

## STEP 1:

```
$ git config --global user.name "Samanjate Sood"
$ git config --global user.email "samanjate@ccs.neu.edu"
$ git config --global core.editor "atom --wait"
$ git config --list
core.excludesfile=~/.gitignore
core.legacyheaders=false
core.quotepath=false
core.editor=atom --wait
mergetool.keepbackup=true
push.default=simple
color.ui=auto
color.interactive=auto
repack.usedeltabaseoffset=true
alias.s=status
alias.a=!git add . && git status
alias.au=!git add -u . && git status
alias.aa=!git add . && git add -u . && git status
alias.c=commit
alias.cm=commit -m
alias.ca=commit --amend
alias.ac=!git add . && git commit
alias.acm=!git add . && git commit -m
alias.l=log --graph --all --pretty=format:%C(yellow)%h%C(cyan)%d%Creset %s %C(white)- %an,
%ar%Creset'
alias.ll=log --stat --abbrev-commit
alias.lg=log --color --graph --pretty=format:%C(bold white)%h%Creset -%C(bold green)%d%Creset
%s %C(bold green)(%cr)%Creset %C(bold blue)<%an>%Creset' --abbrev-commit --date=relative
alias.llg=log --color --graph --pretty=format:%C(bold white)%H %d%Creset%n%s%n%+b%C(bold
blue)%an <%ae>%Creset %C(bold green)%cr (%ci)' --abbrev-commit
alias.d=diff
alias.master=checkout master
alias.spull=svn rebase
alias.spush=svn dcommit
alias.alias=!git config --list | grep 'alias\.' | sed 's/alias\\.\\([^\=]*\\)=\\(.*\\)/\\1\\ => \\2/' | sort
include.path=~/.gitcininclude
include.path=.githubconfig
include.path=.gitcredential
diff.exif.textconv=exif
credential.helper=osxkeychain
user.name=Samanjate Sood
user.email=samanjate@ccs.neu.edu
credential.helper=osxkeychain
core.editor=atom --wait
```

## STEP 2:

```
$ cd ~/Desktop/
:Desktop $ git clone https://github.ccs.neu.edu/CS5500/student-137
Cloning into 'student-137'...
Username for 'https://github.ccs.neu.edu': samanjate
Password for 'https://samanjate@github.ccs.neu.edu':
remote: Counting objects: 13, done.
remote: Total 13 (delta 0), reused 0 (delta 0), pack-reused 13
Unpacking objects: 100% (13/13), done.
$ cd student-137/
:student-137 $ git ls-tree --full-tree -r HEAD
100644 blob 14b456b185450f9677c1d11ab6f668e642d2069c HW1/readme.txt
100644 blob 9377b9ef7b4097dc9e338d551cd6ef3efd492690 HW2/readme.txt
100644 blob 30f6484db5f8eba0f2857669972f088c00c1e9e8 HW3/readme.txt
100644 blob 7fa29633abe6748b98cdf4b15839246f7f35cdee HW4/readme.txt
100644 blob 6f5201e0fa6402e43611dc0b91f62e4622b128ae README.md
```

## STEP 3:

Use the square brackets '[]' with the text you wish to display on the readme file, followed by brackets '()' with the link to the page the user will navigate to when he/she clicks on the text.

## STEP 4:

Since we will be using Java for our project the file extensions included in the .gitignore file are: .log, .temp, .class, .jar, .war, .ear, .zip, .tar-gz, .rar.

```
$ git add .
$ git status
On branch master
Your branch is up to date with 'origin/master'.
```

Changes to be committed:  
(use "git reset HEAD <file>..." to unstage)

```
new file: .gitignore
modified: README.md
```

```
$ git commit -m "Updated the README file and setting up gitignore for the project."
```

```
[master e710887] Updated the README file and setting up gitignore for the project.
2 files changed, 34 insertions(+), 2 deletions(-)
create mode 100644 .gitignore
rewrite README.md (100%)
Samanjates-MacBook-Pro:student-137 samansood$ git log
commit e710887fbe9db856a3859c0b341987e33341efb6 (HEAD -> master)
Author: Samanjate Sood <samanjate@ccs.neu.edu>
Date: Thu Jan 18 23:05:49 2018 -0500
```

Updated the README file and setting up gitignore for the project.

```
commit 79ad032903cc02cffe2518aa1e6ffca13e269dcc (origin/master, origin/HEAD)
Author: vikas91 <chandrashekar.v@husky.neu.edu>
Date: Tue Jan 9 19:08:35 2018 -0500
```

initial commit

```
commit 24f33d8515820c6699041f05108f3551c329d02c
Author: vikas91 <chandrashekar.v@husky.neu.edu>
Date: Tue Jan 9 15:30:56 2018 -0500
```

