

# **Welcome to CS 3250 Software Testing**

# Have You Ever Wondered



[Ref: emoji by Ekarin Apirakthanakorn]

# Why This Course

Testing software properly is a fundamental skill in modern software development

Software testing is a crucial activity to verify that programs fulfill requirements and to ensure software compliance

Software testing is about engineering tests that detect problems in software

Testing is so much more than writing a bunch of JUnit or Selenium tests

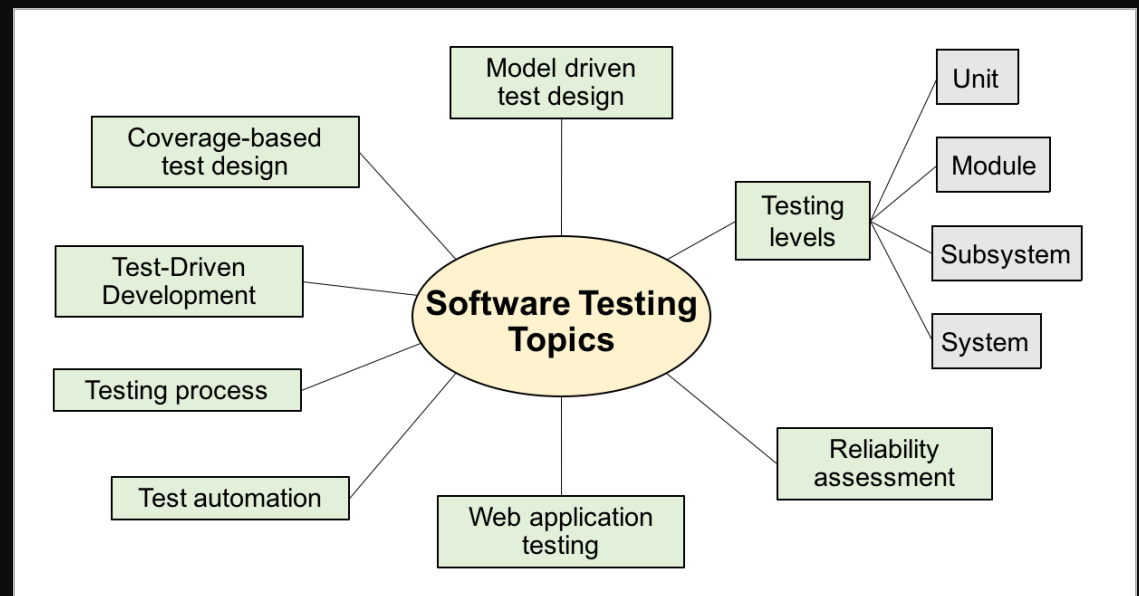
Software testing is about **designing good and effective tests** under time and budget constraints



[Ref: emoji by Ekarin Apirakthanakorn]

# What This Course is About

- Importance of Software Testing
- Testing in Practice: design, automation, execution, evaluation
- Test-Driven Development
- Test Coverage Criteria
  - Input space partitioning
  - Graph coverage
  - Logic coverage
  - Syntax coverage
- Web app testing
- Bypass testing



*“The true subject matter of the tester is not testing, but the design of test cases” —Jeff Offutt*

Instead of how testing **is done**, we focus on how it **should be done** and how it **will be done**

# Our Focus

- **Designing effective tests**
  - How test engineers should design tests
  - Explore a variety of test design concepts
  - Apply the theories to design abstract tests
  - Automate the abstract tests using JUnit and Selenium
  - Be exposed to challenges in test automation and explore best practices in test automation
- **Not training any specific test automation or software development frameworks**
  - Programmers/developers/engineers can explore & learn APIs
  - Provide the fundamentals to help you explore additional test automation frameworks beyond what we cover in this course

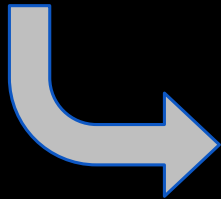
# I Hope to Help You

Test software adequately, appropriately, efficiently, and effectively

Be able to lead and work effectively in collaborative environments

Have confidence in learning and applying new testing approaches or frameworks to test any software

Get excited about exploring and developing new testing approaches to appropriately test any software



Become better testers

- Create high-quality tests at all testing levels
- Understand practical ways to design and automate tests
- Apply theory in practical ways

Become better programmers

- Be aware of potential problems in software and able to create high-quality developer tests

Become better engineers

- Build programs and test them in a unified manner

Become better thinkers

- Solve problems in logical and analytical ways

# You Will be Able to Answer

- What is software testing?
- Why do we test software?
- When should we test software?
- Who should test software?
- How should we test software? Why should we do it that way?
- When should we stop testing? Good enough?
- How many tests do we need to write?
- How do we choose test inputs effectively?
- How can we measure the quality of a test suite?
- How can we improve the quality of a test suite?
- How can we reduce the testing cost?

