accenture technology
Learning and Knowledge Management

Hands-on Workbook

Source Code Management with Git

Table of Contents

Module 2

Exercise 2.1: Install Git

Scenario

To get started hands on for Git, we need to install GIT based on OS we use.

Steps

1 **Installation Commands:**

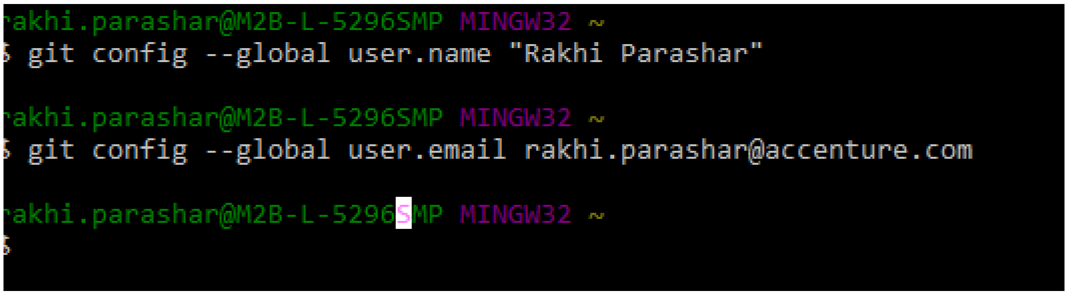
Following table lists the commands to install Git in different operating systems:

Operating System	Command to Install
Linux	Run the following command: sudo yum install git-all
Debian-based like Ubuntu	Run the following command: sudo apt-get install git-all
Windows	Access the following link: http://git-scm.com/download/win The download will start automatically. PFA doc. For git installing git instructions.

2 **Initial Configuration Setup:**

To set an identity, use the following commands:
git config --global user.name
git config --global user.email

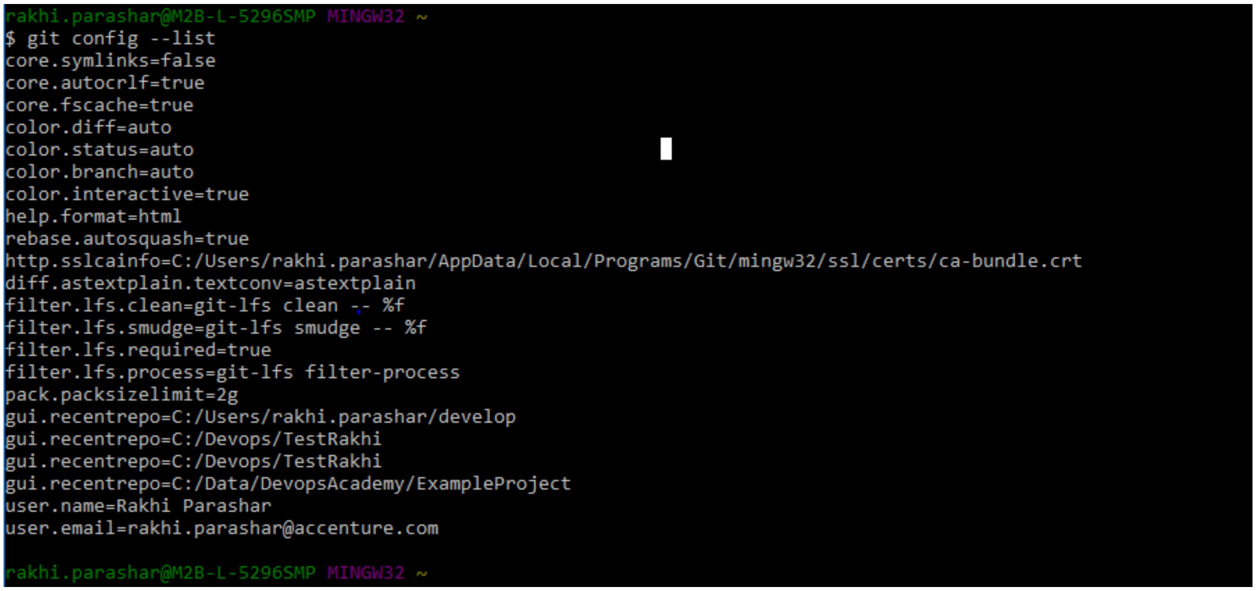
Example:
git config --global user.name “Rakhi Parashar"
git config --global user.email rakhi.parashar@accenture.com



```
rakhi.parashar@M2B-L-5296SMP MINGW32 ~
$ git config --global user.name "Rakhi Parashar"

rakhi.parashar@M2B-L-5296SMP MINGW32 ~
$ git config --global user.email rakhi.parashar@accenture.com

rakhi.parashar@M2B-L-5296SMP MINGW32 ~
$
```

3 **To view all the configurations, use the following command:**

git config –list



```
rakhi.parashar@M2B-L-5296SMP MINGW32 ~
$ git config --list
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
rebase.autosquash=true
http.sslcainfo=C:/Users/rakhi.parashar/AppData/Local/Programs/Git/mingw32/ssl/certs/ca-bundle.crt
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.required=true
filter.lfs.process=git-lfs filter-process
pack.packsizelimit=2g
gui.recentrepo=C:/Users/rakhi.parashar/develop
gui.recentrepo=C:/Devops/TestRakhi
gui.recentrepo=C:/Devops/TestRakhi
gui.recentrepo=C:/Data/DevopsAcademy/ExampleProject
user.name=Rakhi Parashar
user.email=rakhi.parashar@accenture.com

rakhi.parashar@M2B-L-5296SMP MINGW32 ~
```

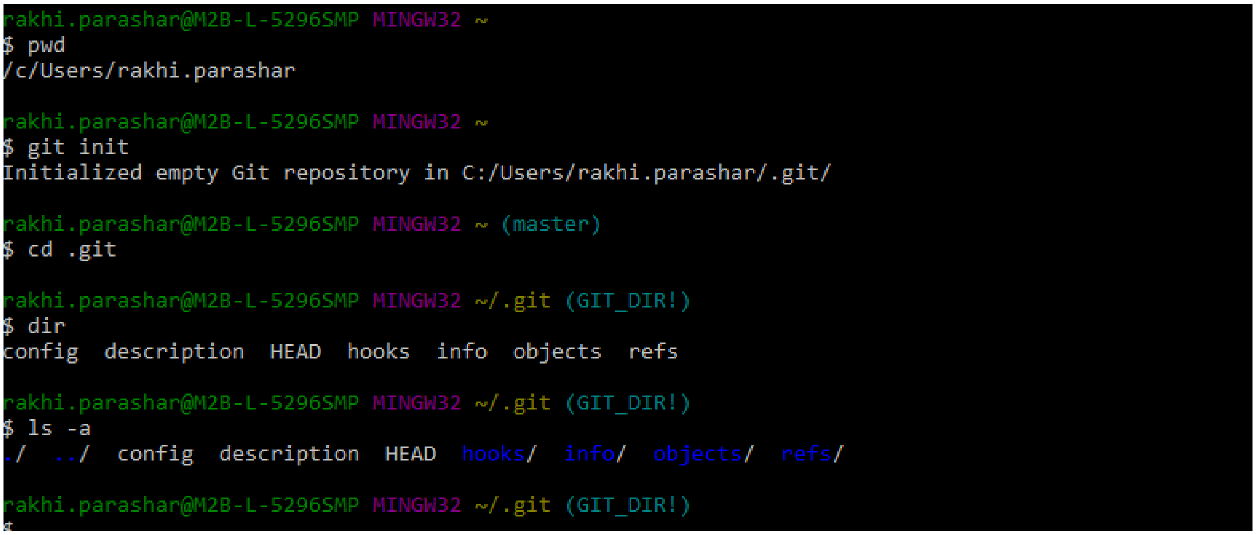
4 **Initializing Git repository:**

To initialize Git repository run the following commands:

git init (to initialize git repository)

cd .git (go to git directory and verifying GIT repository initialization)

dir (to list all files and directories .git contains)



```
rakhi.parashar@M2B-L-5296SMP MINGW32 ~
$ pwd
/c/Users/rakhi.parashar

rakhi.parashar@M2B-L-5296SMP MINGW32 ~
$ git init
Initialized empty Git repository in C:/Users/rakhi.parashar/.git/

rakhi.parashar@M2B-L-5296SMP MINGW32 ~ (master)
$ cd .git

rakhi.parashar@M2B-L-5296SMP MINGW32 ~/.git (GIT_DIR!)
$ dir
config  description  HEAD  hooks  info  objects  refs

rakhi.parashar@M2B-L-5296SMP MINGW32 ~/.git (GIT_DIR!)
$ ls -a
./  ../  config  description  HEAD  hooks/  info/  objects/  refs/

rakhi.parashar@M2B-L-5296SMP MINGW32 ~/.git (GIT_DIR!)
$
```

