

UTM • Fall 2017

- [Syllabus](#)
- [Lectures and Labs](#)
- [Assignments](#)
- [Tests](#)
- [Resources](#)
- [Feedback](#)
- [Discussion Board](#)

Lab 1 (Week 2) Non OO Java, Eclipse and git

References

Arnold:
git clone ssh://UTORID@cslinux.utm.utoronto.ca/student/cslec/207/207prep.git
git clone ssh://UTORID@cslinux.utm.utoronto.ca/student/cslec/207/207_L0101.git

Larry:
git clone ssh://UTORID@cslinux.utm.utoronto.ca/student/cslec/207/207_L0102.git

Sadia:
git clone ssh://UTORID@cslinux.utm.utoronto.ca/student/cslec/207/207_L0103.git

Also, you might want to verify your utorid
https://www.utorid.utoronto.ca
see the bottom left...
https://www.utorid.utoronto.ca/cgi-bin/utorid/verify.pl

Git

We have created repositories for you,

1. Bring up a command prompt (terminal) then do the following

```
cd ~  
git clone https://mcsscm.utm.utoronto.ca/git/207/repo_$USER.git  
# NOTE: $USER will automatically be replaced by your UTORID in the command prompt on our systems.  
# From home, replace $USER with your UTORID.
```

2. Possible problem: git may complain as follows:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
@      WARNING: POSSIBLE DNS SPOOFING DETECTED!                  @  
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@  
The RSA host key for cslinux.utm.utoronto.ca has changed,  
and the key for the corresponding IP address 142.150.1.123  
has a different value. This could either mean that  
DNS SPOOFING is happening or the IP address for the host  
and its host key have changed at the same time.  
Offending key for IP in .../.ssh/known_hosts:11
```

If so, edit the known_hosts file and remove the offending line, in my case, line 11.

3. Notice that there is now a repo_\$USER directory in your main directory.

4. Please familiarize yourself with a few common unix commands...ls, cd, pwd and at least one text editor.

Running Eclipse

1. In a terminal launch Eclipse Oxygen via /opt/eclipse/eclipse
2. It will ask you to create a workspace, do it.
3. In Eclipse, Click on the top right to go to the workbench

LabWeek02

1. Follow along with your TA to create and run your first project (LabWeek02).

Summary:

File->New Java Project
Name the project LabWeek02 then Finish
Expand LabWeek02
Click on src
File->New->Class
Name: HelloWorld
then change its contents so that it matches that below

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

2. Save the files contents
3. Run it! (run icon on the top)
4. You should see "HelloWorld" in the console tab at the bottom.

HelloWorld2

1. Add class HelloWorld2 also with a main method so that when run, this class prints out "HelloWorld2!".
2. How do you run the main in class HelloWorld now?

Making Eclipse aware of the Repo

So far we have not been using the repo. Now we will make Eclipse aware of the repo.

1. On the top right, next to Quick Access, there is a box with a +, click to Open Perspective
2. Select the git perspective
3. In the Git Repositories tab, select the Git+ button which, when you hover over it, reads

Add an existing local Git repository to this view

4. Browse to add repo_\$USER to Eclipse. Then select the repo_\$USER/.git and Finish.
5. Eclipse can now make use of the repo
6. You might want to switch back to the Java Perspective (next to Quick Access, the button with aa J in it).

Using the Repo

1. right-click on the LabWeek02 project, and select Team > Share-Project
2. Choose your repo and Finish
3. Notice the change to all of the icons involved in the project
4. right-click the project, and select Team, then Add to Index (this is git add)

5. right-click the project, and select Team, then select Commit. Now put in the following commit message "My first project commit" and then commit and push. A password dialog should appear.

Adding more to your project

1. Add class SumNums with the following initial content...

```
import java.io.*;
import java.util.*;
public class SumNums {

    public static void main(String[] args) {
        Scanner scanner=new Scanner(System.in);

        String line;
        int start, end;
        while(true){
            System.out.println("Starting number: ");
            start=scanner.nextInt();
            System.out.println("Ending number: ");
            end=scanner.nextInt();
            int sum=0;
            // You have something to fix here!
            System.out.println("The sum from "+start+ " to "+ end + " is: "+ sum);
        }

        // Create static method sumup(s,e) which returns the sum s+(s+1)+...+(e-1)+e
        // or 0 if this sum does not make sense (ie sumup(3,-4)).
    }
}
```

2. Run the above code
3. Add it to the index (right click on the project and then team)
4. Commit and push it (right click on the project and then team)

A first User Story

1. You will complete the following User Story (see class SumNums)...

Story Card:
A user specifies a starting and ending number.
The ending number at least as big as the starting.
The system responds with the sum of all of the
numbers between and including
the starting and ending numbers.
We expect that the user will always supply integers.

2. If you are intersted in what a Scanner is (in SumNums), take a look at [the Java api](#), find the Scanner class (bottom left frame).
3. Make sure you add the public static int sumup(int s, int e) method.
4. For reference, see sum and sum2 in [NonOOSummary.java](#)
5. Run your program
6. Add all files Commit and push your project (right click LabWeek02, Team->Commit. Provide a useful message so that anyone that checks out your project will know why you made the changes.
7. Take a look at the repository in your browser as well as on your desktop. If you want to update your desktop copy, you can go into the LabWeek02 directory and commit changes.

Counting to 10

Story Card:

When the program is run, the system simply prints out the numbers from 1 to 10.

- 1. Create class CountTo10 (inside LabWeek02).
- 2. Create a 'main' method and put your counting and printing code in there.
- 3. Compile and run it.
- 4. Commit your changes to LabWeek02

Prime Numbers

Remember that an integer greater than 1 is prime if its only divisors are one and itself. So 2, 3, 5 and 101 are prime, while -7, 0 and 12 are not.

Story Card:

A user specifies a number n. The system responds with either "n is prime" or "n is not prime".

We expect that the user will supply a number.

- 1. Create class PrimeTester with a main method. Steal code from SumNums to get you started.
- 2. Write an isPrime(n) static method which returns **true** if n is prime and **false** otherwise. Use this in your main to help you complete your task. **Note:** In Java you can find the remainder on division of p by q by using p%q (this is p mod q).

```
if(p%q==0){
    System.out.println(p+" is divisible by "+q);
}
```

- 3. Make sure you commit LabWeek02, we may go in and check your progress (yes, we can check out your repository as well).