**CIS 285: Software Engineering Tools**

**University of Michigan – Dearborn**

**Lab 3 Git**

Assigned: Thursday 1/30

Due Date: Thursday 1/30 – Submission is due by 11:59pm on Canvas

Laboratory tasks:

**Task 1, under master branch**

1. Open GitBash over your personal project folder e.g “\Lab3\MyApp”
2. Add your contacts using [git config --global user.name “your name”] and[ git config --global user.email “your email”] and use ‘git config –list’ command to verify your contact
3. Add a local repository by using ‘git init’ and constantly use ‘git status’ to check status in this step and future steps
4. Use command ‘touch Duck.java’ to create a java file in project folder
5. Use command ‘git add –A’ to add all file in the directory to staging phase
6. Use command [git commit –m “put a reasonable message”] to commit the files and use git log to shows what happened.
7. Add a fly() method in Duck.java (check appendix 3) and use ‘git diff’ to view the difference and then repeat step e, f **(You may try command 7 in appendix 1 to combine step e and f, command 7 is very handy)**

**Task 2, under Tom branch**

1. Use command ‘Git Branch Tom’ to create a branch Tom
2. Use command ‘git checkout Tom’ to switch to this branch
3. Add a swim() method in Duck.java (check appendix 3) and use ‘git diff’ to view the difference
4. Repeat Task 1’s step e,f
5. Use command ‘git checkout master’ to switch back to master branch

**Task 3, under Bob branch**

1. Under Master branch, use ‘Git Branch Bob’ to create a branch Bob
2. Use ‘git checkout Bob’ to switch to this branch
3. Add a eat() method in Duck.java (check appendix 3) and use git diff to view the difference
4. Repeat Task 1’s step e,f
5. Use ‘git checkout master’ to switch back to master branch

**Task 4, Merge branches Tom and Bob into master**

1. Under Master branch, Use command [git merge Tom –m “Some message here”] to merge Tom branch to master
2. Use command [git merge Bob –m “Some message here”] to merge Bob branch to master
3. Check merged duck.java to clean any conflict then repeat Task 1’s step e,f to commit the final version
4. Run command gitk to show branch graph

Final report is a zip file that includes:

1. A Word document shows the screen shot of repository browser Gitk (Task 4, step d). This screen shows your branch graph
2. Your \Lab3\MyApp folder, make sure your final duck.java is in there along with .git subfolder (this subfolder is a hidden one, please choose ‘Hidden Items’ in Windows File Explorer to view it)

Appendix 1 Common Git commands

1. Git config --global user.name “Your Name”
2. Git config --global user.email “Your email address”
3. Git init //Initialize a local Git repository
4. Git status //Check status
5. Git add . or git add –A //Add all new and changed files to the staging
6. Git commit –m “message” //Commit change
7. Git commit –a –m “message” //commands 5 and 6 in combined, much more handy
8. Touch filename //create a new untracked file
9. Git diff //preview changes
10. Git difftool --tool=winmerge //use Winmerge to view changes.
11. Git branch BranchName //Create a new branch
12. Git checkout BranchName //Switch to a branch
13. Git checkout - //Switch to the branch last checked out
14. Git merge BranchName –m “message” //Merge a branch into the active branch
15. Gitk //Open git repository browser

Appendix 2 How to config Git to use Winmerge (Don’t do this in lab and this is not required in lab assignment because you don’t have authority to change gitconfig file in lab machine, try this at home)

Default git diff is not user friendly, so you can config git to use Winmerger to show differences of old file and modified one (Task 1 step g, Task 2,3 step c). Here is the procedure.

1. Download WinMergePortable.zip from Canvas and unzip it, i.e. in my machine, I unzipped to “D:\zxu\Box Sync\Teaching\Software\WinMergePortable”
2. Edit gitconfig file that locates in where git was installed. In my machine, this file is in “C:\Program Files (x86)\Git\mingw32\etc”
3. Add the following in the gitconfig file, pay attention to the single quotes wrapping the path, you must customize your path as well.

[diff]

tool = winmerge

[difftool "winmerge"]

cmd = "'D:/zxu/Box Sync/Teaching/Software/WinMergePortable/WinMergePortable.exe'" -e "$LOCAL" "$REMOTE"

1. Use command ‘git difftool --tool=winmerge’ in (Task 1 step g, Task 2,3 step c)

Appendix 3 Duck.java

Public class duck {

/\*

\* Master branch is responsible for fly() method.

\*/

public fly(){

system.out.println("A duck can fly");

}

/\*

\* Tom branch is responsible for swim() method.

\*/

public swim(){

system.out.println("A duck can swim");

/\*

\* Bob branch is responsible for eat() method.

\*/

public eat(){

system.out.println("A duck can eat");

}

}

Appendix 4 sample gitk screenshot

