤 `git tag -l`
- 新增標籤 `git tag -a v1.4 -m 'my version 1.4'`
- 上傳標籤 `git push origin v1.4`, `git push origin --tags` (上傳所有標籤)
- 刪除標籤 `git tag -d <tagname>`, `git push origin :refs/tags/v1.4`
- 列出全部分支 `git branch -a`
- 詳細列出 branch, `git branch -v`
- 列出已經 merge 的 branch, `git branch -merged`
- 列出尚未 merge 的 branch, `git branch --no-merged`
- 上傳 branch, Format: `git push origin [branch_name]`
- 刪除 branch, Format: `git push origin :[branch_name]`, `git push origin :hotfix`