

Eclipse, Git and Github

A very short introduction

More info in the notes



Your TA this semester

Shabbir Hussain
Graduated EE @ McGill Last Semester
1st yr Masters in Comp Eng
TAed this course twice Before

Contact:

Shabbir.Hussain@mail.mcgill.ca

<https://github.com/shabbir-hussain/>



What are we doing today?

0. Programming Exercise

1. Installing Eclipse

2. Installing Git

3. Configuring Git

4. Short break

5. Basics of Git

Bring your **Laptops** to tutorials!

Eclipse IDE



GETTING STARTED MEMBERS PROJECTS MORE ▾

HOME / DOWNLOADS

» Packages | Developer Builds | Java™ 8 Support

Security release: A vulnerability with **JGit client (CVE-2014-9390)** was found, please **check for updates**.

Eclipse Luna SR1a (4.4.1) Release for



Eclipse IDE for Java Developers, 154 MB

Downloaded 1 Times

The essential tools for any Java developer, including a Java IDE, a CVS client, Git client, XML Editor, Mylyn, Maven integration...

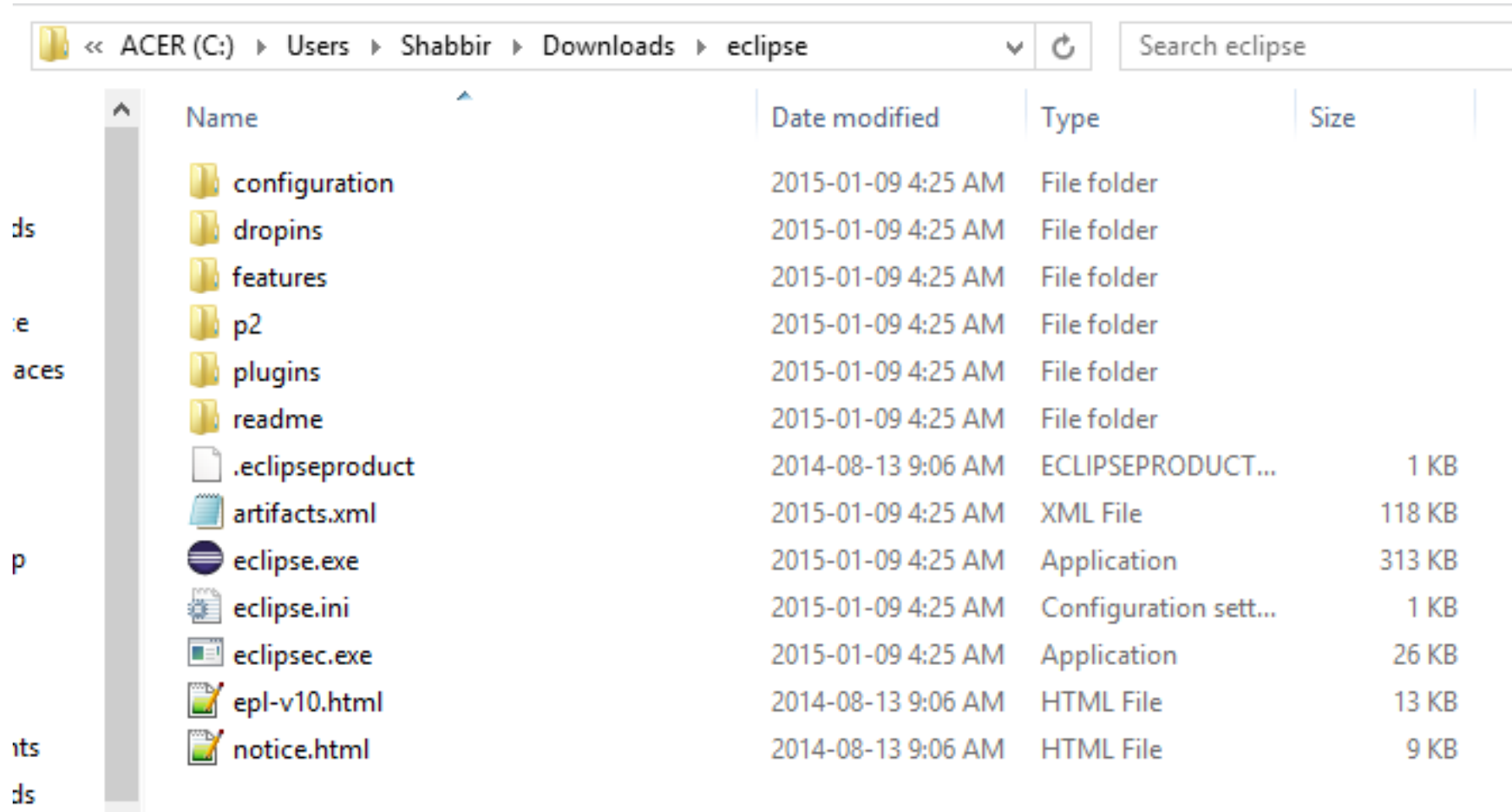


Windows 32 Bit
Windows 64 Bit

• <https://eclipse.org/downloads/>

Eclipse IDE

- **Extract** zip
- **Start** using



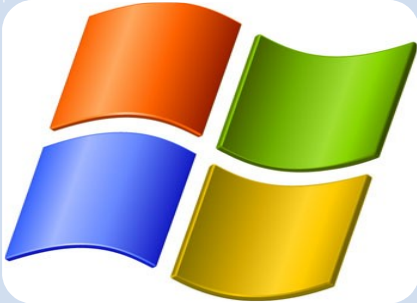
<< ACER (C:) > Users > Shabbir > Downloads > eclipse					Search eclipse	
	Name	Date modified	Type	Size		
	configuration	2015-01-09 4:25 AM	File folder			
	dropins	2015-01-09 4:25 AM	File folder			
	features	2015-01-09 4:25 AM	File folder			
	p2	2015-01-09 4:25 AM	File folder			
	plugins	2015-01-09 4:25 AM	File folder			
	readme	2015-01-09 4:25 AM	File folder			
	.eclipseproduct	2014-08-13 9:06 AM	ECLIPSEPRODUCT...	1 KB		
	artifacts.xml	2015-01-09 4:25 AM	XML File	118 KB		
	eclipse.exe	2015-01-09 4:25 AM	Application	313 KB		
	eclipse.ini	2015-01-09 4:25 AM	Configuration sett...	1 KB		
	eclipsesec.exe	2015-01-09 4:25 AM	Application	26 KB		
	epl-v10.html	2014-08-13 9:06 AM	HTML File	13 KB		
	notice.html	2014-08-13 9:06 AM	HTML File	9 KB		

Eclipse IDE

- IDE: Integrated Development Environment
- **Write** code, **compile**, **test** all in one program
- <https://eclipse.org/downloads/>



