Basic Storage Design Plan for Gitlet

**Overview**

* “Purgatory” is a Set that holds all of the serialized files that have been “added” but not “committed”
* “Commit” Object will NOT be a simple object, but rather something that I create... Responsibilities include:

1. Containing an UN-serialized set of files

2. Keeping track of the time stamp for the files

* “Peter” will be a function that assigns a key to, and finds a place for, the “Commit” objects in Heaven (puts objects in the right TreeMap in Heaven)... and then when he is all done he clears purgatory object (will probably be the most difficult part of this assignment)

* “Heaven” will be a HashTable holding a TreeMap of “Commit” objects

**Sequential Example**

For each of the commands below, describe in detail what files on your computer change after the command is executed. Only list changes made to files. Do not list details regarding the reading of files.

**$ java Gitlet init**

* If it does not already exist, a new folder called .gitlet is created in the current directory.
* If it does not exist a new empty Heaven object hev is created and put in .gitlet folder

**$ java Gitlet add dog.txt**

* If it does not already exist, a new empty Purgatory Object purg is created
* Serializing $dog.txt file and save as new file in purg

**$ java Gitlet commit “initial commit of dog.txt”**

* Create a new empty commit object com
* Unserialize everything in purg (dog.txt in this case) and add it to com (which fulfills its responsibilities)
* Peter takes com (dog.txt) and sends it to Heaven in .gitlet folder
* Peter empties purg

**$ java Gitlet add dog.txt**

* Error: nothing new to add

**$ cp manyDogs.txt dog.txt**

**$ java Gitlet add dog.txt**

* If it does not already exist, a new empty Purgatory Object purg is created
* serialize $dog.text (which is know different because of the bash command) put it in purg

**$ java Gitlet commit “replaced dog.txt with manyDogs”**

* Create a new commit object com
* Unserialize everything in purg (dog.txt in this case) and add it to com (which fulfills its responsibilities)
* Peter takes com (dog.txt) gives it a key and sends it to the right place in Heaven in .gitlet folder
* Peter clears purg

**$ java Gitlet add manyDogs.txt**

* If it does not already exist a new empty Purgatory Object purg is created
* serialize $manyDogs.text put it in purg

**$ java Gitlet add ketchupFriend.txts**

* serialize $ketchupFriend.text put it in purg (which now has two files)

**$ java Gitlet commit “added manyDogs and ketchupFriend”**

* Create a new commit object com
* Unserialize everything in purg (manyDogs and ketchupFriend in this case) and add it to com (which fulfills its responsibilities)
* Peter takes com (manyDogs and ketchupFriend) gives it a key and sends it to Heaven in .gitlet folder
* Peter empties purg

**$ java Gitlet commit YOLO**

* Error: must commit with a message

**$ cp doge.txt manyDogs.txt**

**$ java Gitlet add manyDogs.txt**

* If it does not already exist a new empty Purgatory Object purg is created
* serialize $manyDogs.text (which is know different because of the bash command) put it in purg

**$ java Gitlet commit “replaced manyDogs with doge”**

* Create a new commit object com
* Unserialize everything in purg (Manydogs.txt in this case) and add it to com (which fulfills its responsibilities)
* Peter takes com (Manydogs.txt) gives it a key and sends it to Heaven in .gitlet folder
* Peter empties purg

**$ java Gitlet add dog.txt**

* Error: Nothing to Commit

**$ java Gitlet commit “added dog yet again”**

* Error: Nothing to Commit