# ITP 368 Cross Platform Development

Viterbi logo

units: 4, 2025 spring, section 31852, Tu Th 12:00-13:50 (12pm-1:50pn), OHE 540
Instructor: Dr. Barrett Koster
Office: RRB 211 and zoom 501 267 3551
Office Hours: M W 13-15, T Th 14-15
Contact Info: bkoster ayat usc dawt edu
Learning Assistants:
Sumit Kothawade skothawa ayat usc dawt edu

## Course Description

In today's dynamic computing landscape, where devices span embedded systems, desktops, web browsers, and mobile platforms, the art of crafting seamless and consistent graphical user interfaces (GUIs) across various environments has become an intricate and resource-intensive endeavor. Meeting user expectations demands GUIs that remain synchronized, adaptable to multiple platforms, globally accessible, inclusive, high-performing, and aesthetically pleasing with each software iteration. Industry giants like Facebook, Microsoft, and Google have honed the skills to create such GUIs routinely. But how can you too learn these techniques?

Welcome to our Cross-Platform App Development course, where we delve into the methodologies employed by tech titans to construct GUIs that transcend platform boundaries. We unravel the techniques behind their world-class software and empower you with the tools to replicate their success. By grasping these essential techniques and principles, you will acquire the competence to navigate any GUI framework beyond the scope of this course. This newfound expertise will not only expedite your professional growth but also elevate your contributions to real-world software engineering.

**Catalog Description.** Learn to program cross-platform graphical user interfaces (GUIs) using the same techniques and concepts used by big tech companies.

## Learning Objectives

- Be able to write programs in Dart and Flutter.
- Be able to set up the IDE for such programming, including version control, collaboration, and platform simulation.
- Be familiar with deployment on Android and iOS.
- Be able to employ basic design principles including for access.
- Gain maturity as a programmer.
- Have fun.

Prerequisites: ITP 265 or other OOP programming
Co-requisites: none
Concurrent Enrollment: none

**Media** There is no required text (book) for this course.

**Homework programs** will be assigned and submitted on BrightSpace. Files for class and lecture slides will also be posted on BS. This is the place to look when you want to know what's next in class, what is due.

**Piazza** . We use this platform (preferred over email) to communicate, about the homework, office hours, questions from class. Generally questions should be public, so that others benefit from the answers. If you want to post code, make it 'private' (although our use of GitHub will probably obviate most of this).

**IDE and computer.** You need a fairly strong computer for class and to do your homework. (The ITP department has loaners if you don't have your own.) I will use VSCode, Android Studio, and git.

**Nature of the course.** I will introduce programming features daily, and then there will be lab to practice them. Homework programs will be due each week to exercise these techniques, typically Sunday night. There will be an exam at midterm on language syntax, and then a final project to put it all together.

**Grading** (note that the unweighted scores on BrightSpace may not match)

| what | % of grade |
|---|---|
| labs | 10 |
| homework | 40 |
| midterm exam (and quiz) | 20 |
| final project | 30 |
| total | 100 |

### Grade scale

| | | |
|---|---|---|
| A 93+ | | A- 90-92.99 |
| B+ 87-89.99 | B 83-86.99 | B- 80-82.99 |
| C+ 77-79.99 | C 73-76.99 | C- 70-72.99 |
| D+ 67-69.99 | D 63-66.99 | D- 60-62.99 |
| F 0-59.99 | | |

### Assignment Submission Policy

Lab assignments are supposed to be doable during the class session in which they are assigned. The due dates are usually a couple of days later, in case you need time to finish. Turn them in on BrightSpace. You generally get the points if you make a reasonable attempt. Homework will we try to do via git on GitHub, once that is set up, but you will submit on BrightSpace a notice for each homework that it is ready (done and pushed to GitHub). The notice should include at least a short comment about the work.

**Each homework assignment must be completed individually.** There are no group projects in this course. Do not present someone else's work as your own or let someone else present your work as their own. If you break this rule, I can give you an F and tell the Office of Academic Integrity, which ends up on your transcript. You do not want that. I admit that there is a fuzzy line here. You work together, remind each other about new syntax, help debug tricky snags. At your level I expect maturity, to know what is too much help or flat out copying. I have never had a problem in this course; let's keep it that way.

**Lateness** It is your responsibility to turn submit your work on time. My policy is to deduct up to 10% per day for lateness. Ask me if you need more time -- I can make allowances -- but generally you need to keep up in this course or you will get lost.

