Managing Software Development Assignment 1

- Veera Venkata Sasanka Uppu

1) What are your git config's?

A) git config --list

```
core.excludesfile=~/.gitignore
core.legacyheaders=false
core.quotepath=false
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=~/.gitcinclude
include.path=.githubconfig
include.path=.gitcredential
diff.exif.textconv=exif
credential.helper=osxkeychain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=Veera Venkata Sasanka Uppu
user.email=uppu.v@husky.neu.edu
core.editor=vi
```

2) What files or directories are in the .git directory?

A) Directories - branches
hooks
info
logs
objects
refs

Files - config
description
HEAD
index
packed-refs

---

3) Using markdown, how would one express a link to the course web page (https://course.ccs.neu.edu/cs5500)?

A) We use [ ] braces for the clickable text and put the link in a following ( ) braces. For example - This is a repo for [CS5500 - MSD](https://course.ccs.neu.edu/cs5500) Assignments

4) What rules are in your .gitignore?

A)
```
# Compiled class file
*.class

# Log file
*.log
# BlueJ files
*.ctxt
# Mobile Tools for Java (J2ME)
.mtj.tmp/

# Package Files #
*.jar
*.war
*.ear
*.zip
*.tar.gz
*.rar
```

---

6) What are all the commits you've made on the feature-1 branch? What files and sub-directories are listed in your hmk1 directory on the feature-1 branch. What files and subdirectories are listed in your hmk1 directory on the master branch. Insure it is easy to tell which listing goes with which branch.

A) "satisfying requirement one" commit is made on feature-1 branch.

Files and sub directories present in hmk1 folder for feature-1 branch are - src/Range.java
src/RangePrinter.java
test/RangePrinterTest.java
.classpath
.project

Whereas for master the hmk1 folder is empty

---

10) What are all the commits you've made on the feature-2 branch?

A) "satisfying requirement two" is the only commit made on feature-2 branch

---

13) What are the differences between master and the other three branches? Hint: use git diff.

A) Master and Feature-1: feature-1 has the implementation to find all the numbers in between two numbers. Whereas master has the implementation to find all the even numbers in between two numbers.

```
➔  git diff master feature-1
diff --git a/hmk1/.gitignore b/hmk1/.gitignore
deleted file mode 100644
index ae3c172..0000000
--- a/hmk1/.gitignore
+++ /dev/null
@@ -1 +0,0 @@
-/bin/
diff --git a/hmk1/src/Range.java b/hmk1/src/Range.java
index e84900b..c341fbc 100644
--- a/hmk1/src/Range.java
+++ b/hmk1/src/Range.java
@@ -5,11 +5,11 @@ import java.util.List;
 */
 public interface Range {
 /**
-    * Given two integers the function returns a list of even integers in between
+    * Given two integers the function returns a list of integers in between
 * the two excluding the given two numbers
 * @param num1 is the lower limit in the range
 * @param num2 is the upper limit in the range
-    * @return list of even integers which fall in the range
+    * @return list of integers which fall in the range
 */
```

```
public List<Integer> rangeCalculator(int num1, int num2);
}
diff --git a/hmk1/src/RangePrinter.java b/hmk1/src/RangePrinter.java
index 694c2c0..3b826f8 100644
--- a/hmk1/src/RangePrinter.java
+++ b/hmk1/src/RangePrinter.java
@@ -10,11 +10,11 @@ public class RangePrinter implements Range{

@Override
/**
-    * Given two integers the function returns a list of even integers in between
+    * Given two integers the function returns a list of integers in between
* the two excluding the given two numbers
* @param num1 is the lower limit in the range
* @param num2 is the upper limit in the range
-    * @return list of even integers which fall in the range
+    * @return list of integers which fall in the range
*/
public List<Integer> rangeCalculator(int num1, int num2) {

@@ -27,10 +27,7 @@ public class RangePrinter implements Range{

for (int i=num1+1;i<num2;i++)
{
-                if(Math.abs(i)%2==0)
-                {
-                    temp.add(i);
-                }
+        temp.add(i);
}
return temp;
}
diff --git a/hmk1/test/RangePrinterTest.java b/hmk1/test/RangePrinterTest.java
index 4dd4285..3b3b18a 100644
--- a/hmk1/test/RangePrinterTest.java
+++ b/hmk1/test/RangePrinterTest.java
@@ -2,10 +2,6 @@ import static org.junit.Assert.*;
import java.util.Arrays;
import org.junit.Test;

-/**
- * Test Cases for rangeCalculator method in RangePrinter.java class
- */
-
public class RangePrinterTest {

RangePrinter rp;
@@ -16,11 +12,11 @@ public class RangePrinterTest {
}

/**
-    * The range for (3,7) should be (4,6)
+    * The range for (3,7) should be (4,5,6)
--- a/hmk1/src/Range.java
diff --git a/hmk1/.gitignore b/hmk1/.gitignore
deleted file mode 100644
index ae3c172..0000000
--- a/hmk1/.gitignore
+++ /dev/null
@@ -1 +0,0 @@
-/bin/
diff --git a/hmk1/src/Range.java b/hmk1/src/Range.java
```

```
index e84900b..c341fbc 100644
--- a/hmk1/src/Range.java
+++ b/hmk1/src/Range.java
@@ -5,11 +5,11 @@ import java.util.List;
*/
public interface Range {
/**
-    * Given two integers the function returns a list of even integers in betwe
en
+    * Given two integers the function returns a list of integers in between
* the two excluding the given two numbers
* @param num1 is the lower limit in the range
* @param num2 is the upper limit in the range
-    * @return list of even integers which fall in the range
+    * @return list of integers which fall in the range
```