Let get Git



<http://git-scm.com/download/win>



<http://git-scm.com/download/mac>



```
sudo apt-get  
install git
```

Registering for Github

1

GitHub, Inc. [US] https://github.com

2

Pick a username

Your email

Create a password

Use at least one lowercase letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We will send you account related emails occasionally.

3

Create your personal account

Username

Hitmonlee ✓

Email Address

hitmonleetheoneandonly@gmail.com ✓

You will occasionally receive account related emails. We promise not to share your email with anyone.

Password

..... ✓

Confirm your password

.....

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

Create an account

What is Version Control?

- Version Control software is a tool used to **keep track** of different **versions** of files
- **Revert** to an old version
- **Branch** to create multiple versions
- **Merge** two different versions together
- **Synchronize** files on different **machines**



Pop Quiz

- Would it be a good idea to use **version control** on your **photo album**?
- When would you want to **revert** to a **previous version** of a file?
- Why would you want to **branch** your files?

Why You're going to use it

- You **need** it for this class
- It's a great way to **sync** your code with your **team**
- Because using **dropbox** to sync your code is **so last semester**

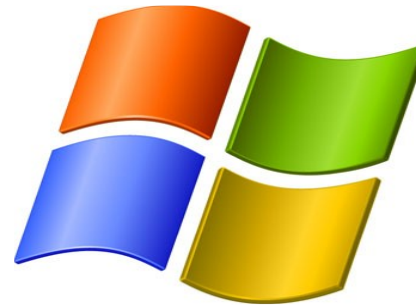
Did I install it correctly



Terminal



Git Bash



Making git work with GitHub

1

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)  
$ git config --global user.name "shabbir-hussain"
```

2

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)  
$ git config --global user.email shabbir.hussain@outlook.com
```

3

<https://help.github.com/articles/generating-ssh-keys>

Getting the Tutorial notes

1

 GitHub, Inc. [US] <https://github.com/shabbir-hussain/ecse321tutF15>

2

HTTPS clone URL

<https://github.com/shabbir-hussain/ecse321tutF15>



You can clone with [HTTPS](#) or [Subversion](#). ?



