# CSC 217 Lab 03

Collections

## Lab Overview

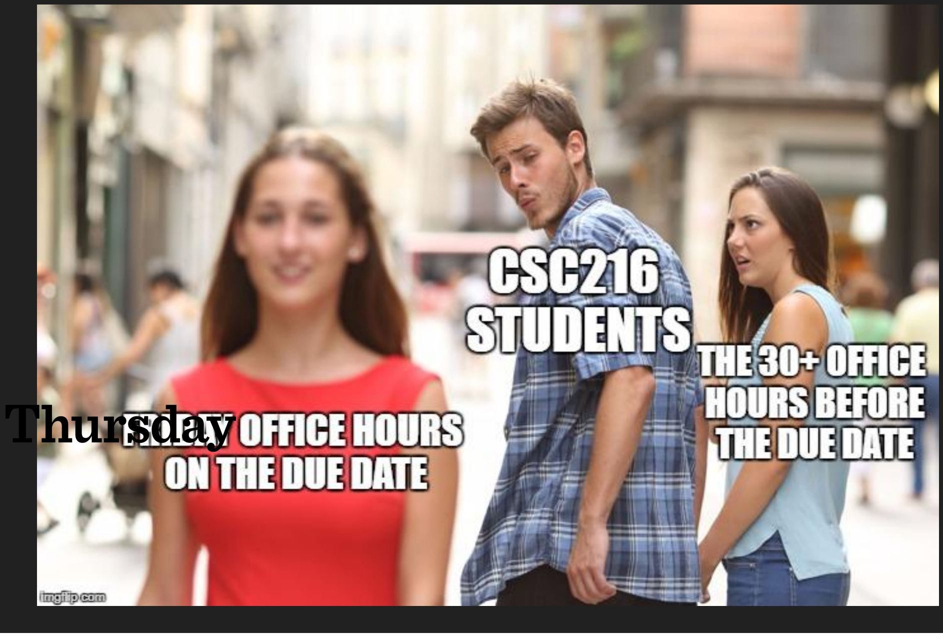
- Reminders
- Libraries
- Testing a Library
- Integrating the Library

# Reminders

- Deadlines
  - Lab 3 Due Week of September 20th
  - GP3 Due Thursday, September 16th at 11:45pm
    - Late Deadline Until September 18th at 11:45pm
- Reminders
  - Review grades and submit regrade within a week of the grade's return
  - Check Jenkins console output!!!!!
  - Javadoc
    - Generate HTML API pages from your Javadoc in the doc directory
    - Check the doc/ folder to ensure ALL files were created
    - Check NCSU GitHub to ensure that ALL files were pushed to the remote!
    - Check Piazza "Javadoc 101" post for main points - don't forget @throws for any exceptions thrown by the method/constructor
  - Out-of-lab
    - If you pair program, you MUST note it in the commit message or there will be deductions
    - Students who don't let their partner contribute will receive deductions
    - Teams that don't update their README.md will receive deductions

# Reminders

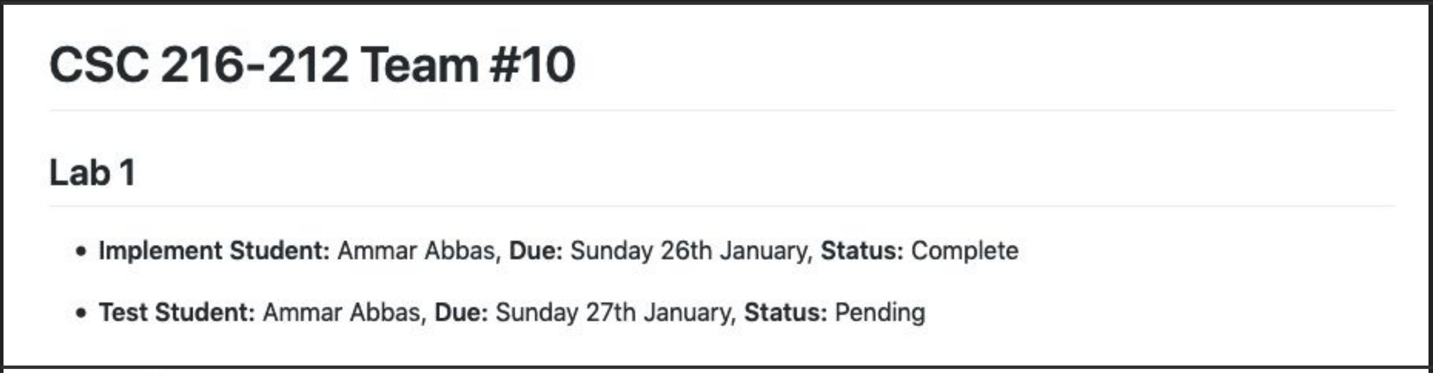
- Do NOT wait till the last minute to come to office hours!!!