Initial commit

## STEP 5:

```
$ git branch feature-1
$ git checkout feature-1
Switched to branch 'feature-1'
$ git branch
* feature-1
  master
$ cd HW1/
```

## STEP 6:

The feature-1 branch has 3 commits. It updates the .gitignore file to ignore the IDE and OS related files. The second commit adds the test spec's and the interface definition. The third adds the function logic which passes the test case written before. The 'HW1' directory contains the 'src' and 'test' directories. Each of the folder is a source folder for java. The code is under package com.neu.msd for both the source folders. Please see commits below to see the files that were added in this step.

```
$ git commit -m "Updated .gitignore file"
[feature-1 c921ff1] Updated .gitignore file
```

```
1 file changed, 27 insertions(+)
create mode 100644 .gitignore
$ git status
On branch feature-1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
```

```
HW1/src/
HW1/test/
```

```
nothing added to commit but untracked files present (use "git add" to track)
$ git add .
$ git status
On branch feature-1
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
```

```
new file:   HW1/src/com/neu/msd/Printer.java
new file:   HW1/src/com/neu/msd/Range.java
new file:   HW1/src/com/neu/msd/RangImp.java
new file:   HW1/test/com/neu/msd/TestRange.java
```

```
$ git commit -m "Satisfying requirement one: interface and test"
[feature-1 886de22] Satisfying requirement one: interface and test
4 files changed, 170 insertions(+)
create mode 100644 HW1/src/com/neu/msd/Printer.java
create mode 100644 HW1/src/com/neu/msd/Range.java
create mode 100644 HW1/src/com/neu/msd/RangImp.java
create mode 100644 HW1/test/com/neu/msd/TestRange.java
$ git commit -m "Satisfying requirement one: logic"
[feature-1 9072d1d] Satisfying requirement one: logic
2 files changed, 12 insertions(+), 2 deletions(-)
```

## STEP 7:

```
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
$ git merge feature-1
Merge made by the 'recursive' strategy.
 HW1/src/com/neu/msd/Printer.java | 49
+++++++
 HW1/src/com/neu/msd/Range.java   | 23 ++++++
```

```
HW1/src/com/neu/msd/RangImp.java | 36
```

```
+++++
```

```
HW1/test/com/neu/msd/TestRange.java | 72
```

```
+++++
```

```
+++++
```

```
4 files changed, 180 insertions(+)
```

```
create mode 100644 HW1/src/com/neu/msd/Printer.java
```

```
create mode 100644 HW1/src/com/neu/msd/Range.java
```

```
create mode 100644 HW1/src/com/neu/msd/RangImp.java
```

```
create mode 100644 HW1/test/com/neu/msd/TestRange.java
```

## STEP 8:

```
$ git branch feature-2
```

## STEP 9:

```
$ git branch feature-3
```

## STEP 10:

```
$ git checkout feature-2
```

```
Switched to branch 'feature-2'
```

```
$ git branch
```

```
feature-1
```

```
* feature-2
```

```
feature-3
```

```
master
```

```
$ git status
```

```
On branch feature-2
```

```
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: src/com/neu/msd/RangImp.java
```

```
modified: test/com/neu/msd/TestRange.java
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git add .
```

```
$ git commit -m "satisfying requirement change two"
```

```
[feature-2 68551b7] satisfying requirement change two
```

2 files changed, 16 insertions(+), 13 deletions(-)

## STEP 11:

```
$ git checkout feature-3
Switched to branch 'feature-3'
$ git branch
  feature-1
  feature-2
* feature-3
  master
$ git status
On branch feature-3
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:   src/com/neu/msd/RangImp.java
    modified:   test/com/neu/msd/TestRange.java
```

no changes added to commit (use "git add" and/or "git commit -a")

```
$ git add .
$ git commit -m "satisfying requirement change three"
[feature-3 c51f29f] satisfying requirement change three
2 files changed, 16 insertions(+), 13 deletions(-)
```

## STEP 12:

```
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 3 commits.
  (use "git push" to publish your local commits)
$ git merge feature-2
Updating d771912..e8cfbd0
Fast-forward
 HW1/src/com/neu/msd/RangImp.java | 5 +++--
 HW1/test/com/neu/msd/TestRange.java | 24 ++++++++-----
2 files changed, 16 insertions(+), 13 deletions(-)
```

## STEP 13:

```
$ git merge feature-3
Auto-merging HW1/test/com/neu/msd/TestRange.java
```

CONFLICT (content): Merge conflict in HW1/test/com/neu/msd/TestRange.java  
Auto-merging HW1/src/com/neu/msd/RangImp.java  
CONFLICT (content): Merge conflict in HW1/src/com/neu/msd/RangImp.java  
Automatic merge failed; fix conflicts and then commit the result.