------- End of Exercise -------

Module 3

Exercise 3.1: Working in GitHub

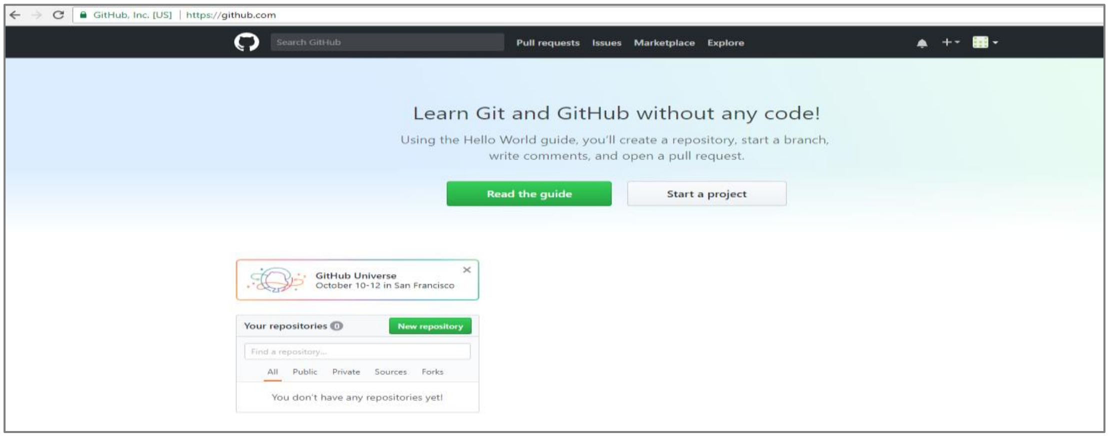
Scenario

Create an account in GitHub. Login to the GitHub and create a new repository as NewProject and branch gitbash. Create and add the file to the project. Make some changes in the code and then commit your changes in the master as well develop.

Steps

1 **Creating a GitHub Account:**

1. Login to https://github.com/join in new browser.
2. Create new account by giving username, email address and password and login.



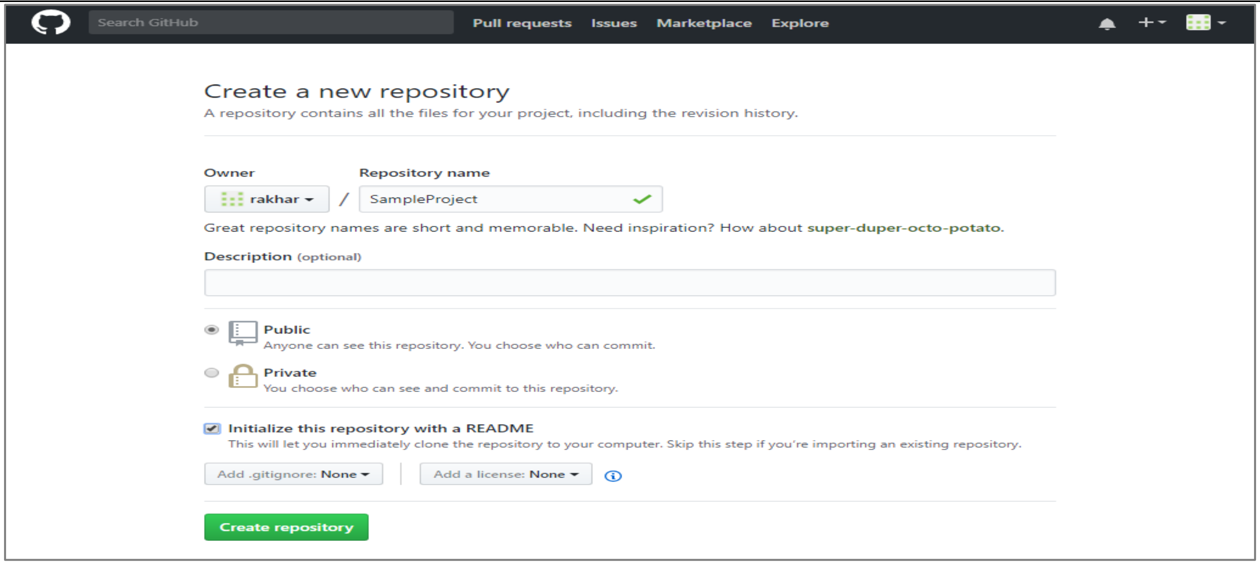
Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

You don't have any repositories yet!

2 **Creating Github Repository:**

1. Click **New repository** to create a repository.
2. Provide name of new Repository Example: **SampleProject.**
3. Select for **Public** button.
4. Check "**Initialize this repository with a README**"
5. Click **Create repository**.



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: rakhar / Repository name: SampleProject ✓

Great repository names are short and memorable. Need inspiration? How about **super-duper-octo-potato**.

Description (optional)

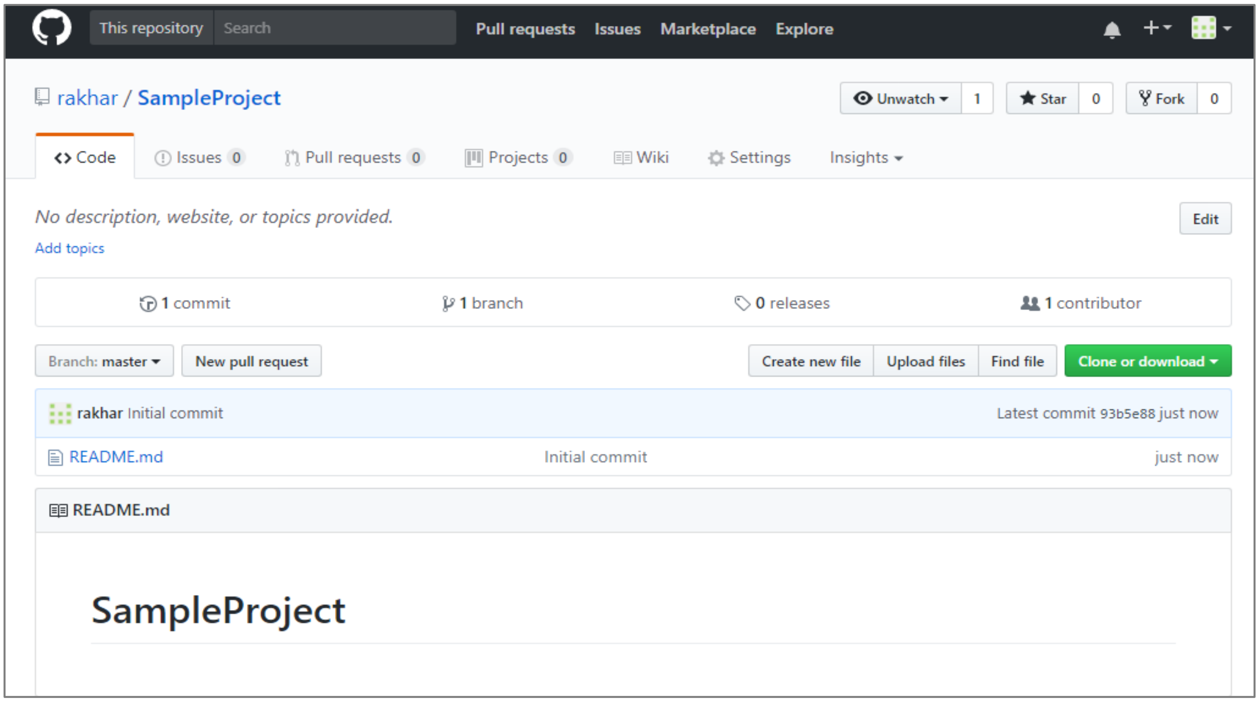
● Public
Anyone can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

☑ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None

Please review your repository created as below:



rakhar / **SampleProject**

No description, website, or topics provided.

