

UTM • Fall 2017

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

Lab 2 (Week 3) Writing an OO Java Program

Overview

This week, you are going to implement two Java classes. Given time, you should also write some test cases and document the classes by providing Javadoc comments.

Notes

- Please login and work on the lab systems!!
- Marks for the lab are for committing and pushing your work to the repo.
- During the lab, if you know what you are doing, and can help your neighbour, then help them. If you need help, ask us or your neighbour.

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`

Setup and Submission

- Launch Eclipse (if you see a blank screen, close the blank tab)
- Switch to Git perspective (or Window > Show View > Other > Git > Git Repositories) to view a list of all your git repositories in your Eclipse sidebar.
- See last weeks lab if you need to clone your repo and connect Eclipse to it.
- Right click on your repo_UTORID.git repo and select "Pull".
- File > Import > Git > Projects from git
- Select "Existing local repository"
- Select repo_UTORID

- Select "Import existing Eclipse project" and from the dropdown, select LabWeek03 (**the whole directory; don't browse into it**). Select "Next", then "Finish"
- Now switch to Java Perspective and work with the files in the LabWeek03 project
- As you work on the lab, commit your changes as follows: Right-click LabWeek03, and select Team > Add to Index (this is git add), then right-click again and select Team > Commit.
- In the "Git Staging" window, enter appropriate commit message, and then click the "Commit and Push" button.
- **SUBMISSION: Remember to "Commit and Push" your project at the end of your lab so we can see the most updated version of your lab work.**

1. Class SodaCan

Implement the class SodaCan using instance variables and methods to capture the following (see [hints](#)):

- A SodaCan has a **type** (i.e. Coke, Sprite, Root Beer), it is either **open or closed**, it has an **amount** of soda in it (at least 0).
- For this question, assume that SodaCans start out closed with 250cc of soda in them.
- A SodaCan be **opened** (but can't be closed once opened).
- Once open, either **sips** (remove 10cc) or **gulps** (remove 50cc) can be taken from them. Remember, amount must be at least 0 (i.e. if removing 10cc would make amount negative, amount should become 0). Both methods should return the amount of soda actually removed.
- The **string** representation (from method toString) of a SodaCan displays its type, whether or not it is open/closed, and its amount.

Once you are done implementing this class, **write a main method** which creates a few SodaCans and then takes sips and gulps from them.

2. Class Person

Implement the class Person using instance variables and methods to capture the following:

- A Person has a **thirstStatus** which is 'satisfied', 'thirsty' or 'very thirsty', and **amountDrunk** which keeps track of how much soda they have drunk so far.
- A very thirsty person has drank no soda. A thirsty person has already drank 175cc of soda. A satisfied Person has drank 375cc of soda.
- Initially, a Person is very thirsty.
- A person can **takeSip(SodaCan)**, they can also **takeGulp(SodaCan)**.
- The **string** representation of a Person displays their thirstStatus and their amountDrunk.

3. Playing with both classes

- Within a **main method**, create a few SodaCans and a few Persons.
- Open some of the SodaCans and have the Persons take some sips and gulps from them. You should see the amount of soda in the cans drop at the same time the amount of soda that the person has drunk going up. You should see some of the People going from 'very thirsty' to 'thirsty' to 'satisfied'.
- Given time, write some test cases and Javadoc your code.