The branch feature-1 is identical, and the differences between the other two branches are shown below, before the merge.

```
$ git diff master feature-1
```

```
$ git diff master feature-2
```

```
diff --git a/HW1/src/com/neu/msd/RangImp.java  
b/HW1/src/com/neu/msd/RangImp.java  
index d7b5a7b..a10aa40 100644
```

```
--- a/HW1/src/com/neu/msd/RangImp.java  
+++ b/HW1/src/com/neu/msd/RangImp.java
```

```
@@ -1,7 +1,6 @@
```

```
package com.neu.msdc;
```

```
import java.util.ArrayList;
```

```
-import java.util.Collections;
```

```
import java.util.List;
```

```
/**
```

```
@@ -29,7 +28,9 @@ public class RangImp implements Range {  
    }
```

```
    List<Integer> nums = new ArrayList<Integer>();
```

```
    for(int i = num1 + 1; i < num2; i++) {
```

```
-        nums.add(i);
```

```
+        if(i % 2 != 0) {
```

```
+            nums.add(i);
```

```
+        }
```

```
    }
```

```
    return nums;
```

```
}
```

```
diff --git a/HW1/test/com/neu/msd/TestRange.java
```

```
b/HW1/test/com/neu/msd/TestRange.java
```

```
index 73e01cf..c4bd4c9 100644
```

```
--- a/HW1/test/com/neu/msd/TestRange.java
```

```
+++ b/HW1/test/com/neu/msd/TestRange.java
```

```
@@ -11,14 +11,14 @@ import org.junit.Test;
```

```
/**
```

```
*
```

```
* To test different classes implementing the Range Interface
```

```
*
```

```

* @author Samanjate Sood
*
*/
public class TestRange {

```

```

    /**
$ git diff master feature-3
diff --git a/HW1/src/com/neu/msd/RangImp.java
b/HW1/src/com/neu/msd/RangImp.java
index d7b5a7b..cf8ef0d 100644
--- a/HW1/src/com/neu/msd/RangImp.java
+++ b/HW1/src/com/neu/msd/RangImp.java
@@ -1,7 +1,6 @@
package com.neu.msdc;

```

```

import java.util.ArrayList;
-import java.util.Collections;
import java.util.List;

```

```

    /**
@@ -29,7 +28,9 @@ public class RangImp implements Range {
    }
    List<Integer> nums = new ArrayList<Integer>();
    for(int i = num1 + 1; i < num2; i++) {
-        nums.add(i);
+        if(i % 2 == 0) {
+            nums.add(i);
+        }
    }
    return nums;
}

```

```

diff --git a/HW1/test/com/neu/msd/TestRange.java
b/HW1/test/com/neu/msd/TestRange.java
index 73e01cf..a3d082e 100644
--- a/HW1/test/com/neu/msd/TestRange.java
+++ b/HW1/test/com/neu/msd/TestRange.java
@@ -11,14 +11,14 @@ import org.junit.Test;

```

```

    /**
*
* To test different classes implementing the Range Interface
*
* @author Samanjate Sood
*
*/

```



```
public class TestRange {
```

```
    /**
```

```
$ git status
```

```
On branch master
```

```
Your branch is ahead of 'origin/master' by 4 commits.
```

```
(use "git push" to publish your local commits)
```

```
All conflicts fixed but you are still merging.
```

```
(use "git commit" to conclude merge)
```

```
Changes to be committed:
```

```
    modified:   src/com/neu/msd/RangImp.java
```

```
    modified:   test/com/neu/msd/TestRange.java
```

```
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:   src/com/neu/msd/RangImp.java
```

```
    modified:   test/com/neu/msd/TestRange.java
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
    src/com/neu/msd/RangImp.java.orig
```

```
    test/com/neu/msd/TestRange.java.orig
```

```
$ git add .
```

```
$ git commit -m "kept requirement change three"
```

```
[master d510324] kept requirement change three
```

## STEP 14:

```
$ git diff master feature-1
```

```
diff --git a/HW1/src/com/neu/msd/RangImp.java
```

```
b/HW1/src/com/neu/msd/RangImp.java
```

```
index a10aa40..d7b5a7b 100644
```

```
--- a/HW1/src/com/neu/msd/RangImp.java
```

```
+++ b/HW1/src/com/neu/msd/RangImp.java
```

```
@@ -1,6 +1,7 @@
```

```
package com.neu.msdc;
```

```
import java.util.ArrayList;
```

```
+import java.util.Collections;
import java.util.List;
```

```
/**
@@ -28,9 +29,7 @@ public class RangeImp implements Range {
    }
    List<Integer> nums = new ArrayList<Integer>();
    for(int i = num1 + 1; i < num2; i++) {
-        if(i % 2 != 0) {
-            nums.add(i);
-        }
+        nums.add(i);
    }
    return nums;
}
```