1 commit | 1 branch | 0 releases | 1 contributor

rakhar Initial commit — Latest commit 93b5e88 just now

README.md — Initial commit — just now

README.md

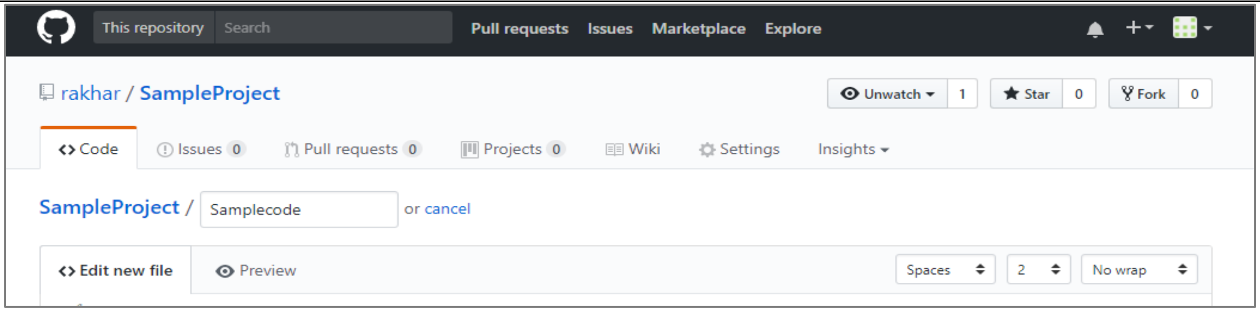
SampleProject

3 **Adding a file & editing an existing file:**

1. Click **Create new file**.
2. Name the file. For this exercise “Samplecode.sh”
3. Add lines of code.

For example:

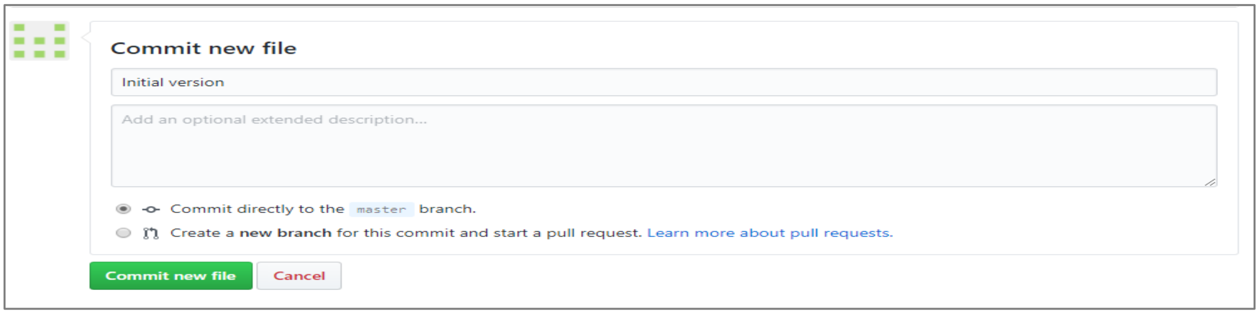
echo “welcome to sample project”



rakhar / SampleProject

SampleProject / Samplecode

4. Give comments.
5. Ensure the at you can committing in the Master branch.
6. Click **Commit new file**.

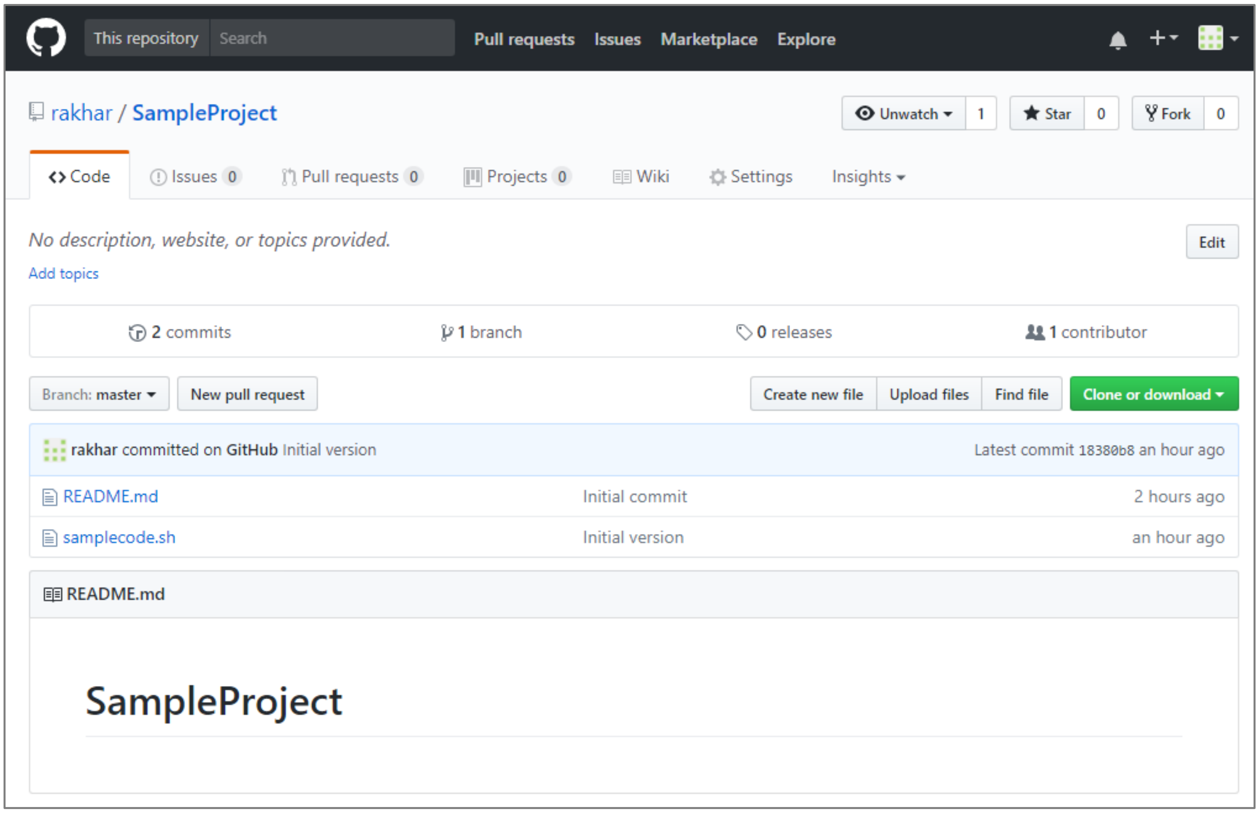

Commit new file
Add an optional extended description…
Learn more about pull requests.
Commit new file
Cancel

Initial version

● Commit directly to the master branch.

○ Create a new branch for this commit and start a pull request.

View the new files added



rakhar / SampleProject

No description, website, or topics provided.

2 commits | 1 branch | 0 releases | 1 contributor

rakhar committed on GitHub Initial version — Latest commit 18380b8 an hour ago

README.md	Initial commit	2 hours ago
samplecode.sh	Initial version	an hour ago

README.md

SampleProject

7. Click any file that had been created in the repository or click on the default **README.md** file which was created along with the repository.
8. Click the **Edit this file** icon.
9. Edit the file content in the **Edit file** tab.
10. Click **Preview changes** to view the changes. The new content will be displayed in Green.
11. Give a meaningful comment when committing which explains the changes made to the file.

Note: You can choose to commit the changes to the same branch or to a new branch.

12. Click **Commit changes** to propose the file changes.

------- End of Exercise -------

Exercise 3.2: Using Git Basic Commands

Scenario

Clone the repository that we created in GitHub into local machine. Add a file to the repository and commit the changes in the repository.