Master and Feature-2: feature-2 has the implementation to find all the odd numbers in between two numbers. Whereas master has the implementation to find all the even numbers in between two numbers.

```
➔  git diff master feature-2
diff --git a/hmk1/src/Range.java b/hmk1/src/Range.java
index e84900b..07259af 100644
--- a/hmk1/src/Range.java
+++ b/hmk1/src/Range.java
@@ -5,11 +5,11 @@ import java.util.List;
 */
public interface Range {
    /**
-    * Given two integers the function returns a list of even integers in between
+    * Given two integers the function returns a list of odd integers in between
     * the two excluding the given two numbers
     * @param num1 is the lower limit in the range
     * @param num2 is the upper limit in the range
-    * @return list of even integers which fall in the range
+    * @return list of odd integers which fall in the range
     */
    public List<Integer> rangeCalculator(int num1, int num2);
}
diff --git a/hmk1/src/RangePrinter.java b/hmk1/src/RangePrinter.java
index 694c2c0..5438b02 100644
--- a/hmk1/src/RangePrinter.java
+++ b/hmk1/src/RangePrinter.java
@@ -10,11 +10,11 @@ public class RangePrinter implements Range{

    @Override
    /**
-    * Given two integers the function returns a list of even integers in between
+    * Given two integers the function returns a list of odd integers in between
     * the two excluding the given two numbers
     * @param num1 is the lower limit in the range
     * @param num2 is the upper limit in the range
-    * @return list of even integers which fall in the range
+    * @return list of odd integers which fall in the range
     */
    public List<Integer> rangeCalculator(int num1, int num2) {

@@ -27,9 +27,9 @@ public class RangePrinter implements Range{
```

```
      for (int i=num1+1;i<num2;i++)
      {
-                 if(Math.abs(i)%2==0)
+                  if(Math.abs(i)%2==1)
                  {
-                       temp.add(i);
+                            temp.add(i);
                  }
      }
      return temp;
diff --git a/hmk1/test/RangePrinterTest.java b/hmk1/test/RangePrinterTest.java
index 4dd4285..d8ea730 100644
--- a/hmk1/test/RangePrinterTest.java
+++ b/hmk1/test/RangePrinterTest.java
@@ -16,11 +16,11 @@ public class RangePrinterTest {
      }

    /**
-    * The range for (3,7) should be (4,6)
    public List<Integer> rangeCalculator(int num1, int num2) {

@@ -27,9 +27,9 @@ public class RangePrinter implements Range{

      for (int i=num1+1;i<num2;i++)
      {
-                 if(Math.abs(i)%2==0)
+                  if(Math.abs(i)%2==1)
                  {
-                       temp.add(i);
+                            temp.add(i);
                  }
      }
      return temp;
diff --git a/hmk1/test/RangePrinterTest.java b/hmk1/test/RangePrinterTest.java
index 4dd4285..d8ea730 100644
--- a/hmk1/test/RangePrinterTest.java
+++ b/hmk1/test/RangePrinterTest.java
@@ -16,11 +16,11 @@ public class RangePrinterTest {
      }

    /**
-    * The range for (3,7) should be (4,6)
+    * The range for (3,7) should be (5)
     */
      @Test
      public void test1() {
-            assertEquals(Arrays.asList(4,6), rp.rangeCalculator(3, 7));
+             assertEquals(Arrays.asList(5), rp.rangeCalculator(3, 7));
      }

      /**
@@ -48,11 +48,11 @@ public class RangePrinterTest {
      }

      /**
-    * The range for (-2, 3) should be (0,2)
+    * The range for (-2, 3) should be (-1,1)
     */
      @Test
      public void test5() {
-            assertEquals(Arrays.asList(0,2), rp.rangeCalculator(-2, 3));
```

```
+           assertEquals(Arrays.asList(-1,1), rp.rangeCalculator(-2, 3));
    }
```