CSC216 STUDENTS
THE 30+ OFFICE HOURS BEFORE THE DUE DATE
Thursday OFFICE HOURS ON THE DUE DATE
imgflip.com

## Markdown 101

| Format | Style |
|---|---|
| *Italics* | *Italics* |
| **Bold** | **Bold** |
| *  Bullet Point | ● Bullet Point |
| # Heading 1 | Heading 1 |
| ## Heading 2 | Heading 2 |
| ### Heading 3 | Heading 3 |

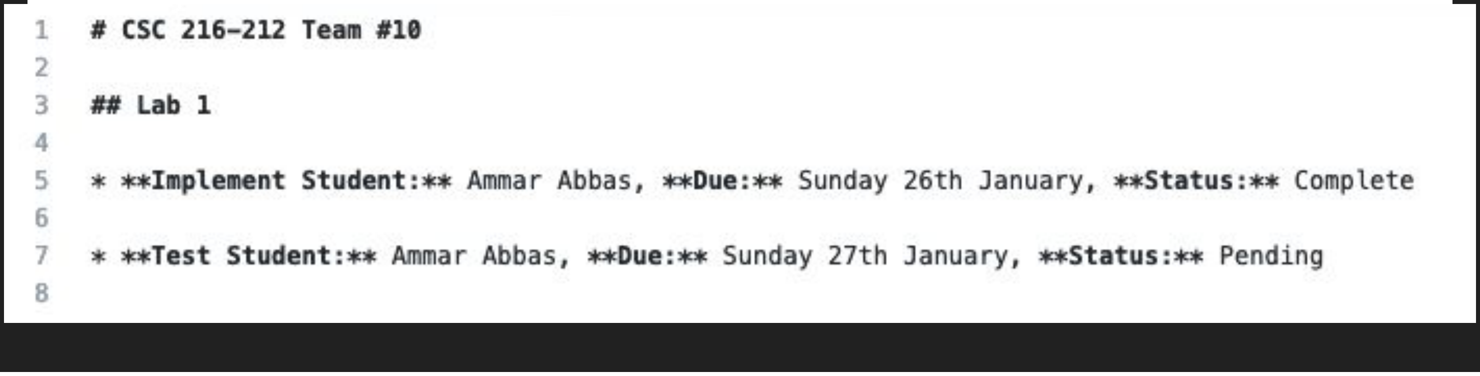
## Markdown 101 - Example



# CSC 216-212 Team #10

## Lab 1

- **Implement Student:** Ammar Abbas, **Due:** Sunday 26th January, **Status:** Complete
- **Test Student:** Ammar Abbas, **Due:** Sunday 27th January, **Status:** Pending