Clone in Desktop



Download ZIP

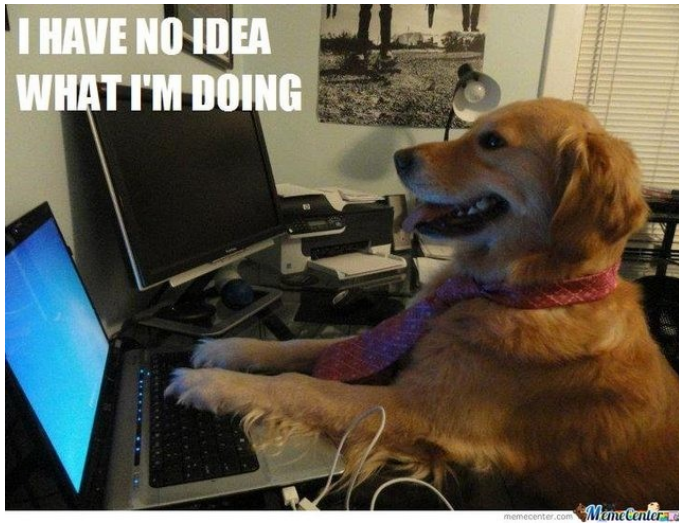
3

```
Shabbir@SHABBIR-PC ~/Documents/code
$ git clone git@github.com:shabbir-hussain/ecse321tutF15.git
```

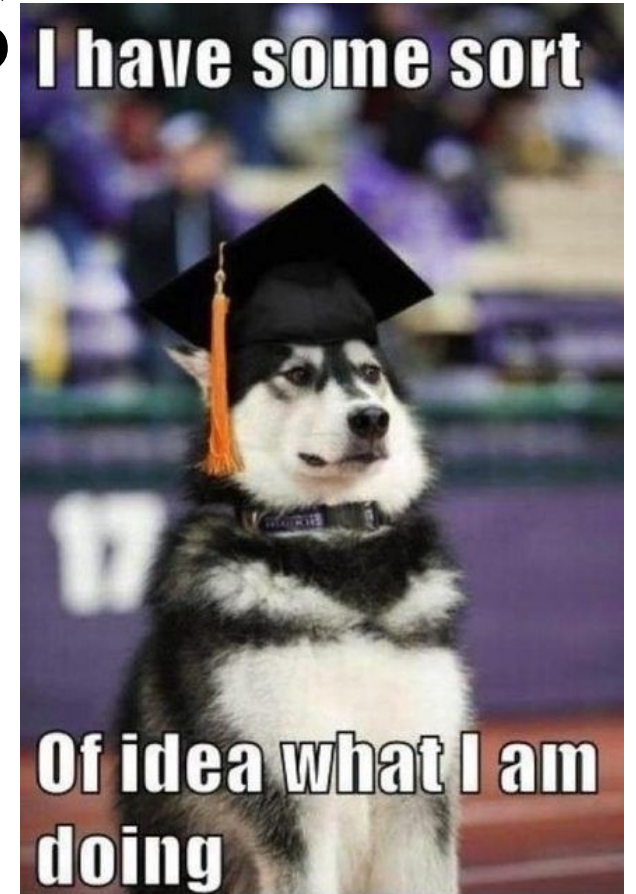
Getting the Tutorial notes

```
Git Bash
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university
$ git clone git@github.com:mcgill-ecse321/class-notes.git
Cloning into 'class-notes'...
remote: Counting objects: 290, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 290 (delta 0), reused 0 (delta 0)Receiving objects: 96% (279/290), 5.68 MiB | 314 KiB/s
Receiving objects: 100% (290/290), 5.91 MiB | 313 KiB/s, done.
Resolving deltas: 100% (59/59), done.
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university
$
```