Master and feature-3: Master branch and feature-3 branch are both the same as feature-3 has just been merged to master, so the difference is null

---

14) QUESTION: What are the differences now between master and the other three branches? Hint: use git diff.

A) Master and Feature-1: feature-1 has the implementation to find all the numbers in between two numbers. Whereas master has the implementation to find all the odd numbers in between two numbers.

```
➔  git diff master feature-1
diff --git a/hmk1/src/Range.java b/hmk1/src/Range.java
index 07259af..c341fbc 100644
--- a/hmk1/src/Range.java
+++ b/hmk1/src/Range.java
@@ -5,11 +5,11 @@ import java.util.List;
 */
public interface Range {
   /**
-   * Given two integers the function returns a list of odd integers in between
+   * Given two integers the function returns a list of integers in between
    * the two excluding the given two numbers
    * @param num1 is the lower limit in the range
    * @param num2 is the upper limit in the range
-   * @return list of odd integers which fall in the range
+   * @return list of integers which fall in the range
    */
   public List<Integer> rangeCalculator(int num1, int num2);
}
diff --git a/hmk1/src/RangePrinter.java b/hmk1/src/RangePrinter.java
index 5438b02..3b826f8 100644
--- a/hmk1/src/RangePrinter.java
+++ b/hmk1/src/RangePrinter.java
@@ -10,11 +10,11 @@ public class RangePrinter implements Range{

   @Override
   /**
-   * Given two integers the function returns a list of odd integers in between
+   * Given two integers the function returns a list of integers in between
    * the two excluding the given two numbers
    * @param num1 is the lower limit in the range
    * @param num2 is the upper limit in the range
-   * @return list of odd integers which fall in the range
+   * @return list of integers which fall in the range
    */
   public List<Integer> rangeCalculator(int num1, int num2) {

@@ -27,10 +27,7 @@ public class RangePrinter implements Range{

      for (int i=num1+1;i<num2;i++)
      {
-               if(Math.abs(i)%2==1)
```

```
-                {
-                        temp.add(i);
-                }
+        temp.add(i);
        }
        return temp;
    }
diff --git a/hmk1/test/RangePrinterTest.java b/hmk1/test/RangePrinterTest.java
index d8ea730..3b3b18a 100644
--- a/hmk1/test/RangePrinterTest.java
+++ b/hmk1/test/RangePrinterTest.java
@@ -2,10 +2,6 @@ import static org.junit.Assert.*;
 import java.util.Arrays;
 import org.junit.Test;

-/**
- * Test Cases for rangeCalculator method in RangePrinter.java class
- */
-
 public class RangePrinterTest {

     RangePrinter rp;
@@ -16,11 +12,11 @@ public class RangePrinterTest {
     }

    /**
-    * The range for (3,7) should be (5)
+    * The range for (3,7) should be (4,5,6)
     */
     @Test
     public void test1() {
-            assertEquals(Arrays.asList(5), rp.rangeCalculator(3, 7));
+            assertEquals(Arrays.asList(4,5,6), rp.rangeCalculator(3, 7));
     }

     /**
@@ -48,11 +44,11 @@ public class RangePrinterTest {
     }

     /**
-    * The range for (-2, 3) should be (-1,1)
+    * The range for (-2, 3) should be (-1,0,1,2)
     */
     @Test
     public void test5() {
-            assertEquals(Arrays.asList(-1,1), rp.rangeCalculator(-2, 3));
+            assertEquals(Arrays.asList(-1,0,1,2), rp.rangeCalculator(-2, 3));
     }

 }
```