**diff --git a/HW1/test/com/neu/msd/TestRange.java**  
**b/HW1/test/com/neu/msd/TestRange.java**  
**index c4bd4c9..73e01cf 100644**

**--- a/HW1/test/com/neu/msd/TestRange.java**  
**+++ b/HW1/test/com/neu/msd/TestRange.java**

```
@@ -11,14 +11,14 @@ import org.junit.Test;
/**
 *
 * To test different classes implementing the Range Interface
- *
+ *
 * @author Samanjate Sood
 *
 */
public class TestRange {

    /**
-    *
+    *
    * Consists of a mix of function testing for all possible scenarios
    */
    @Test
@@ -34,10 +34,10 @@ public class TestRange {
    validateRange(-54, -22);
    validateRange(-14, -22);
}
-
+
/**
```

```

-      * Helper function for testRangeImp()
+      *
+      * Validates that the list returned by the function false in the range
+      * of the given numbers
+      */
@@ -49,25 +49,23 @@ public class TestRange {
    num1 = num2;
    num2 = temp;
}
+    int expectedListSize = (num2 - num1 - 1);
+    expectedListSize = expectedListSize < 0 ? 0 : expectedListSize;
    Set<Integer> expectedNumbers = new HashSet<Integer>();
    for(int i = num1 + 1; i < num2; i++) {
-        if(i % 2 != 0) {
-            expectedNumbers.add(i);
-        }
+        expectedNumbers.add(i);
    }
-    int expectedListSize = expectedNumbers.size();
    assertNotNull(output1);
-    assertEquals("The size of the list returned is incorrect",
-        output1.size(),
+    assertEquals("The size of the list returned is incorrect",
+        output1.size(),
        expectedListSize);
    for(Integer i : output1) {
-        assertTrue("The list either has a duplicate value or an unexpected value",
+        assertTrue("The list either has a duplicate value or an unexpected value",
            expectedNumbers.remove(i));
        assertTrue("The list contains number outside the given range", i > num1);
        assertTrue("The list contains number outside the given range", i < num2);
-        assertTrue("The number is even", i % 2 != 0);
    }

-
+    assertTrue("The list doesnot contain all the expected values",
        expectedNumbers.isEmpty());
    }

```

\$ git diff master feature-2

\$ git diff master revert-feature-3

\$ git diff master feature-3

**diff --git a/HW1/src/com/neu/msd/RangeImp.java**

**b/HW1/src/com/neu/msd/RangImp.java**

**index a10aa40..cf8ef0d 100644**

**--- a/HW1/src/com/neu/msd/RangImp.java**

**+++ b/HW1/src/com/neu/msd/RangImp.java**

```
@@ -28,7 +28,7 @@ public class RangImp implements Range {
    }
    List<Integer> nums = new ArrayList<Integer>();
    for(int i = num1 + 1; i < num2; i++) {
-        if(i % 2 != 0) {
+        if(i % 2 == 0) {
            nums.add(i);
        }
    }
}
```

**diff --git a/HW1/test/com/neu/msd/TestRange.java**

**b/HW1/test/com/neu/msd/TestRange.java**

**index c4bd4c9..a3d082e 100644**

**--- a/HW1/test/com/neu/msd/TestRange.java**

**+++ b/HW1/test/com/neu/msd/TestRange.java**

```
@@ -51,7 +51,7 @@ public class TestRange {
    }
    Set<Integer> expectedNumbers = new HashSet<Integer>();
    for(int i = num1 + 1; i < num2; i++) {
-        if(i % 2 != 0) {
+        if(i % 2 == 0) {
            expectedNumbers.add(i);
        }
    }

@@ -65,7 +65,7 @@ public class TestRange {
        expectedNumbers.remove(i));
    assertTrue("The list contains number outside the given range", i > num1);
    assertTrue("The list contains number outside the given range", i < num2);
-    assertTrue("The number is even", i % 2 != 0);
+    assertTrue("The number is even", i % 2 == 0);
    }

    assertTrue("The list doesnot contain all the expected values",
expectedNumbers.isEmpty());
```

## **STEP 15:**

\$ git branch

feature-1

feature-2

feature-3

\* master

```
    revert-feature-3
$ git push --all origin
Counting objects: 53, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (29/29), done.
Writing objects: 100% (53/53), 5.40 KiB | 920.00 KiB/s, done.
Total 53 (delta 13), reused 0 (delta 0)
remote: Resolving deltas: 100% (13/13), done.
To https://github.ccs.neu.edu/CS5500/student-137
    79ad032..e8cfbd0  master -> master
* [new branch]    feature-1 -> feature-1
* [new branch]    feature-2 -> feature-2
* [new branch]    feature-3 -> feature-3
* [new branch]    revert-feature-3 -> revert-feature-3
```

## **STEP 16:**

```
$ git push origin master
```