Understanding Git and some basic functions



Going from this

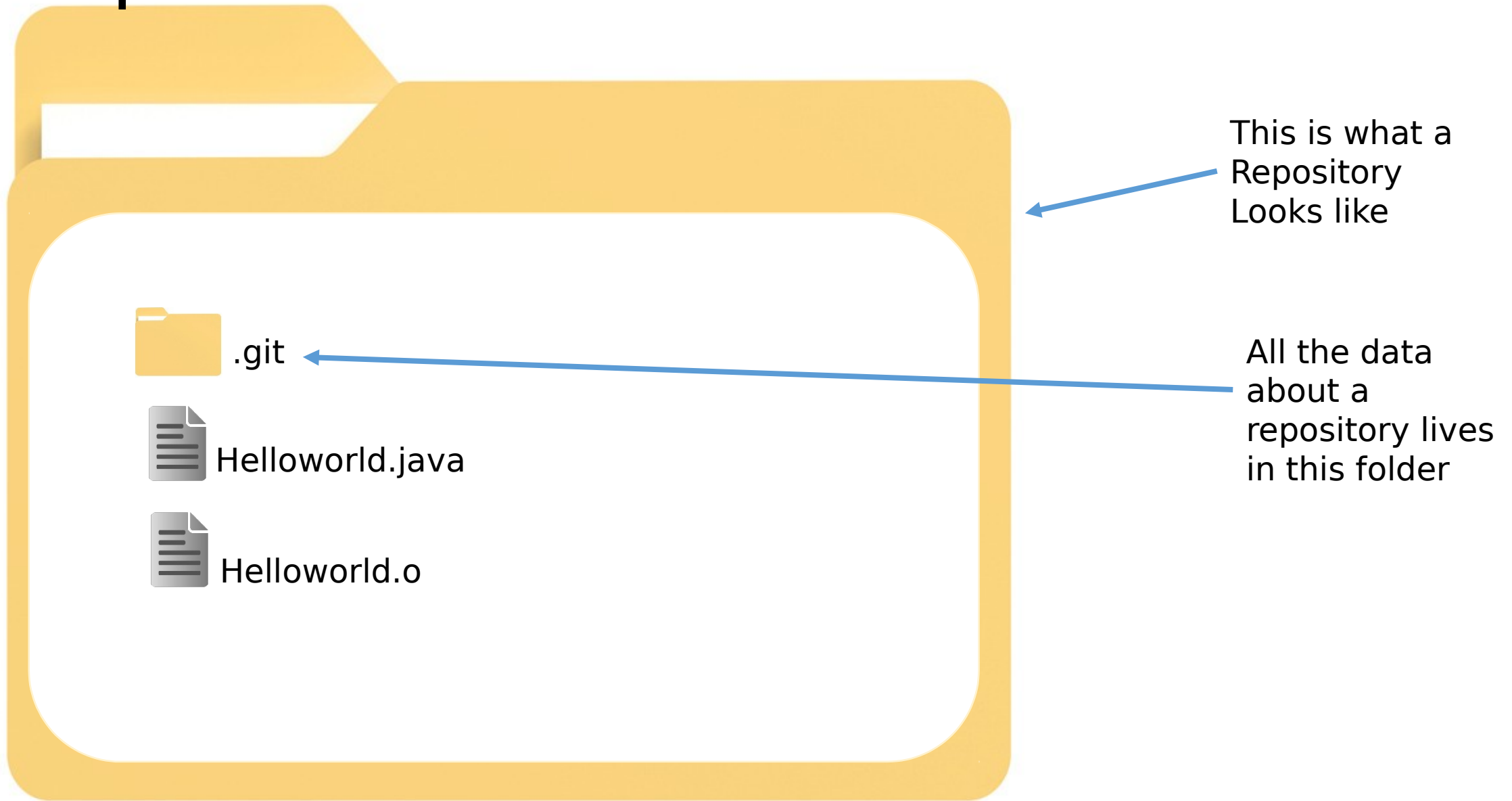


To this

Quick glossary

- **Git** is your **version control software**
- **Git Hub** is a **website** that hosts your repositories
- **Repositories** are a collection of **files** and file **histories**
- **Commits** are like **save states**

Repositories



Creating our own Repository

1

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university  
$ mkdir myfirstrepo
```

2

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university  
$ cd myfirstrepo/
```

3

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo  
$ git init  
Initialized empty Git repository in c:/Users/Shabbir/Documents/code/university/myfirstrepo/.git/  
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
```

What is a Commit?

```
commit 2a0735092cea1b7f7c850a48b86e8847bf979236
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date: Thu Aug 28 15:33:09 2014 -0400

    almost finished seat checking

commit 90bfbac1c8134a87d16caf89c9ff66104f8b7fb7
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date: Thu Aug 28 14:30:07 2014 -0400

    fixed wishlist null ptr exception

commit ca4a6921005e89dace34226560921c9770a82574
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date: Thu Aug 28 11:03:19 2014 -0400

    grade checker hotfix

commit a23213b973f66bc3e2799cd5012f1a5f6fca783e
Merge: ee4df1f 8185947
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date: Thu Aug 28 10:41:13 2014 -0400

    solve merge

commit ee4df1f79e4c5c23c25df3e8873a282c3ddb0cdc
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date: Thu Aug 28 10:33:26 2014 -0400

    added grade checker to the login and splash activity

commit bfb12e8d9b8d13b90f5ec0fd0044ff9edf01b28b
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date: Sat Aug 23 10:58:34 2014 -0400

    added grade comparison on a per course basis

commit 6d4ee5bbb3c37973d26db50754c37014fd0f3031
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date: Sat Aug 23 10:21:39 2014 -0400