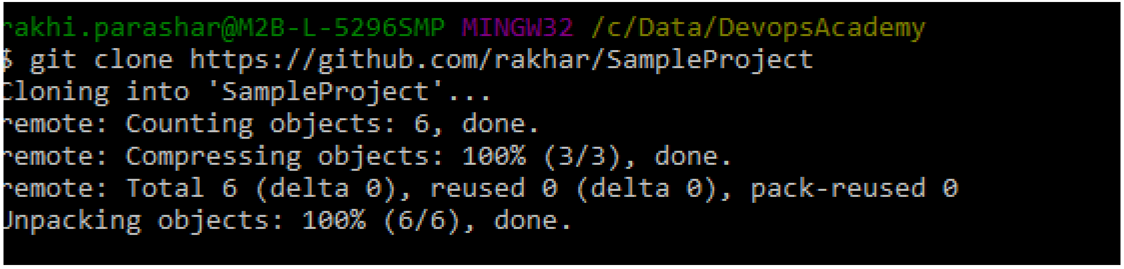
Steps

1 **Clone the repository**

Execute the following command in order to clone the repository into local machine which we created in GitHub:

$ git clone https://github.com/rakhar/SampleProject or
$ git clone https://github.com/Accenture/adop-doa-materials.git
$ git checkout -b develop

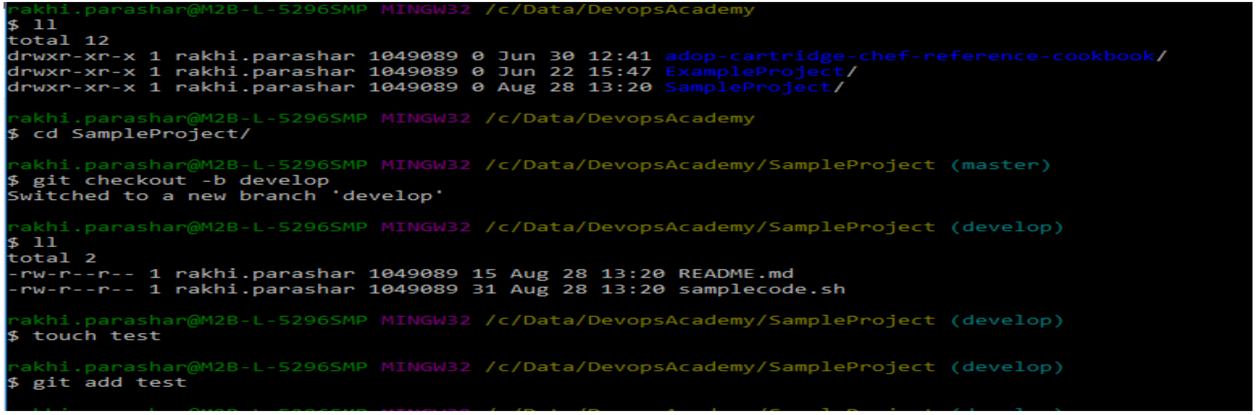
The result of using the command is as follows:



```
akhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy
 git clone https://github.com/rakhar/SampleProject
loning into 'SampleProject'...
emote: Counting objects: 6, done.
emote: Compressing objects: 100% (3/3), done.
emote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
npacking objects: 100% (6/6), done.
```

2 **Adding file to a repository**

Create a text file in GIT initialized folder and save the changes to the staging using the command, “git add <filename>” as shown in the following screen capture.



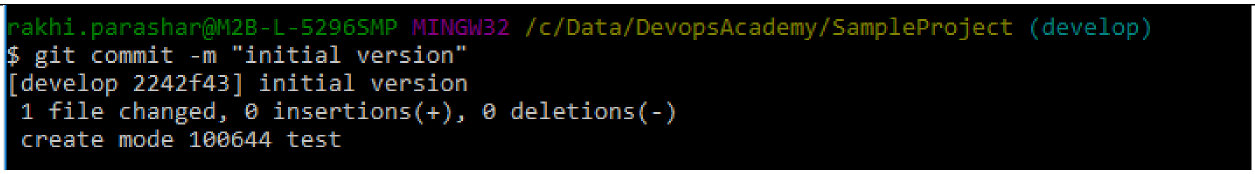
```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy
$ ll
total 12
drwxr-xr-x 1 rakhi.parashar 1049089 0 Jun 30 12:41 adop-cartridge-chef-reference-cookbook/
drwxr-xr-x 1 rakhi.parashar 1049089 0 Jun 22 15:47 ExampleProject/
drwxr-xr-x 1 rakhi.parashar 1049089 0 Aug 28 13:20 SampleProject/

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy
$ cd SampleProject/

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (master)
$ git checkout -b develop
Switched to a new branch 'develop'

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ ll
total 2
-rw-r--r-- 1 rakhi.parashar 1049089 15 Aug 28 13:20 README.md
-rw-r--r-- 1 rakhi.parashar 1049089 31 Aug 28 13:20 samplecode.sh

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ touch test

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git add test
```

3 **Committing the changes to the Repository**

To commit the changes to the GIT repository, execute the following command:

$ git commit –m “initial version”

The changes committed to the repository is shown in the following screen capture:

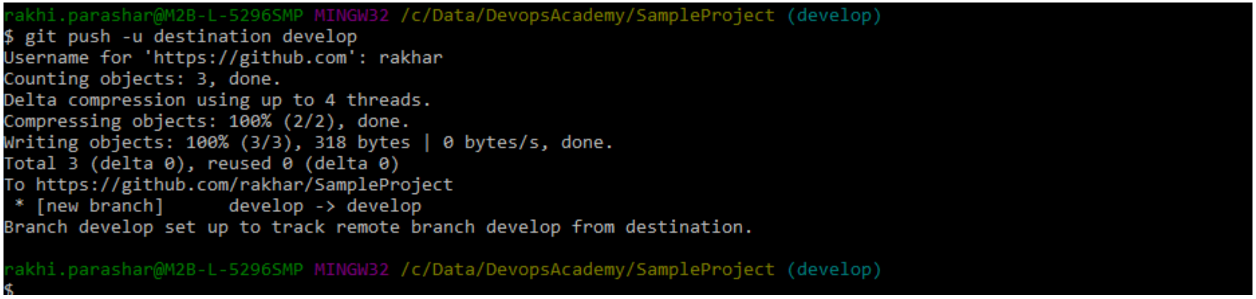


```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git commit -m "initial version"
[develop 2242f43] initial version
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 test
```

4 **Using GIT Push Command**

Run the following command to push commit on the repository:

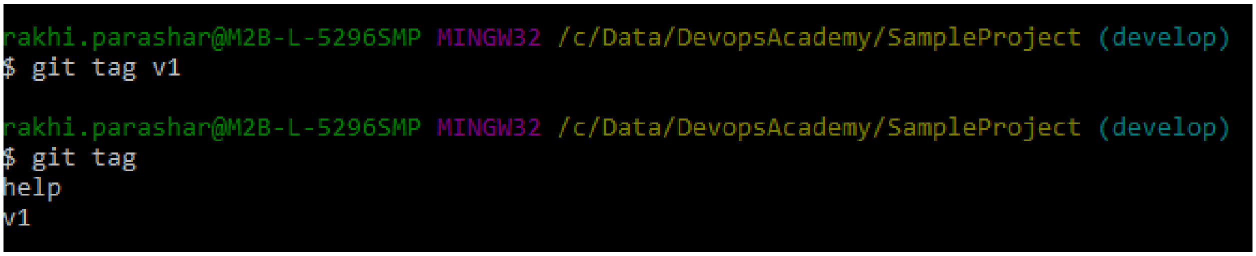
$ git push –u origin master



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git push -u destination develop
Username for 'https://github.com': rakhar
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 318 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/rakhar/SampleProject
 * [new branch]      develop -> develop
Branch develop set up to track remote branch develop from destination.

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$
```

5 **Tagging a repository**

Commands	Usage
$git tag v1	Assign a tag to the current version of the file.
$git checkout v1^	Assign a tag to the previous version, instead of mentioning the hash while checking out.
$git tag v0	Name the previous version as v0.
$git tag	View all the tags.
$git show v1	View a tagged version of a file.

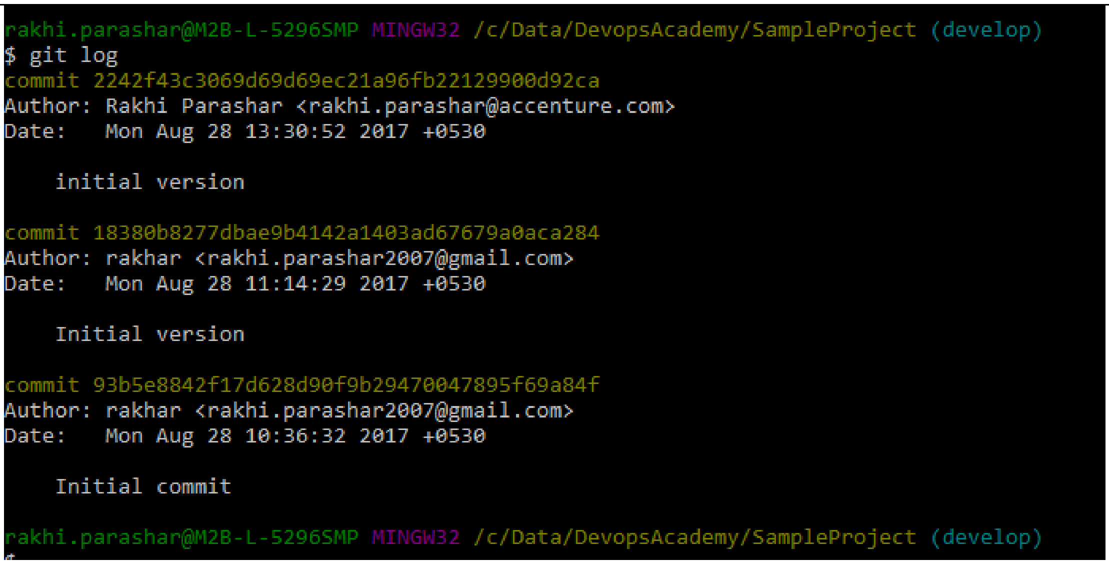