# What You will Do

## Quizzes:

- 5 quizzes
- First 15 minutes of class
- Close book/note
- In class
- Reflection & correction

## Practice-of-the-days:

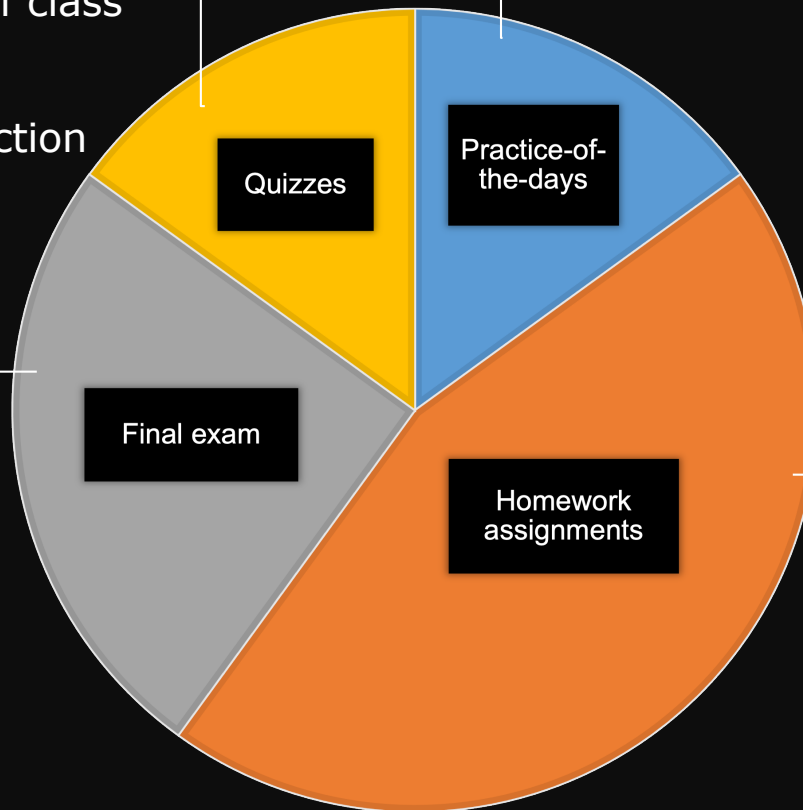
- Weekly
- Grade on done/not done
- Due 1pm EST the next day
- -25% for 24 hours late;  
-50% for 48 hours late
- Not accepted after 48 hours past due

## Final exam:

- Comprehensive
- Close book/note
- In class

## Hands-on activities:

- Almost every meeting
- No submission



## Homework assignments:

- Almost weekly
- Due before class
- Early-submission bonus points
- -25% for 24 hours late;  
-50% for 48 hours late
- Not accepted after 48 hours past due



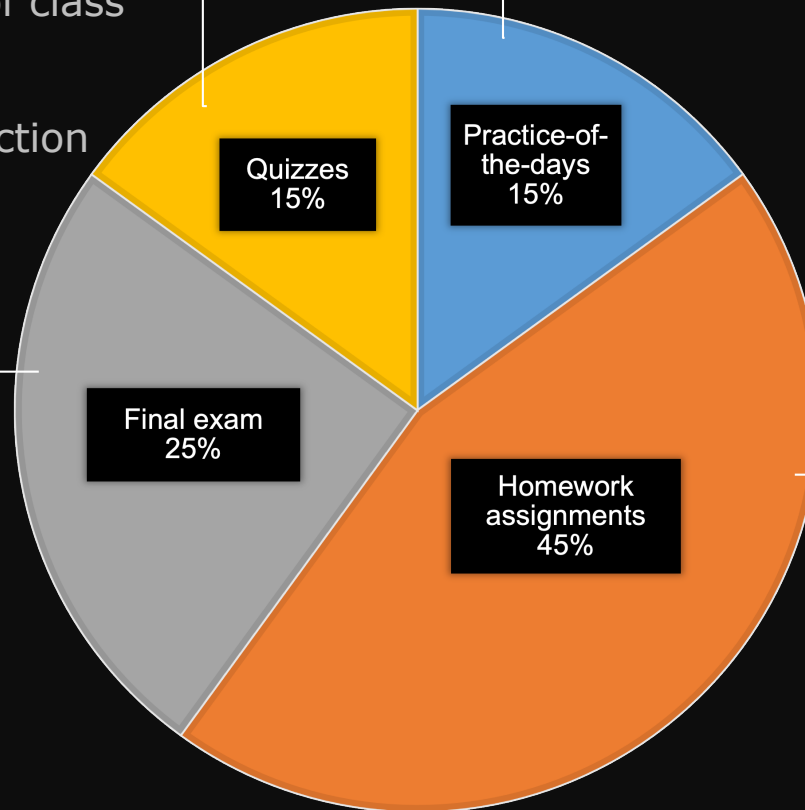
# How Your Learning will be Evaluated

## Quizzes:

- 5 quizzes
- First 15 minutes of class
- Close book/note
- In class
- Reflection & correction