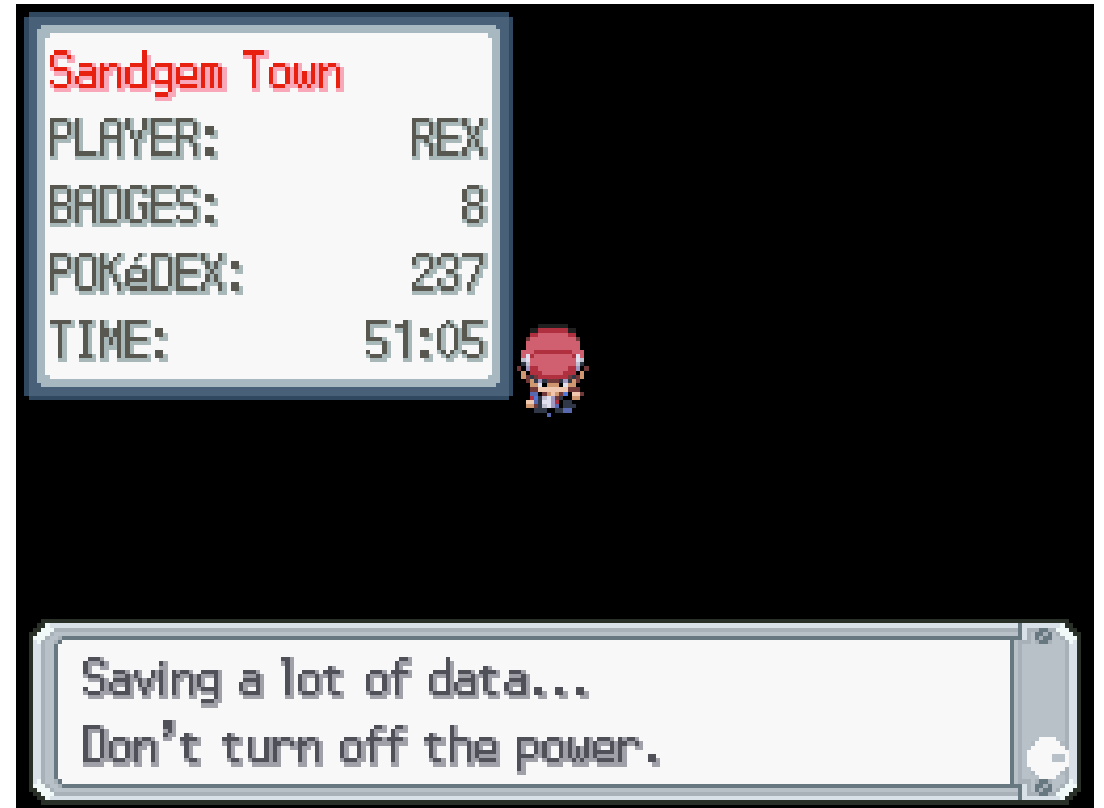
    added notification method

commit 4b91420ed909987ef840d89f0e3383219afdcc46
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date: Wed Aug 20 11:34:24 2014 -0400

    changed Activity class to Context in networking senarios

commit fb8b570faf5c04419fd1b727a90cc251a6457243
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date: Wed Aug 20 10:59:05 2014 -0400
```

Commits



Save states

Adding and committing a file

1

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)  
$ touch helloworld.java
```