```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git tag v1

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git tag
help
v1
```

6 **Viewing the log details of the repository**

To log into the GIT repository, execute the following command and see the result:

$ git log



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git log
commit 2242f43c3069d69d69ec21a96fb22129900d92ca
Author: Rakhi Parashar <rakhi.parashar@accenture.com>
Date:   Mon Aug 28 13:30:52 2017 +0530

    initial version

commit 18380b8277dbae9b4142a1403ad67679a0aca284
Author: rakhar <rakhi.parashar2007@gmail.com>
Date:   Mon Aug 28 11:14:29 2017 +0530

    Initial version

commit 93b5e8842f17d628d90f9b29470047895f69a84f
Author: rakhar <rakhi.parashar2007@gmail.com>
Date:   Mon Aug 28 10:36:32 2017 +0530

    Initial commit

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$
```

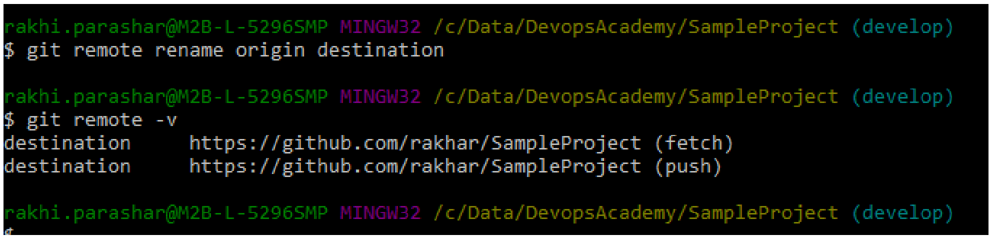
7 **Synchronizing local & remote repository**

See remote repository in GIT bash with the following command:

$ git remote –v

Use the following command to rename an existing remote repository:

$ git remote rename origin destination



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git remote rename origin destination

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git remote -v
destination     https://github.com/rakhar/SampleProject (fetch)
destination     https://github.com/rakhar/SampleProject (push)

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$
```

Note: The Origin is the default repository created when a project is created.

8 **Using pull command to update the changes to the remote repository**

To pull the changes to local repository from master, run the following command.

$ git pull

Note: If there are changes updated to the remote repository (could be by a different user), the changes can be pulled into/updated to the local repository using the GIT Pull command.

------- End of Exercise -------

Module 4

Exercise 4.1: Using Git Advance Commands

Scenario

Task 1: Create a branch devTest, pull the changes from the master branch over here and then make some more changes, save those changes and then commit your changes. Go to develop branch and then merge the changes of devTest branch in develop branch.

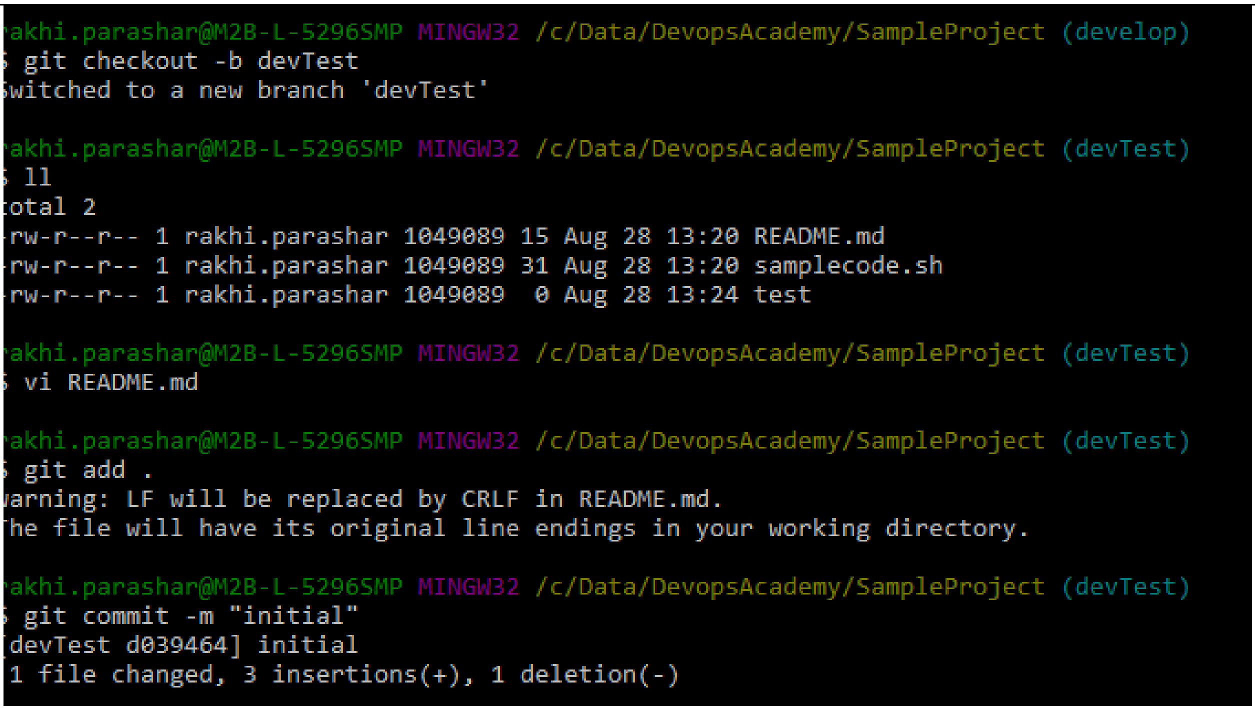
Task 2: Compare both the branches develop and devTest with master, if any file is missing from the master then fetch that file and place in both branches (develop and devTest). Use commands like grep, log, show, status, diff and bisect for analyzing history.

Task 3: Add a new file in the working tree and make the changes and commit. Post this copy this file to another file File2 and then rename the newly added file. Finally remove the File2 from the working tree. Check status of the working tree.

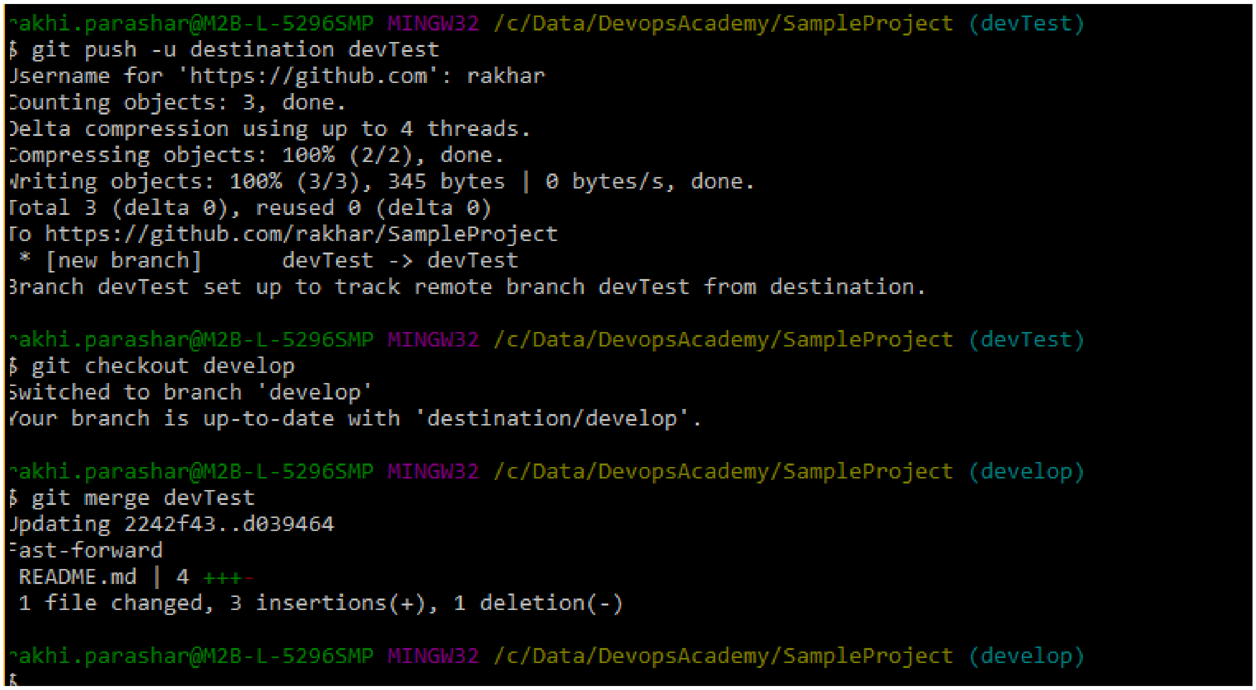
Task 4: Clone the sample-java-project.git project from git repository. Open pom.xml and remove the first few lines. OR you can create any other bug and commit the changes. Now use binary search to find the commit that introduced a bug. Hint: use Bisect and blame command.

Steps

1	**Scenario Task** • Create a branch devTest, pull the changes from the master branch over here and then make some more changes, save those changes and then commit your changes. • Go to develop branch and then merge the changes of devTest branch in develop branch. Following steps and screen captures should help you to accomplish the considered task: 1. Run the command in gitbash "**git checkout -b devTest**" for creating new branch. 2. Use command "**git pull**" to get the latest code from master in this new devTest branch. 3. Make changes to any of the file for example: README.md file. 4. Save and commit the changes. 5. Checkout develop branch using command "**git checkout develop**". 6. Merge the changes of devTest branch to develop branch using command "**git merge devTest**".

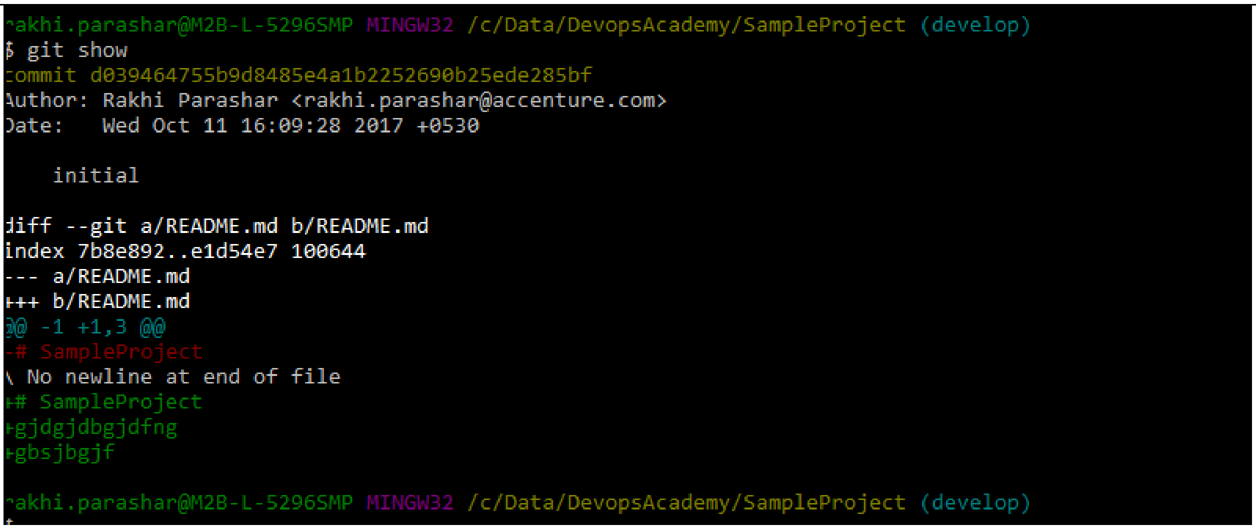