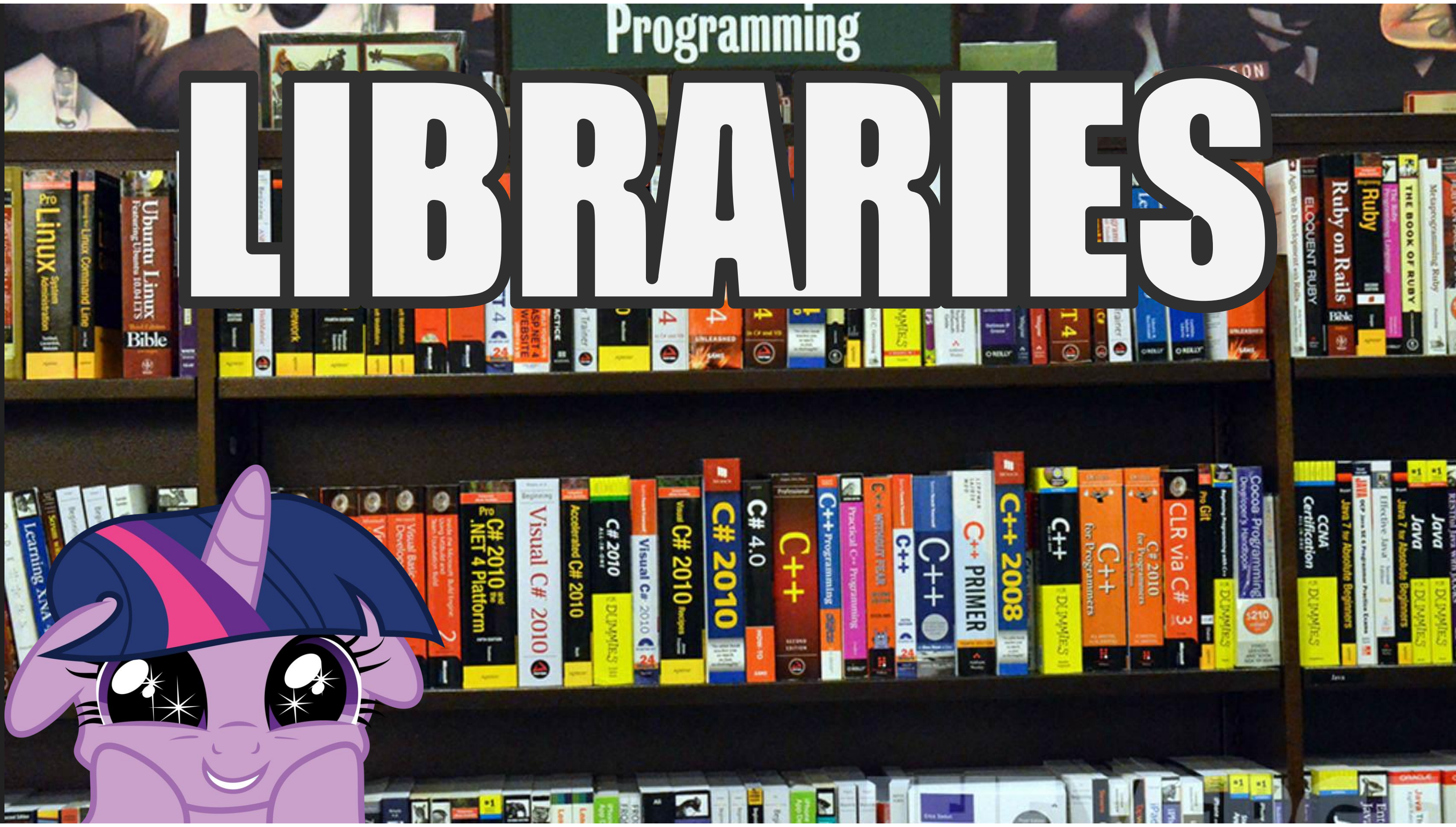
```
# CSC 216-212 Team #10

## Lab 1

* **Implement Student:** Ammar Abbas, **Due:** Sunday 26th January, **Status:** Complete

* **Test Student:** Ammar Abbas, **Due:** Sunday 27th January, **Status:** Pending