2

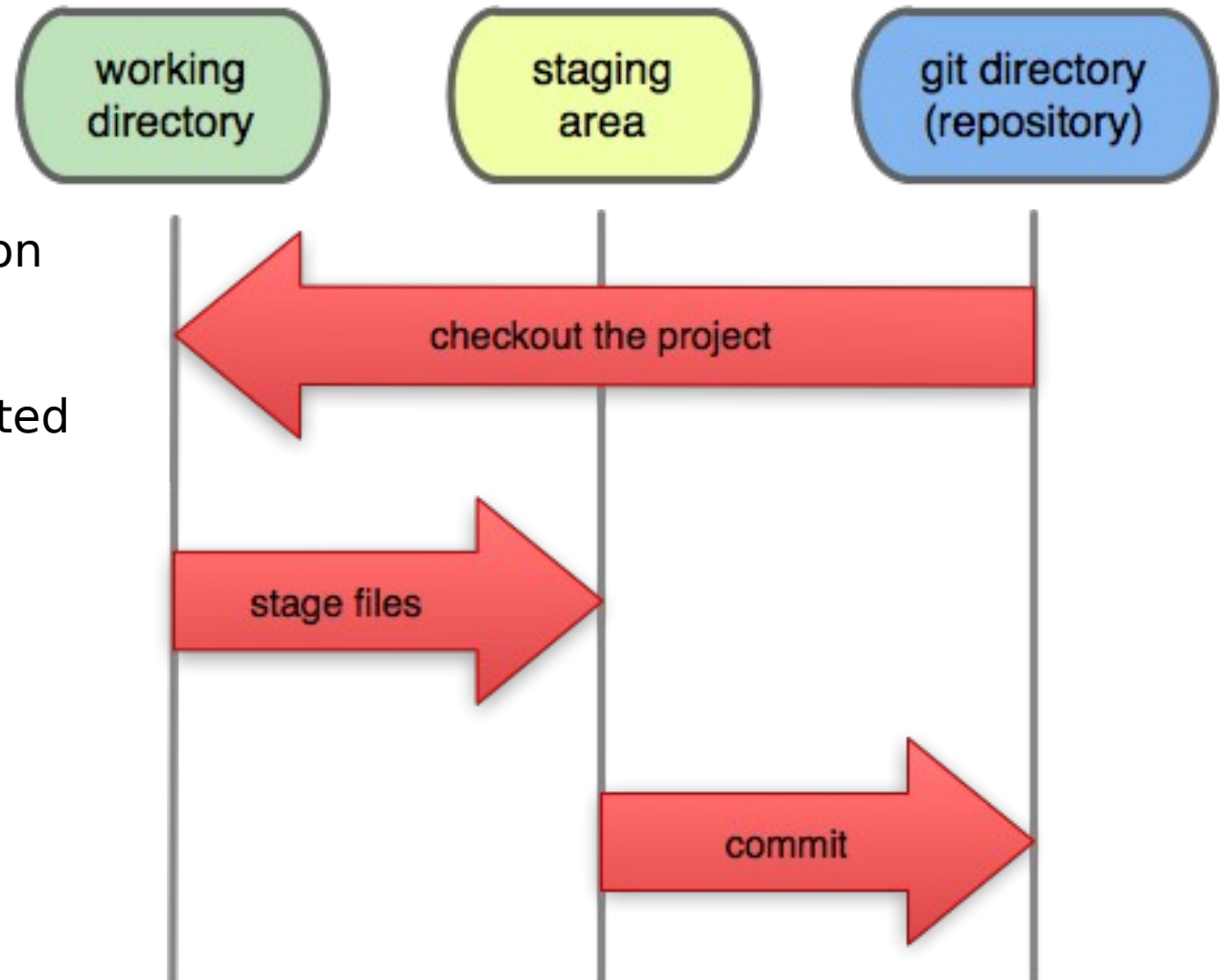
```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)  
$ git add helloworld.java
```

3

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)  
$ git commit -m 'added hello world file to the project'  
[master (root-commit) f4a1ddc] added hello world file to the project  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 helloworld.java
```

Inside a Repo

Local Operations



- Working Directory: Files being worked on right now
- Staging area: Files ready to be committed
- Repository: A collection of commits

Why should we stage files instead of directly committing them?

File Status Updates

1

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ vi helloworld.java
```

2

```
public class helloworld{

    public static void main(String[] args){
        System.out.println("Hello World");
    }

}
```

3

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (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 directory)
#
#       modified:   helloworld.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

Staging and Unstaging files

1

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ javac helloworld.java
```


2

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git add .
```

3

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   helloworld.class
#       modified:   helloworld.java
#
```


Staging and Unstaging files



```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git reset helloworld.class
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   helloworld.java
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       helloworld.class
```


Committing files (cont.)

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git commit -m 'wrote a working version of hello world'
[master c7b0bc6] wrote a working version of hello world
1 file changed, 6 insertions(+)
```



Only Staged files are committed.

Reviewing commits



```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git log
commit c7b0bc617da921ea6ede009700edaaad6183dc98
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date:   Fri Sep 5 09:45:47 2014 -0400

    wrote a working version of hello world

commit f4a1dddc8b8b35deee1ac69eb5e605b254606181f
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date:   Tue Sep 2 22:50:02 2014 -0400

    added hello world file to the project
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
```