```
akhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
 git checkout -b devTest
witched to a new branch 'devTest'

akhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (devTest)
 ll
otal 2
rw-r--r-- 1 rakhi.parashar 1049089 15 Aug 28 13:20 README.md
rw-r--r-- 1 rakhi.parashar 1049089 31 Aug 28 13:20 samplecode.sh
rw-r--r-- 1 rakhi.parashar 1049089  0 Aug 28 13:24 test

akhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (devTest)
 vi README.md

akhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (devTest)
 git add .
arning: LF will be replaced by CRLF in README.md.
he file will have its original line endings in your working directory.

akhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (devTest)
 git commit -m "initial"
devTest d039464] initial
1 file changed, 3 insertions(+), 1 deletion(-)
```



```
akhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (devTest)
$ git push -u destination devTest
Jsername for 'https://github.com': rakhar
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 345 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/rakhar/SampleProject
 * [new branch]      devTest -> devTest
Branch devTest set up to track remote branch devTest from destination.

akhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (devTest)
$ git checkout develop
Switched to branch 'develop'
Your branch is up-to-date with 'destination/develop'.

akhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git merge devTest
Jpdating 2242f43..d039464
Fast-forward
 README.md | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)

akhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$
```

2	**Scenario Task** Compare both the branches develop and devTest with master, if any file is missing from the master then fetch that file and place in both branches (develop and devTest). Use commands like grep, log, show, status, diff etc.



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git show
commit d039464755b9d8485e4a1b2252690b25ede285bf
Author: Rakhi Parashar <rakhi.parashar@accenture.com>
Date:   Wed Oct 11 16:09:28 2017 +0530

    initial

diff --git a/README.md b/README.md
index 7b8e892..e1d54e7 100644
--- a/README.md
+++ b/README.md
@@ -1 +1,3 @@
-# SampleProject
\ No newline at end of file
+# SampleProject
+gjdgjdbgjdfng
+gbsjbgjf

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$
```

3

Scenario Task

- Add a new file in the working tree and make the changes and commit.
- Post this copy this file to another file File2 and then rename the newly added file.
- Finally remove the File2 from the working tree. Check status of the working tree.
- Also now if you don’t want to commit these changes just stash/save your changes and also give them tags.
- Use log for logging.

Following steps should help you to accomplish the considered task:

