1. What is a repository?

   Source Code (based) Repository
   Version Controlling

2. What are the different architectures of repositories?

   1. Centralized
   2. Decentralized

3. Which architecture Git Follows / how is that a better fit compared to the other options

   1. Flexibility (Cost, Work)
   2. Transactional Workflow

4. Actually doing git hub

5. Doing Git

   - Create File
   - Add the file to the staging area
   - Commit the file
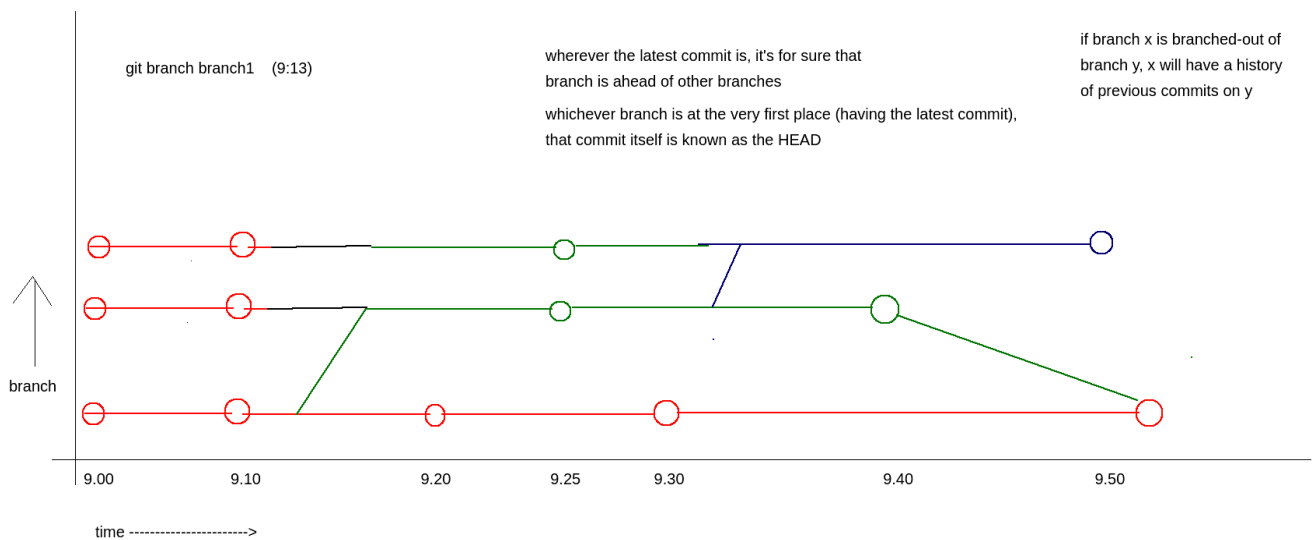


```
 1  sudo yum update
 2  clear
 3  sudo yum install git -y        ----> Git installation
 4  clear
 5  pwd
 6  ls
 7  mkdir mylocalrepo              ----> Creating a local repo folder
 8  ls
 9  cd mylocalrepo/
10  pwd
11  ls -la
12  clear
13  git init                       ----> Initializing my local Git Repository
14  ls -la
15  ls
16  cd .git
17  ls -la
18  cd ..
19  clear
20  vi file1.txt                   ----> Creating a new file
```

21 ls
22 git status   ----> Checking the current git status
23 git add file1.txt   ----> Adding the new file to staging area
24 git status
25 git commit   ----> Committing the changes (creating a version)
26 cat file1.txt
27 cat > file1.txt   ----> Modifying file1
28 cat file1.txt
29 git status
30 git commit -m "hi"   ----> Trying to commit without add (FAILURE!!)
31 git add file1.txt
32 git commit -m "Modified file 1"
33 git log   ----> Checking the commit history

Class 2:

git branch branch1   (9:13)

wherever the latest commit is, it's for sure that
branch is ahead of other branches

if branch x is branched-out of
branch y, x will have a history
of previous commits on y

whichever branch is at the very first place (having the latest commit),
that commit itself is known as the HEAD