Tagging commits

1

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/
$ git log
commit c7b0bc617da921ea6ede009700edaaad6183dc98
Author: Shabbir Hussain <mohd.husn001@gmail.com>
Date:   Fri Sep 5 09:45:47 2014 -0400

    wrote a working version of hello world
```

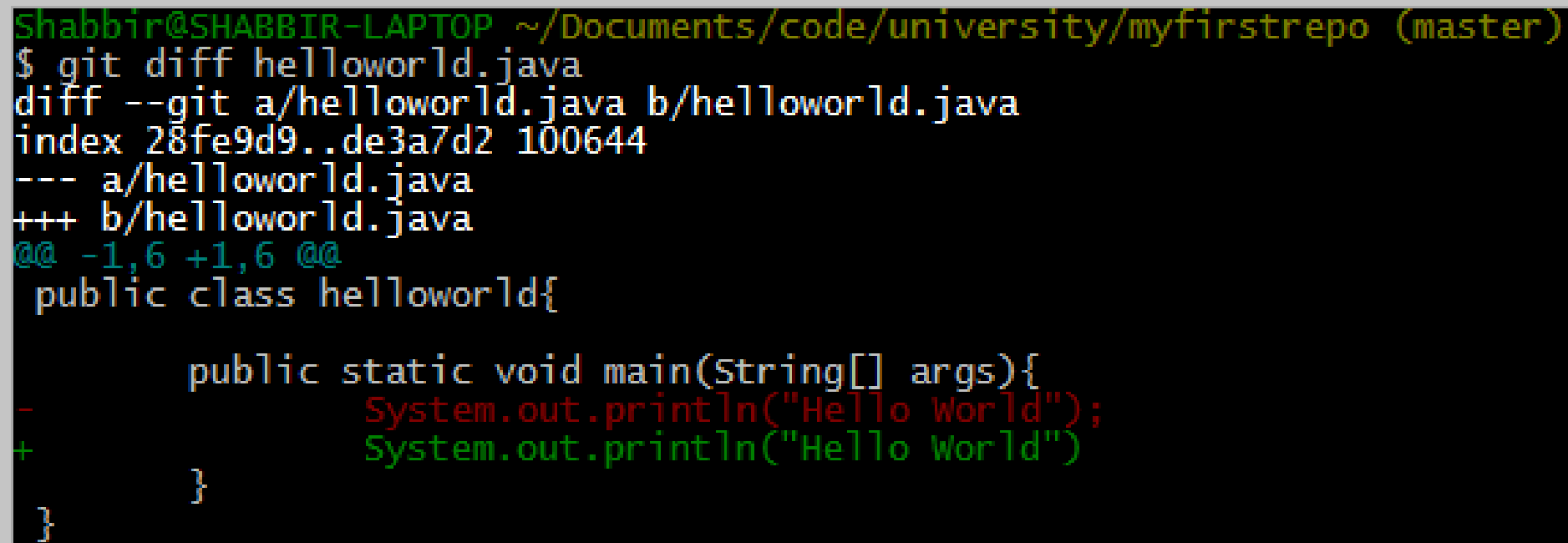
2

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git tag v1.0 c7b0bc617
```

3

```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git log --graph --oneline --decorate --all
* c7b0bc6 (HEAD, tag: v1.0, master) wrote a working version of hello world
* f4a1ddc (origin/master) added hello world file to the project
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
```

Finding Mistakes with Diff



```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git diff helloworld.java
diff --git a/helloworld.java b/helloworld.java
index 28fe9d9..de3a7d2 100644
--- a/helloworld.java
+++ b/helloworld.java
@@ -1,6 +1,6 @@
 public class helloworld{

     public static void main(String[] args){
-        System.out.println("Hello World");
+        System.out.println("Hello World")
     }
 }
```

Fixing Mistakes with checkout

```
public class helloworld{  
    public static void main(String[] args){  
        System.out.println("Hello World")  
    }  
}
```

No semicolon



```
Shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)  
$ git checkout helloworld.java
```

```
public class helloworld{  
    public static void main(String[] args){  
        System.out.println("Hello World");  
    }  
}
```

Replaced semicolon




Uploading your local Repository to the cloud

1

GitHub, Inc. [US] <https://github.com/new>

2

Owner:  shabbir-hussain / Repository name: ✓

Great repository names are short and memorable. Need inspiration? How about [laughing-adventure](#).

