Lecture Notes
September 15, 2020

“Assignment I0” due before class today. Did anyone have any trouble doing the assignment?

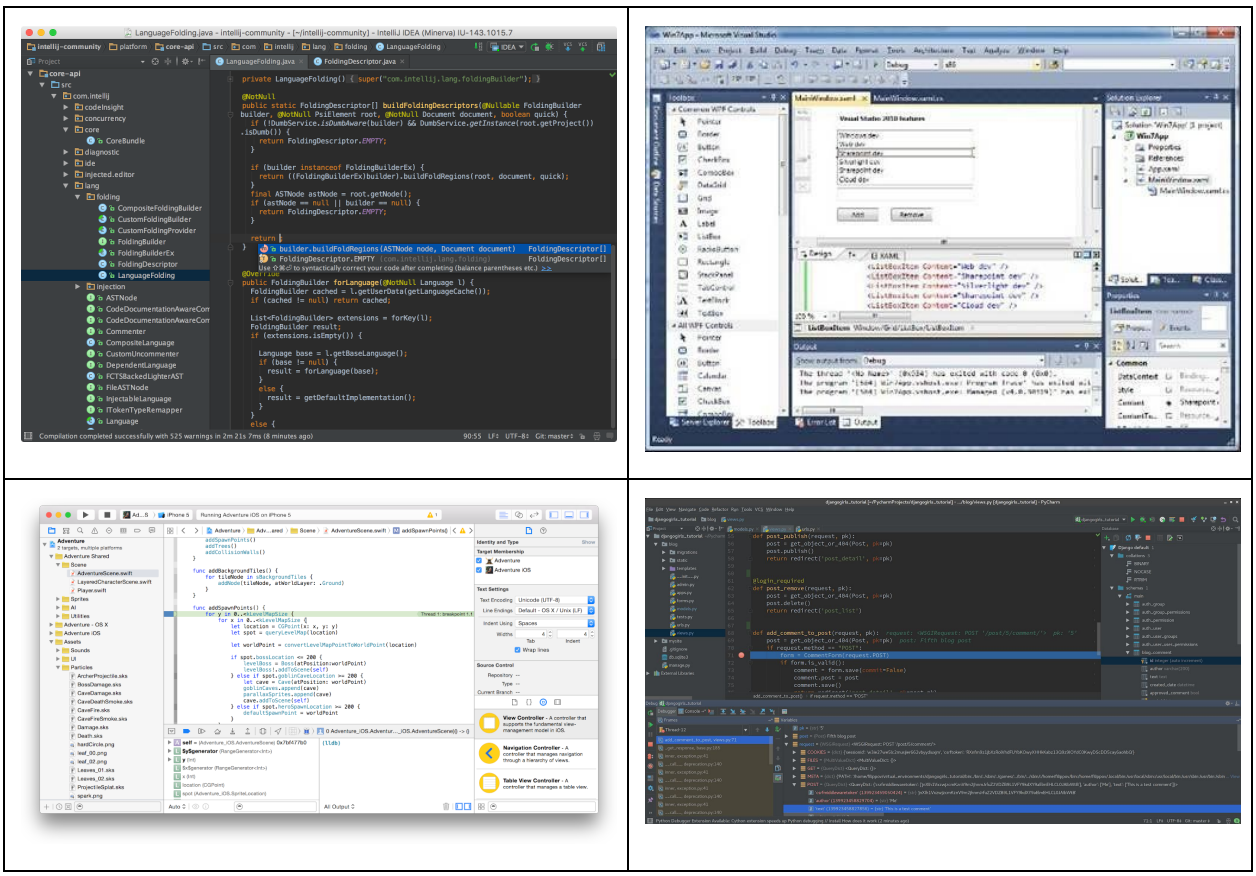
(Students who first joined the class this week have until next Tuesday morning to submit.)

Visit addendums from September 10 (pages 7 and 17).

Working in a team: *IDEs*

IDE = Integrated Development Environment

Comprehensive support for coding, testing and debugging



Minimum IDE features: edit, compile, run

Why bother? Why not use a standalone text editor and a standalone compiler, and run your program from the command line?

Some software engineers use **code editors** instead of IDEs, special text editors that “know” that you are writing code in a particular programming language. Code editors typically highlight language keywords and catch simple syntax errors, and may compile and run simple programs.

Code editors are usually designed for editing individual files one at a time, whereas IDEs are intended for projects with *many* files and make it easy to move between interdependent files. Many IDEs support code completion based on context of entire project.

Code editors are usually fast and easy to use, whereas IDEs can be slow to startup (since they load the entire project) and their UIs can be daunting because most IDEs do a lot more than edit, compile, run.

Typical features include: refactoring, integration with version control, code search, automated testing, test coverage tracking, interactive debugging, style checking, various static analysis, software metrics… We will cover most of these kinds of tools later on in course.

Eclipse, in particular, ‘does everything’. Besides the built-in features, there are plugin marketplaces.

You need to use Eclipse for the individual project, assignments I1-I3. (You do not need to use Eclipse for the team project, your team chooses your toolset.)

"Assignment I1": https://courseworks2.columbia.edu/courses/104335/assignments/482592

To do this assignment, you need to code a simple web application in Java (a tic-tac-toe game) using the Javalin lightweight framework. You have to use Java because the assignment provides skeleton code, in Java, for you to fill in. We provide the UI.

The list of download/installs may seem overwhelming, which is why we provide today's tutorial.

There will be two more individual programming assignments building on this one, which will use JUnit to test your tic-tac-toe code and then add a database for saving moves during the game (and test it). There will also be in-class tutorials corresponding to those two assignments.

Now Shirish will present his tutorial on what you need to do with Eclipse for assignment I1.