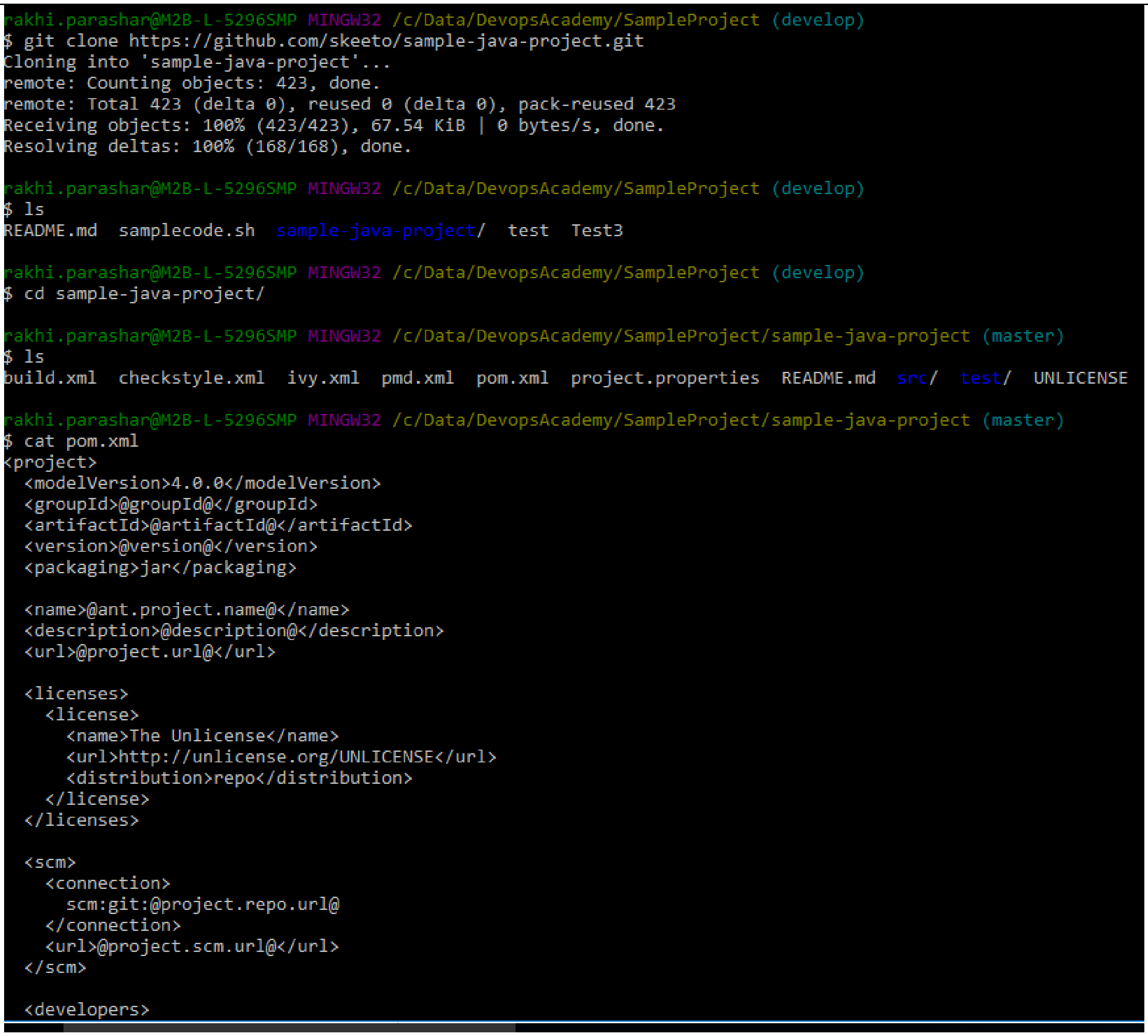
1. To create new file use command “**touch Test2**”
2. Make changes to the file and save it.
3. Run the following command:
   git add Test2
   git commit -m “initial” and “git push”
4. Use command “**cp Test2 Test3**” to copy files.
5. Rename Test3 as NewTest by executing the following command:
   mv Test3 NewTest
6. Remove Test2 “**rm -f Test2**”

4

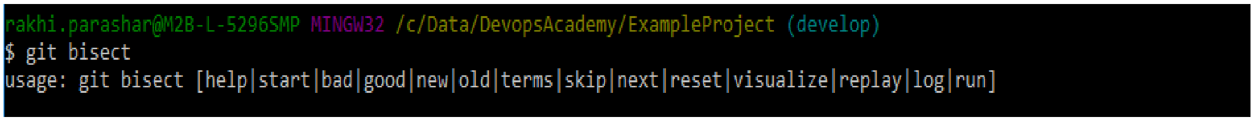
Scenario Task

Following steps and screen captures should help you to accomplish the considered task:

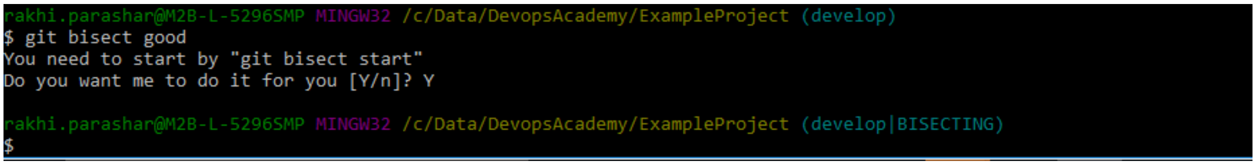
1. Git clone https://github.com/skeeto/sample-java-project.git
2. Open **pom.xml** and remove few lines.
3. Commit the changes.
4. Now use Bisect and Blame command to find which commit has introduced the bug and who has done.



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git clone https://github.com/skeeto/sample-java-project.git
Cloning into 'sample-java-project'...
remote: Counting objects: 423, done.
remote: Total 423 (delta 0), reused 0 (delta 0), pack-reused 423
Receiving objects: 100% (423/423), 67.54 KiB | 0 bytes/s, done.
Resolving deltas: 100% (168/168), done.

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ ls
README.md  samplecode.sh  sample-java-project/  test  Test3

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ cd sample-java-project/

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject/sample-java-project (master)
$ ls
build.xml  checkstyle.xml  ivy.xml  pmd.xml  pom.xml  project.properties  README.md  src/  test/  UNLICENSE

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject/sample-java-project (master)
$ cat pom.xml
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>@groupId@</groupId>
  <artifactId>@artifactId@</artifactId>
  <version>@version@</version>
  <packaging>jar</packaging>

  <name>@ant.project.name@</name>
  <description>@description@</description>
  <url>@project.url@</url>

  <licenses>
    <license>
      <name>The Unlicense</name>
      <url>http://unlicense.org/UNLICENSE</url>
      <distribution>repo</distribution>
    </license>
  </licenses>

  <scm>
    <connection>
      scm:git:@project.repo.url@
    </connection>
    <url>@project.scm.url@</url>
  </scm>

  <developers>
```



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject (develop)
$ git bisect
usage: git bisect [help|start|bad|good|new|old|terms|skip|next|reset|visualize|replay|log|run]
```



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject (develop)
$ git bisect good
You need to start by "git bisect start"
Do you want me to do it for you [Y/n]? Y

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject (develop|BISECTING)
$
```

------- End of Exercise -------

Exercise 4.2: Using Git Advance Commands

Scenario

Following are the key scenarios where you need to use some of the other Git Advance Commands:

- Creating an alias for most frequently used command based on the history of commit.
- Identifying the file status by displaying which stage is a file in Git
- Merging without using merge command
- Keeping the changes in local branch that you want to commit later.
- Committing only specific commits from master to your current
- Removing untracked files and directories

Steps

1 **Scenario Task - Creating an alias for most frequently used command based on the history of commit.**

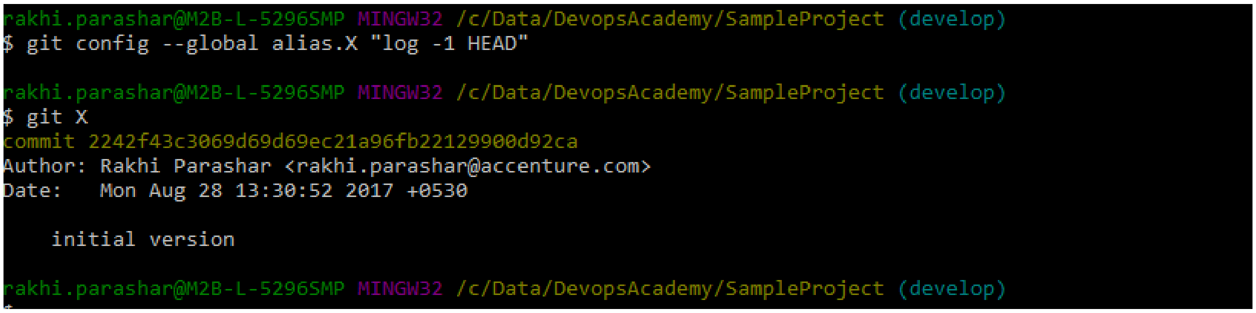
Alan wants to create an alias for most frequently used command based on the history of commit. He creates an alias by using the following command to save his writing time.

Command: git config –global alias.XZ

- Example: git config --global alias.X "log -1 HEAD"

After creating alias, the command can be as follows.

- git X instead of git log -1 HEAD.

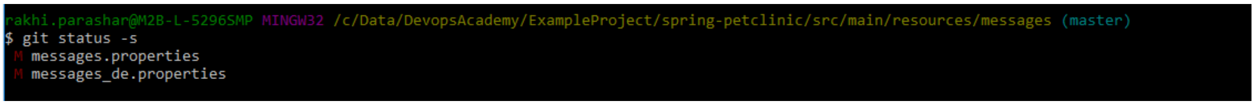


```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git config --global alias.X "log -1 HEAD"

