



CS 398: Application Development

Week 06 Lecture: Daily Standup

TDD; Unit testing; Daily standups

This Week

You should be working through your assigned issues.

Today is a “regular working day”.

Reminder of the overall schedule:

	Mon	Wed	Fri
Week 1	Kickoff	(Standup) ork)	(Standup) ork)
Week 2	(Standup)	(Standup)	Demo

Fridays will be 2 hour classes (first hour mandatory, but Jeff will stay for the second hour if you want to continue working).

Daily Standup

You need to do this every day during the sprint (except Mon kickoff and Fri demo days).

Everyone on the team answers 3 questions:

1. What have you done since the last standup?
2. What are you doing today?
3. Is there anything blocking you and preventing you from progressing?

Fill out the Daily Standup sheet each class to track your progress.

This replaces the Meeting Minutes that you were taking.

<https://student.cs.uwaterloo.ca/~cs398/b-templates/files/daily-standup-online-template.pdf>

Daily Standup Minutes should be stored with your Meeting Minutes on GitLab!

CS 398: Daily Standup

Team #: _____

Date: _____

Questions

Enter comments below. If a team member is absent, fill in their name, and put a line through the answer section.

Name	What have you completed since last meeting?	What are you going to complete today?	What obstacles are in your way?

Daily Standup
Team 310
Wed Feb 9, 2022

Constant

- completed spec
- will design database interface today
- no issues

Jeff

- fixed UI scrolling bug
- will complete layout today
- needs help testing under Windows - Xiaoyan

Xiaoyan

- rewrote notes class to handle rich text
- will expand unit tests and fix failing test
- no issues

You can also just write them in a TXT file and save it in your repo.

<https://student.cs.uwaterloo.ca/~cs398/b-templates/files/daily-standup-online-template.pdf>

Development Model

1. Daily Standup

2. Pick a task from the list. Move it to Open.

3. Create a branch in Git - a feature branch for your task.

```
git checkout -b branch-name
```

4. Start coding!

If you're doing pair programming, decide who is driving and who is navigating.

If you're doing TDD, write unit tests first.