

CS 6456 Principles of User Interface Software

Semester and Academic Year: Spring 2027

Credits: 3 Credit Hours

Instructor Information

Jeff Wilson, PhD
Senior Research Scientist, Institute for People and Technology
jeff.wilson@gatech.edu

General Course Information

Course Description

Considers the architectural and algorithmic principles behind the implementation of interactive software systems and the tools that support them.

Course Learning Outcomes

By the end of the course, students should be able to:

- Explain the architectural and algorithmic principles behind the development of two-dimensional graphical user interfaces, including UI software organization, the Seeheim model, and model-view architectures.
- Describe how output and input are handled by graphics hardware, toolkits, and window systems, including damage management and layout.
- Implement interaction techniques for pen, touch, gesture, animation, and audio-based interaction.
- Develop interactive applications using a component-based GUI toolkit and development environment.

Required Course Materials

- No textbook is required; readings for each lecture topic are provided electronically through Canvas.
- Students need access to a Windows or macOS computer capable of running the required course programming tools. Homeworks use Python 3 and the Kivy GUI framework.
- Optional: User Interface Software and Technology, Amy Ko, Jake Wobbrock, and Eric Whitmire, University of Washington (online textbook).

Grading Policy

- Homeworks: 75% total
 - Five homework assignments: 15% each
- Quizzes: 25% total
 - Six quizzes; the best five are counted: 5% each
- Homework assignments include additional graduate-level requirements for CS 6456 students.
- Final letter grades use the standard scale (90% and above is an A, 80% to below 90% is a B, and so on).

Assignment details, due dates, and final grading logistics are maintained in Canvas.

Late Policy

A 24-hour grace period is allowed for late submission with no penalty. Submissions are not accepted beyond the grace period unless institutionally approved. Foreseen events, even if institutionally approved, must be accounted for by completing work early.

Additional Criteria for Successful Completion

- Complete module quizzes according to Canvas instructions.
- Complete individual programming assignments as individual work, including the graduate-level requirements.
- Keep assignment source code private during and after the semester.
- Submit work by the stated deadlines, subject to the course late policy and any approved accommodations.

Attendance and Participation

For on-campus sections, lecture attendance is recommended but not required. Asynchronous lecture materials are available.

Academic Integrity

Georgia Tech aims to cultivate a community based on trust, academic integrity, and honor. Students are expected to act according to the Georgia Tech Academic Honor Code. Suspected academic misconduct will be referred to the Office of Student Integrity. Students may discuss high-level ideas, but may not share code or detailed solutions unless assignment instructions explicitly allow it. Assignment source code must not be made publicly visible, including by forking starter projects into public repositories. AI-based assistance is treated like collaboration with another person. Assignment instructions may restrict AI use, and copied or uncited AI output is treated as plagiarism.

Student Conduct

Students and faculty are expected to maintain an atmosphere of mutual respect, acknowledgement, and responsibility consistent with Georgia Tech's Student-Faculty Expectations Agreement.

Accommodations for Students with Disabilities

Students with learning needs that require accommodation should contact the Office of Disability Services as soon as possible to discuss their needs and obtain an accommodations letter. Students should also contact the instructor so approved accommodations can be implemented.

Core IMPACTS

Not applicable.