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 allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

Add .gitignore: **None** | Add a license: **None** ⓘ

Create repository

Uploading your local Repository to the cloud

3

done this kind of thing before

HTTP

SSH

git@github.com:shabbir-hussain/justatemprepo.git



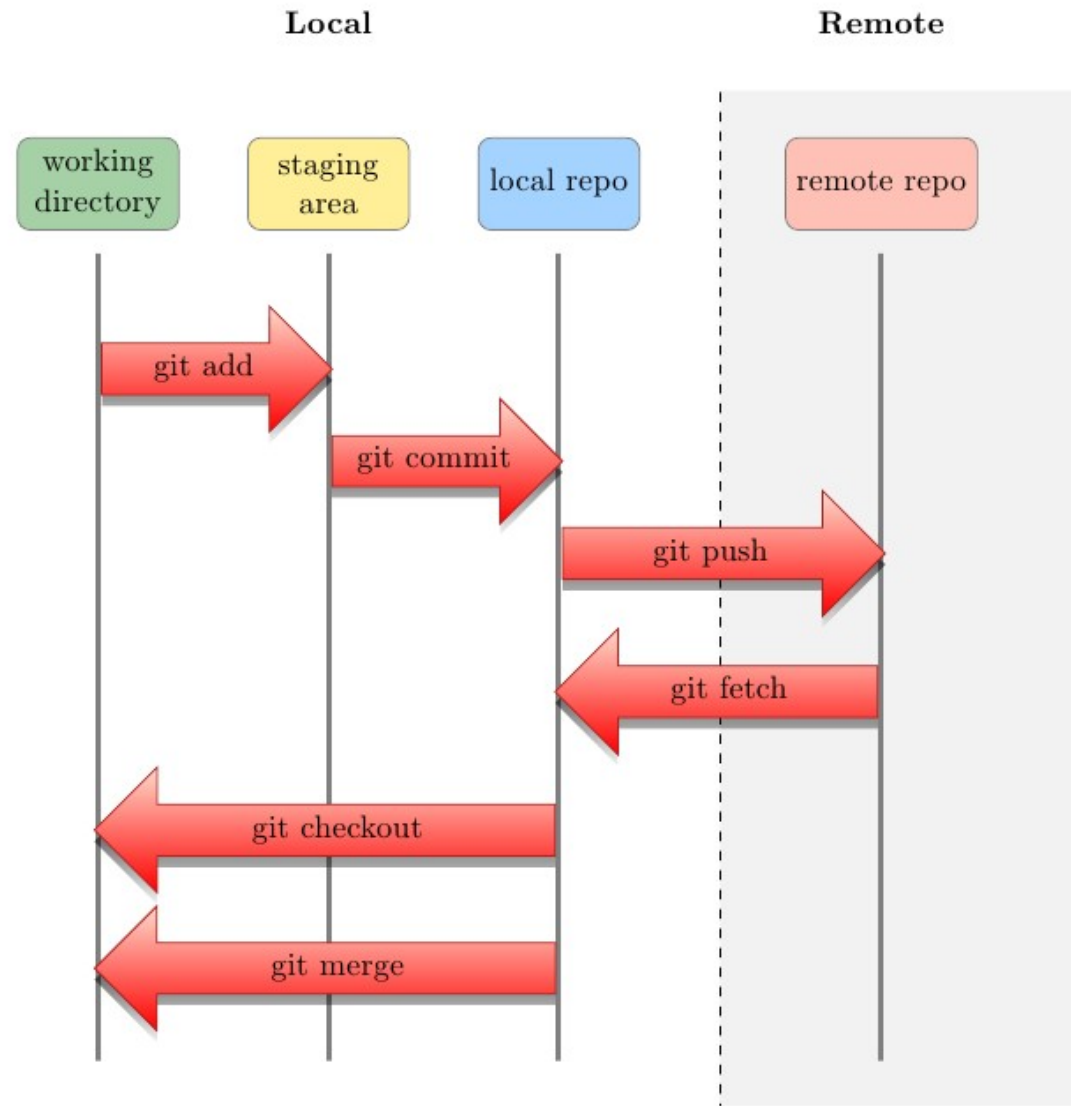
4

```
shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git remote add origin git@github.com:shabbir-hussain/justatemprepo.git
```

5

```
shabbir@SHABBIR-LAPTOP ~/Documents/code/university/myfirstrepo (master)
$ git push -u origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 242 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:shabbir-hussain/justatemprepo.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```


Uploading your local Repository to the cloud



Extra Credit

1. Try github Interactive Tutorial:

<https://try.github.io/levels/1/challenges/1>

2. Visual Explanation: <http://www.wei-wang.com/ExplainGitWithD3/>

More resources

1.Vogella Reference:

<http://www.vogella.com/tutorials/Git/article.html>

2.SSH keys tutorial:

<https://help.github.com/articles/generating-ssh-keys>

3.Git Cheatsheet:

<https://raw.githubusercontent.com/nerdgirl/git-cheatsheet-visual/master/gitcheatsheet.png>

4.Dominic's Tutorial slides (Previous TA):

<http://slides.com/dominiccharleyroy/tutorial-1-git#/>