```

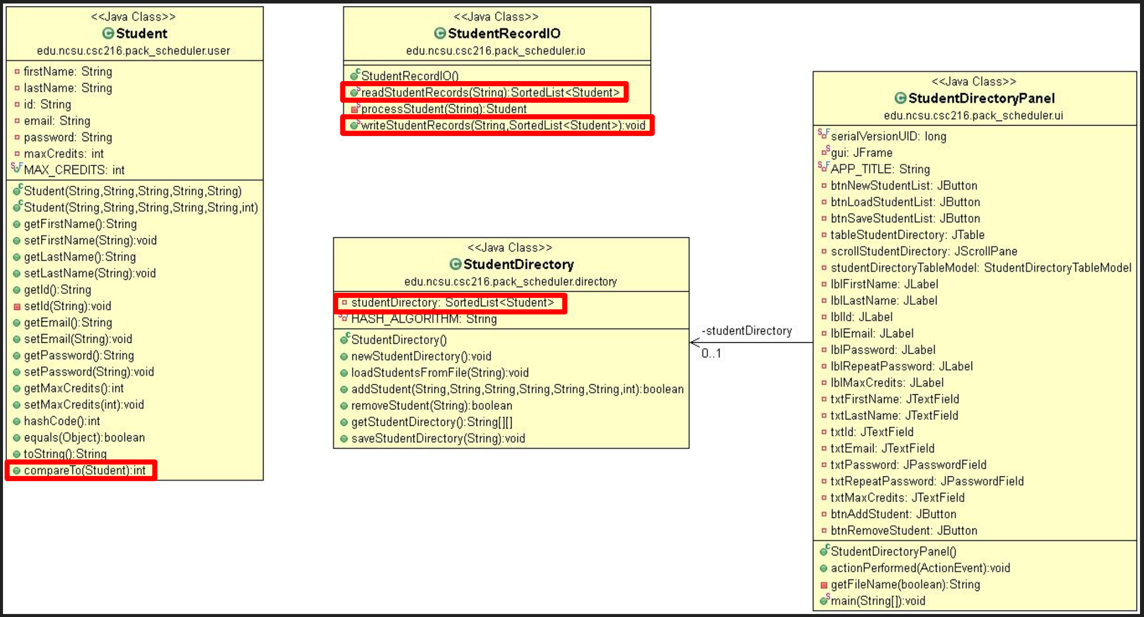


# CSC217Collections Library

Library contains SortedList class

- StudentRecordIO
- StudentDirectory

Student must implement Comparable



```
<<Java Class>>
Student
edu.ncsu.csc216.pack_scheduler.user
firstName: String
lastName: String
id: String
email: String
password: String
maxCredits: int
MAX_CREDITS: int
Student(String,String,String,String,String)
Student(String,String,String,String,String,int)
getFirstName():String
setFirstName(String):void
getLastName():String
setLastName(String):void
getId():String
setId(String):void
getEmail():String
setEmail(String):void
getPassword():String
setPassword(String):void
getMaxCredits():int
setMaxCredits(int):void
hashCode():int
equals(Object):boolean
toString():String
compareTo(Student):int
```

```
<<Java Class>>
StudentRecordIO
edu.ncsu.csc216.pack_scheduler.io
StudentRecordIO()
readStudentRecords(String):SortedList<Student>
processStudent(String):Student
writeStudentRecords(String,SortedList<Student>):void
```

```
<<Java Class>>
StudentDirectory
edu.ncsu.csc216.pack_scheduler.directory
studentDirectory: SortedList<Student>
HASH_ALGORITHM: String
StudentDirectory()
newStudentDirectory():void
loadStudentsFromFile(String):void
addStudent(String,String,String,String,String,String,int):boolean
removeStudent(String):boolean
getStudentDirectory():String[][]
saveStudentDirectory(String):void
```

```
<<Java Class>>
StudentDirectoryPanel
edu.ncsu.csc216.pack_scheduler.ui
serialVersionUID: long
gui: JFrame
APP_TITLE: String
btnNewStudentList: JButton
btnLoadStudentList: JButton
btnSaveStudentList: JButton
tableStudentDirectory: JTable
scrollStudentDirectory: JScrollPane
studentDirectoryTableModel: StudentDirectoryTableModel
lblFirstName: JLabel
lblLastName: JLabel
lblId: JLabel
lblEmail: JLabel
lblPassword: JLabel
lblRepeatPassword: JLabel
lblMaxCredits: JLabel
txtFirstName: JTextField
txtLastName: JTextField
txtId: JTextField
txtEmail: JTextField
txtPassword: JPasswordField
txtRepeatPassword: JPasswordField
txtMaxCredits: JTextField
btnAddStudent: JButton
btnRemoveStudent: JButton
StudentDirectoryPanel()
actionPerformed(ActionEvent):void
getFileName(boolean):String
main(String[]):void
```

# Lab Overview

Lab activities are in Moodle

- Partner 1 & 2: Pull the latest from the repo so that you're starting from a common set of code
- Write TEST code for the library…
- … while pair programming!
- Integrate the library…
- … while pair programming!
- If you have any technology questions, now is the time to troubleshoot!

# Writing Good List Tests

Test each scenario:

- empty list
- front of the list
- middle of the list
- end of the list

TODO in test skeleton to provide guidance on steps

After doing an action (i.e., add(), remove(), etc.), you should check

- The size of the list
- The element you modified
- **All** other elements in the list
- The return value from the method

Checking the size and/or a single element is NOT enough!

# Wrap-Up

General Wrap-Up

- Deadline Reminder
- Exchange contact information with your partner
- Make a plan for finishing up the lab

Participation Outside of Lab (Guess which the teaching staff prefer?)

- If you pair program, **note that in the commit comments so everyone gets credit!**
- If you split the work, at least one contribution by each partner

REMINDER: We are expecting a significant contribution from all team members outside of lab!

- If you pair program, you **MUST** note it in your commit messages or there will be deductions
- Student who don't allow their partners to contribute will also receive deductions
- Students who don't contribute will receive deductions

# Record Tasks & Owners

Tasks only get done when someone owns them!

Identify the tasks required to complete Lab 3

- Edit README.md to list the tasks required to complete Lab 3 (at top of README - should come before Lab 2 tasks)
- Add an owner to each task
- Add a deadline to each task

Deadlines should be at least 48 hours before the lab deadline so team members can help out and finish the lab if a team member runs into issues.

Notify team early if you run into problems with your tasks!  Communication is key!