lab7
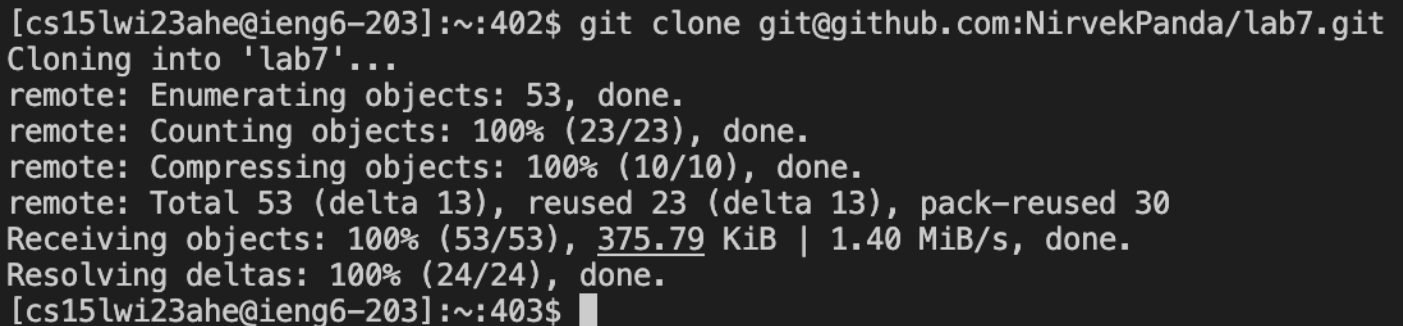
Lab Report 4

For the lab report this week, reproduce the task from the competition on your own. For each numbered step starting right after the timer (so steps 4-9), take a screenshot, and write down exactly which keys you pressed to get to that step. For special characters like or , write them in angle brackets with code formatting. Then, summarize the commands you ran and what the effect of those keypresses were.

Clone Repository

```
git clone git@github.com:NirvekPanda/lab7.git
ls
cd lab7
ls
```

```
[cs15lwi23ahe@ieng6-203]:~:402$ git clone git@github.com:NirvekPanda/lab7.git
Cloning into 'lab7'...
remote: Enumerating objects: 53, done.
remote: Counting objects: 100% (23/23), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 53 (delta 13), reused 23 (delta 13), pack-reused 30
Receiving objects: 100% (53/53), 375.79 KiB | 1.40 MiB/s, done.
Resolving deltas: 100% (24/24), done.
[cs15lwi23ahe@ieng6-203]:~:403$
```

```
the first command clones a repo from a given source
cd changes the current directory to the new one
```

Test Code

```
javac -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar *.java
java -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar org.junit.runner.JUnitCore
```

```
[cs15lwi23ahe@ieng6-203]:~:404$ cd lab7
[cs15lwi23ahe@ieng6-203]:lab7:405$ ls
ListExamples.java  TestListExamples.java  index.md  lib
[cs15lwi23ahe@ieng6-203]:lab7:406$ javac -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar *.java
[cs15lwi23ahe@ieng6-203]:lab7:407$ java -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar org.junit.runner.JUnitCore ListExamplesTests
JUnit version 4.13.2
.E
Time: 0.002
There was 1 failure:
1) initializationError(org.junit.runner.JUnitCommandLineParseResult)
java.lang.IllegalArgumentException: Could not find class [ListExamplesTests]
        at org.junit.runner.JUnitCommandLineParseResult.parseParameters(JUnitCommandLineParseResult.java:100)
        at org.junit.runner.JUnitCommandLineParseResult.parseArgs(JUnitCommandLineParseResult.java:50)
        at org.junit.runner.JUnitCommandLineParseResult.parse(JUnitCommandLineParseResult.java:44)
        at org.junit.runner.JUnitCore.runMain(JUnitCore.java:72)
        at org.junit.runner.JUnitCore.main(JUnitCore.java:36)
Caused by: java.lang.ClassNotFoundException: ListExamplesTests
        at java.net.URLClassLoader.findClass(URLClassLoader.java:387)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:418)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:352)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:351)
        at java.lang.Class.forName0(Native Method)
        at java.lang.Class.forName(Class.java:348)
        at org.junit.internal.Classes.getClass(Classes.java:42)
        at org.junit.internal.Classes.getClass(Classes.java:27)
        at org.junit.runner.JUnitCommandLineParseResult.parseParameters(JUnitCommandLineParseResult.java:98)
        ... 4 more

FAILURES!!!
Tests run: 1,  Failures: 1
```

the first command compiles all the files
the second command runs the tester file

Debug Code

```
nano ListExamples.java
    <down> 42 times
    change 1 to 2 to update correct variables
Ctrl + O
Ctrl + X
```

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

interface StringChecker { boolean checkString(String s); }

class ListExamples {

  // Returns a new list that has all the elements of the input list for which
  // the StringChecker returns true, and not the elements that return false, in
  // the same order they appeared in the input list;
  static List<String> filter(List<String> list, StringChecker sc) {
    List<String> result = new ArrayList<>();
    for(String s: list) {
      if(sc.checkString(s)) {
        result.add(0, s);
      }
    }
    return result;
  }


  // Takes two sorted list of strings (so "a" appears before "b" and so on),
  // and return a new list that has all the strings in both list in sorted order.
  static List<String> merge(List<String> list1, List<String> list2) {
    List<String> result = new ArrayList<>();
    int index1 = 0, index2 = 0;
    while(index1 < list1.size() && index2 < list2.size()) {
      if(list1.get(index1).compareTo(list2.get(index2)) < 0) {
        result.add(list1.get(index1));
        index1 += 1;
      }
      else {
        result.add(list2.get(index2));
        index2 += 1;
      }
    }
    while(index1 < list1.size()) {
      result.add(list1.get(index1));
      index1 += 1;
    }
    while(index2 < list2.size()) {
      result.add(list2.get(index2));
      index2 += 1;
    }
    return result;
  }


}
```

```
^G Get Help    ^O WriteOut    ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is     ^V Next Page    ^U UnCut Text   ^T To Spell
```

```
nano opens the file
go down to the error and change it
save the changes
```

Test Code Again

```
javac -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar *.java
java -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar org.junit.runner.JUnitCore
```

```
[cs15lwi23ahe@ieng6-203]:lab7:418$ javac -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar *.java
[cs15lwi23ahe@ieng6-203]:lab7:419$ java -cp .:lib/hamcrest-core-1.3.jar:lib/junit-4.13.2.jar org.junit.runner.JUnitCore TestListExamples
JUnit version 4.13.2
..
Time: 0.021

OK (2 tests)
```

tests come back working

Commit and Push to Github

```
git add ListExamples.java
git commit —m "lab report 4"
git push origin main
```

```
[cs15lwi23ahe@ieng6-203]:lab7:421$ git add ListExamples.java
[cs15lwi23ahe@ieng6-203]:lab7:422$ git commit -m "lab report 4"
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ListExamples.class
        StringChecker.class
        TestListExamples.class

nothing added to commit but untracked files present (use "git add" to track)
[cs15lwi23ahe@ieng6-203]:lab7:423$ git add ListExamples.java ListExamples.class StringChecker.class TestListExamples.class
[cs15lwi23ahe@ieng6-203]:lab7:424$ git commit -m "lab report 4"
[main 7edb2e6] lab report 4
 Committer: Nirvek Pandey <cs15lwi23ahe@ieng6-203.ucsd.edu>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 ListExamples.class
 create mode 100644 StringChecker.class
 create mode 100644 TestListExamples.class
[cs15lwi23ahe@ieng6-203]:lab7:425$ git push orgin main
```

sends files back to github with the message