branch

9.00   9.10   9.20   9.25   9.30   9.40   9.50

time ---------------------->

1 sudo yum update -y
2 sudo yum install git -y
3 clear
4 git -v

```
 5  git --v
 6  git --version
 7  clear
 8  mkdir localRepo
 9  cd localRepo/
10  git init
11  ls -la
12  clea
13  clear
14  git status
15  git log
16  clear
17  cat >> file1.txt
18  cat file1.txt
19  git add file1.txt
20  git status
21  git commit -m "initial commit, file1 added"
22  git remote add origin https://github.com/sonasonaram/8am_repo.git
23  git push
24  git branch
25  git push -u origin master
26  clear
27  cat >> file2.txt
28  cat >> file3.txt
29  git status
30  vi file1.txt
31  git status
32  clear
33  pwd
34  git add .
35  git status
36  git commit -m "file1 mod, file2_3 added"
37  git push
38  git log
39  clear
40  git branch
41  git branch branch1
42  git branch
43  git status
44  git log
45  clear
46  vi file3.txt
47  git commit -a file3.txt -m "editing file3"
48  git commit -a . -m "editing file3"
49  git commit --help
50  git commit -m "editing file3" -a .
```

```
51  git commit -a -m "editing file3" .
52  git commit . -m "sdf"
53  git log
54  git checkout branch1
55  git branch
56  git checkout master
57  git checkout branch1
58  git log
59  cat > file_b1
60  git add .
61  git commit -m "file_b1 added"
62  git log
63  git checkout master
64  git log
65  git push
66  ls
67  git checkout branch1
68  ls
69  cat file3.txt
70  git checkout master
71  cat file3.txt
72  git checkout branch1
73  git push
74  git push --set-upstream origin feature1
75  git push --set-upstream origin branch1
```

Class 3:

```
 1  sudo yum update -y
 2  sudo yum install git -y
 3  clear
 4  mkdir localrepo
 5  cd localrepo/
 6  git init
 7  cat README.md
 8  cat > README.md
 9  git add .
10  git commit -m "initial commit"
11  git log
12  <NOT THERE BCZ VISHAL DID A MISTAKE>
13  git branch
14  git branch feature1
15  git checkout feature1
16  git checkout master
```

```
17  clear
18  ls
19  cat > code1.c
20  git branch feature2
21  git add .
22  git commit code1.c -m "first code added"
23  git branch feature3
24  ls
25  git checkout feature1
26  ls
27  cat >> code2.c
28  git add .
29  git commit - "code 2 development completed"
30  git commit -m "code 2 development completed"
31  ls
32  git checkout master
33  git merge feature1
34  ls
35  git log
36  git checkout feature2
37  ls
38  cat >> code3.c
39  git add .
40  git commit -m "code3 added"
41  git checkout master
42  ls
43  cat >> code4.c
44  git add .
45  git commit -m "code 4 developed"
46  ls
47  git log
48  git merge feature2
49  ls
50  git log
51  cat >> code4.c
52  cat code4.c
53  git add .
54  git commit -m "change in code 4"
55  cat >> code4.c
56  git add .
57  git commit -m "change in code 4 again"
58  ls
59  git branch
60  git checkout -b branch1
61  ls
62  vi code4.c
```

```
 63  git add .
 64  git commit -m "editing code4 on b1"
 65  git checkout master
 66  git checkout -b branch2
 67  ls
 68  vi code4.c
 69  git add .
 70  git commit -m "editing code4 on b2"
 71  git checkout master
 72  clear
 73  git merge branch1
 74  git log
 75  git branch branch1
 76  git checkout branch1
 77  git log
 78  git checkout branch2
 79  git log
 80  clear
 81  git checkout master
 82  git merge branch2
 83  cat code4.c
 84  git checkout branch2
 85  vi code4.c
 86  git status
 87  git add .
 88  git commit -m "solved a conflict on code 4, kept both"
 89  git log
 90  cat code4.c
 91  git checkout -b branch3
 92  cat code4.c
 93  cd ..
 94  rm *\
 95  rm *
 96  rm -rf *
 97  clear
 98  mkdir localrepo
 99  cd localrepo/
100  git init
101  clear
102  cat > file1
103  git add .
104  git commit -m "file1 on master"
105  cat > file2
106  git add .
107  git commit -m "file2 on master"
108  clear
```

```
109  git checkout -b branch1
110  ls
111  cat > file3
112  git add .
113  git commit -m "file 3 on branch 1"
114  cat > file4
115  git add .
116  git commit -m 'file 4 on branch 1"
117  git commit -m "file 4 on branch 1"
118  clear
119  git checkout master
120  cat > file5
121  git add .
122  git commit -m "file 5 on master"
123  ls
124  clear
125  ls
126  git rebase branch1
127  git log
128  clear
129  ls
130  git checkout branch1
131  ls
132  git checkout -b branch1_1
133  ls
134  cat >> file1_1
135  cat >> file1_2
136  git add .
137  git status
138  git restore --staged file1_1
139  git status
140  git commit -m "file1_2"
141  git status
142  git add .
143  git commit -m "file1_1"
144  git log
145  git checkout branch1
146  cat >> file_b1
147  git add .
148  git commit -m "file_b1"
149  git checkout branch1_1
150  cat > file1_3
151  git add .
152  git commit -m "file1_3"
153  git checkout branch1
154  git rebase branch1_1
```

```
155  git log
156  clear
157  cd ..
158  rm -rf *
159  mkdir localRepo
160  cd localRepo/
161  git init
162  git pull https://github.com/sonasonaram/repo_morn_8.git
163  ls
164  git log
```