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
$ git X
commit 2242f43c3069d69d69ec21a96fb22129900d92ca
Author: Rakhi Parashar <rakhi.parashar@accenture.com>
Date:   Mon Aug 28 13:30:52 2017 +0530

    initial version

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/SampleProject (develop)
```

2 **Scenario Task - Identifying the file status by displaying which stage is a file in Git**

Alan has his local files with similar names. He has forgotten what was the last action taken on those files. He wants to display which stage is a file in GIT.

Command: git status –s



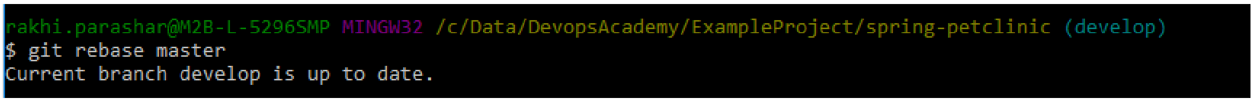
```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject/spring-petclinic/src/main/resources/messages (master)
$ git status -s
 M messages.properties
 M messages_de.properties
```

"**M**" indicates file is modified and the file needs to staged for commit.

3 **Scenario Task - Merging without using merge command**

Alan wants to merge the repository without using merge command.

Pull upstream changes with rebase instead of merge.



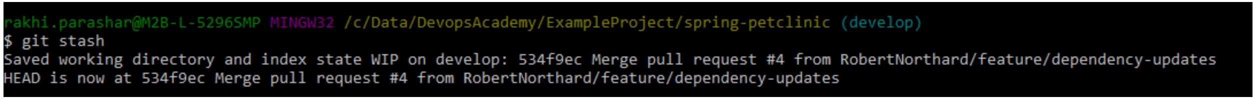
```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject/spring-petclinic (develop)
$ git rebase master
Current branch develop is up to date.
```

4

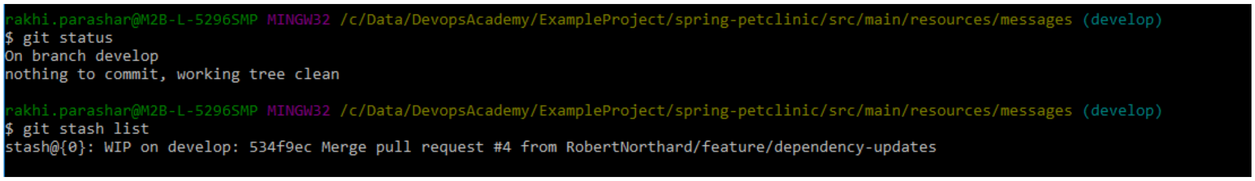
Scenario Task - Keeping the changes in local branch that you want to commit later

Alan wants to keep the changes in local branch and do not want to commit right now, and may wish to commit later.

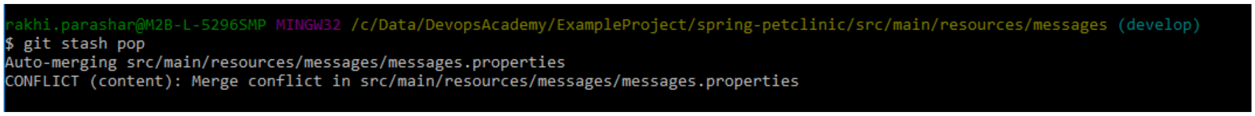
Stash : Temporarily save/stash the changes in current branch
Command : git stash
git status
git list
git stash pop



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject/spring-petclinic (develop)
$ git stash
Saved working directory and index state WIP on develop: 534f9ec Merge pull request #4 from RobertNorthard/feature/dependency-updates
HEAD is now at 534f9ec Merge pull request #4 from RobertNorthard/feature/dependency-updates
```



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject/spring-petclinic/src/main/resources/messages (develop)
$ git status
On branch develop
nothing to commit, working tree clean

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject/spring-petclinic/src/main/resources/messages (develop)
$ git stash list
stash@{0}: WIP on develop: 534f9ec Merge pull request #4 from RobertNorthard/feature/dependency-updates
```



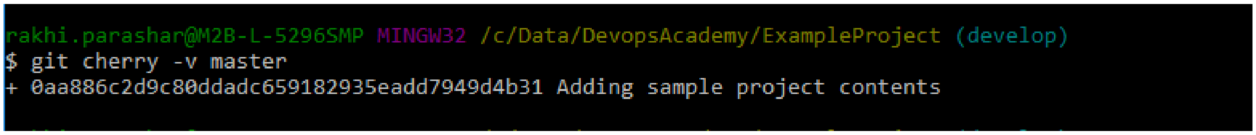
```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject/spring-petclinic/src/main/resources/messages (develop)
$ git stash pop
Auto-merging src/main/resources/messages/messages.properties
CONFLICT (content): Merge conflict in src/main/resources/messages/messages.properties
```

5

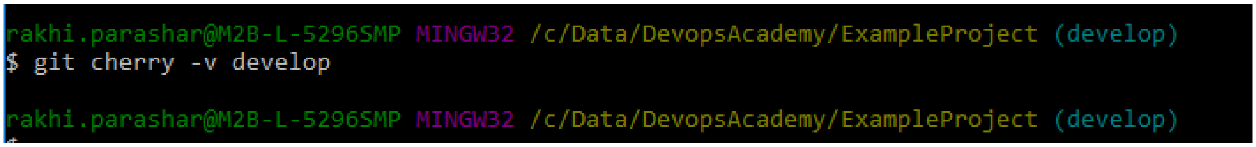
Scenario Task - Committing only specific commits from master to your current

Alan wants commit only specific commits from master to the current branch.

- Merge a cherry-picked remote commit with your branch
- Command: git cherry-pick <commitSHA>
- Git cherry –v SampleBranchName
- Git cherry –v master



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject (develop)
$ git cherry -v master
+ 0aa886c2d9c80ddadc659182935eadd7949d4b31 Adding sample project contents
```



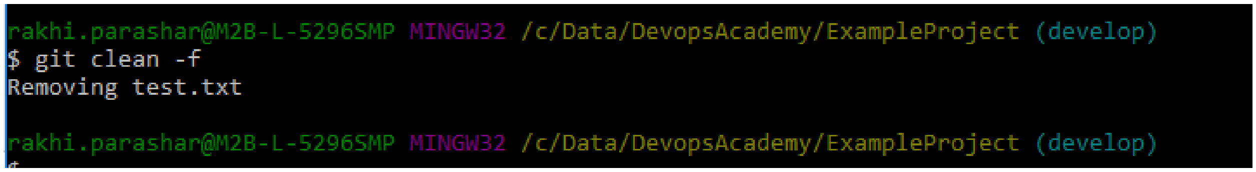
```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject (develop)
$ git cherry -v develop

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject (develop)
```

6

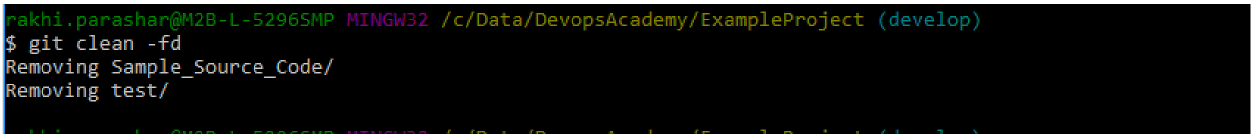
Scenario Task - Removing untracked files and directories

Alan wants to remove untracked files and directories.

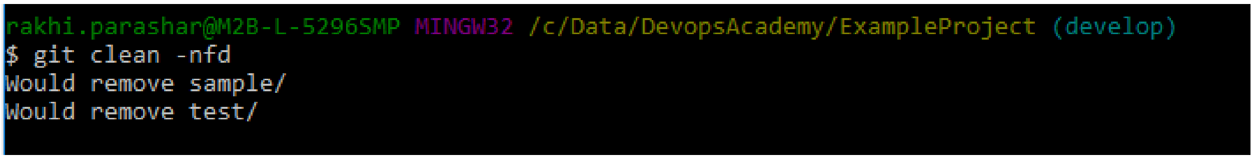
Git clean –f (remove untracked files)
Git clean –fd (remove untracked files/directories)
Git clean –nfd (list all which will be removed)



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject (develop)
$ git clean -f
Removing test.txt

rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject (develop)
```



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject (develop)
$ git clean -fd
Removing Sample_Source_Code/
Removing test/
```



```
rakhi.parashar@M2B-L-5296SMP MINGW32 /c/Data/DevopsAcademy/ExampleProject (develop)
$ git clean -nfd
Would remove sample/
Would remove test/
```

------- End of Exercise -------