Master and feature-2: Master branch and feature-2 branch are both the same as master has been with revert-fearture-3 branch, so the difference is null

Master and Feature-3: feature-3 has the implementation to find all the even numbers in between two numbers. Whereas master has the implementation to find all the odd numbers in between two numbers.

```
➔  git diff master feature-3
diff --git a/hmk1/src/Range.java b/hmk1/src/Range.java
index 07259af..e84900b 100644
--- a/hmk1/src/Range.java
+++ b/hmk1/src/Range.java
@@ -5,11 +5,11 @@ import java.util.List;
 */
public interface Range {
    /**
-    * Given two integers the function returns a list of odd integers in between
+    * Given two integers the function returns a list of even integers in between
     * the two excluding the given two numbers
     * @param num1 is the lower limit in the range
     * @param num2 is the upper limit in the range
-    * @return list of odd integers which fall in the range
+    * @return list of even integers which fall in the range
     */
    public List<Integer> rangeCalculator(int num1, int num2);
}
diff --git a/hmk1/src/RangePrinter.java b/hmk1/src/RangePrinter.java
index 5438b02..694c2c0 100644
--- a/hmk1/src/RangePrinter.java
+++ b/hmk1/src/RangePrinter.java
@@ -10,11 +10,11 @@ public class RangePrinter implements Range{

    @Override
    /**
-    * Given two integers the function returns a list of odd integers in between
+    * Given two integers the function returns a list of even integers in between
     * the two excluding the given two numbers
     * @param num1 is the lower limit in the range
     * @param num2 is the upper limit in the range
-    * @return list of odd integers which fall in the range
+    * @return list of even integers which fall in the range
     */
    public List<Integer> rangeCalculator(int num1, int num2) {

@@ -27,9 +27,9 @@ public class RangePrinter implements Range{

        for (int i=num1+1;i<num2;i++)
        {
-               if(Math.abs(i)%2==1)
+               if(Math.abs(i)%2==0)
                {
-                       temp.add(i);
+                       temp.add(i);
                }
        }
        return temp;
diff --git a/hmk1/test/RangePrinterTest.java b/hmk1/test/RangePrinterTest.java
index d8ea730..4dd4285 100644
--- a/hmk1/test/RangePrinterTest.java
+++ b/hmk1/test/RangePrinterTest.java
@@ -16,11 +16,11 @@ public class RangePrinterTest {
    }
```

```
    /**
-    * The range for (3,7) should be (5)
+    * The range for (3,7) should be (4,6)
     */
      @Test
      public void test1() {
-            assertEquals(Arrays.asList(5), rp.rangeCalculator(3, 7));
+             assertEquals(Arrays.asList(4,6), rp.rangeCalculator(3, 7));
      }

      /**
@@ -48,11 +48,11 @@ public class RangePrinterTest {
      }

      /**
-    * The range for (-2, 3) should be (-1,1)
+    * The range for (-2, 3) should be (0,2)
     */
      @Test
      public void test5() {
-            assertEquals(Arrays.asList(-1,1), rp.rangeCalculator(-2, 3));
+             assertEquals(Arrays.asList(0,2), rp.rangeCalculator(-2, 3));
      }

 }
```