Some times students do not feel good about their work and they do not turn it in. Do not do that. Some times it is VERY late and they assume there's no point. Wrong. TURN STUFF IN. If you have anything, late or partial, show me. Do not sit on your work. If you have tried things that do not work, explain this in the comments.

**procedure for extension** To request an extension, submit to the homework page an explanation with any documentation. I will 'grade' it by answering whether you can have the extension, and then later when you submit the real thing, it is all in one place. If I say "ok" to a request verbally, do the submission to BrightSpace anyway, lest I forget.

**Additional Policies** If you have a documented reason for missing the exam or if you need special accommodations, please let me know ahead of time and we will make arrangements. Back up your work. A crashed computer is not a valid excuse for late work. If you think that something of your homework was not graded correctly, let me know and I will have another look.

The university allows students to join courses up to the end of week 3. You will need my permission to join the course, and I generally give it with the understanding that late-comers will catch up within a week of arrival and at the latest by the beginning of week 5.

# Schedule for ITP-368 Cross Platform Development

Links on this page only work on MY computer. - BEK

| week | Tue | what | Thu | what | Sun | homework |
|---|---|---|---|---|---|---|
| 1 | Jan 14 | 1. syllabus, Intro , set up IDE ,<br>hello.dart ,<br>lab: get Dart going , | Jan 16 | 2. more dart basics ,<br>class_stuff.dart ,<br>lab: Horse | Jan 19 | HW1: 3x+1 |
| 2 | Jan 21 | 3. Hello Flutter , 3x+1 solution , first program<br>lab: up and down | Jan 23 | 4. a few widgets , boggle.dart<br>lab: Coke Machine | Jan 26 | |
| 3 | Jan 28 | 5. **git** , DicePrep.dart ,<br>lab: Dice with git | Jan 30 | 6. dice1/2/3/4 (Yahtzee prep, more widgets)<br>lab : dice with git continued | Feb 2 | HW2: robot |
| 4 | Feb 4 | 7. BLoC providers , entry.dart ,<br>yahtzee.dart , yahtzeeBLoC.dart ,<br>lab: counter with BLoC | Feb 6 | 8. more state , robot2_bloc ,<br>lab: sized grid | Feb 9 | HW3: Lights Out |
| 5 | Feb 11 | 9. Files ,<br>lab: lister / groceries , | Feb 13 | 10. robot with file<br>lab: groceries or fix directory on R3 | Feb 16 | HW4: Converter |

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | Feb 18 | 11. Hydration , fd24 demo with state ,<br>lab: Hydrate Chess | Feb 20 | 12. food diary , code some?<br>lab: hydration and files (catch up) | Feb 23 | HW5: Quizzle |
| 7 | Feb 25 | 13. Navigation , Themes ,<br>lab: Splash123 | Feb 27 | 14. Schneiderman's 8 principles , Gestalt , What Users Do , Study Guide , use_keyboard.dart<br>lab: group design of Deal | Mar 2 | |
| 8 | Mar 4 | 15. test , | Mar 6 | 16. project - GO!<br>Project general, demos<br>lab: project ideas | Mar 9 | HW: HW Deal or No Deal |
| 9 | Mar 11 | 17. networks 1<br>lab: weather , | Mar 13 | 18. JSON ,<br>lab: API , | Mar 16 | break |
| spring break no class | | | | | | |
| 10 | Mar 25 | 19. synchronization theory ,<br>? a-sync communication<br>lab: chatter | Mar 27 | 20. dragging .<br>lab: chatter (continued?) | Mar 30 | project spec , |
| 11 | Apr 1 | 21. TTT raw shared<br>lab: tic tac toe upgrades | Apr 3 | 22. drawing, sizing, on screen keyboard, swiping<br>lab: project discussion, scrabble help | Apr 6 | HW8: Scrabble shared |
| 12 | Apr 8 | 23. sound ,<br>lab: sound board | Apr 10 | 24. animation<br>lab: fish shark life? | Apr 13 | |
| 13 | Apr 15 | 25. Localization ,<br>lab: show different languages | Apr 17 | 26. Deployment ,<br>platforms<br>lab: countdown | Apr 20 | |
| 14 | Apr 22 | 27. Scrabble review and project time<br>lab: project time | Apr 24 | 28. my programs FlashCards<br>lab: project time | Apr 27 | |
| 15 | Apr 29 | 29. project presentations<br>lab: projet time | May 1 | 30. student eval.s, project demos<br>lab: projet time | | |
| proj | May 14 Wed | 2pm projects due, present 2-4pm | | | | |

left over Design for Access ,