## Practice-of-the-days:

- Weekly
- Grade on done/not done
- Due 1pm EST the next day
- -25% for 24 hours late;  
-50% for 48 hours late
- Not accepted after 48 hours past due



## Final exam:

- Comprehensive
- Close book/note
- In class

## Homework assignments:

- Almost weekly
- Due before class
- Early-submission bonus points
- -25% for 24 hours late;  
-50% for 48 hours late
- Not accepted after 48 hours past due

## Hands-on activities:

- Almost every meeting
- No submission

# Prerequisites

- CS 2150 or (CS 2100 & CS 2120) with C- or better (or COVID CR)
- Discrete math, data representation, (some) software engineering
- Java, syntax and semantics of multiple programming languages
- Programming skills
- Software installation and troubleshooting skills
- **Willingness** and **excitement** to learn new concepts and experience new frameworks

*Prerequisites define what you need to know **before** taking a class to succeed in the class. Please do not ask if you can take the class without the prerequisites, I have to advise against it*

*The CS student services staff will monitor and remove students not meeting prerequisites in courses in which they are enrolled*

# More Info to Help You Learn

- **Class URL:** <http://www.cs.virginia.edu/~up3f/cs3250/>
- **Readings:** Check schedule page, please read before class
- **Textbook:**



(book website) <https://cs.gmu.edu/~offutt/softwaretest/>

(student solution)

<https://cs.gmu.edu/~offutt/softwaretest/exer-student.pdf>

- **Emails:** Use UVA email, check it regularly, include **"CS 3250"**
- **No eating** in class

# Share Ideas, Get Help, Just Talk

- Instructor office hours: days/times on class website
- TA office hours: days/times on class website
- Piazza: <https://piazza.com/virginia/spring2024/cs3250s24>
  - You should have gotten an invitation today
  - If not, check your UVA email or check with me
- Questions should be posted to an appropriate thread
  - Answered by instructor, TAs, and your peers
  - Public: general questions and answers
  - Private: Grade/homework-specific questions – instructor and TAs
- Do not email general homework or class questions, use Piazza

# Learn with "Purpose" and "Vision"

Help yourself learn: <https://forms.office.com/r/uwbbLJEGQT>

- To tailor our class to better support and emphasize your differences, energy, and passions, as well as to enhance your strengths
- You are encouraged to be involved in designing and shaping this course to maximize your learning experience, skills, and knowledge
- Please feel free to share your ideas / opinions / suggestions on things that we should do or try to help you learn software testing, using the "Help yourself learn" form

# Accommodations

<https://www.cs.virginia.edu/~up3f/cs3250/policies.html#no-extension>

## Accommodations: Religious

If you wish to request academic accommodation for religious observance, please let me know via email as far in advance as possible.

If there exists a religious conflict, please mention it.

Do not say *"You are busy"* or *"You have a busy start to the week."* *"Being busy / having a busy schedule / having a busy start to the week"* is **not** a sufficient justification for an accommodation / extension.

*"Being busy / having a busy schedule / having a busy start to the week"* does not imply a religious conflict.

If you need religious accommodations, please say that you need religious accommodations.

## Accommodations: Schedule conflict

If there exists a schedule conflict, please provide the necessary information to determine an accommodation.

Do not say *"You are busy"* or *"You have an unavoidable conflict."* *"Being busy / having a busy schedule / having an unavoidable conflict"* is **not** a sufficient justification for an accommodation / extension.

It is important to note that being busy with other courses (homework assignments, projects, exams, quizzes, tests, ...) is **not** a sufficient justification for an accommodation / extension. Please consider adjusting or planning your time management.

## Accommodations: Things in life and special circumstances

If you believe that circumstances warrant an extension or exception, come talk to the course instructor and we will resolve the situation as best we can.

# Additional Things to Know

- **Regrade request:** Submit to Gradescope
  - Regrade the “entire” work (exam, quiz, assignment, showcase, POTD)
  - Correct any grading mistakes
  - May result in either an increase or a decrease in grade
  - Not try to scare off students whose work was graded incorrectly, but try to avoid frivolous requests
- Check schedule page regularly
  - In case of emergency, email & post update on the schedule page

# "You" are the Main Driver!!

"You" are the main driver of your learning success.

Your engagement and participation is the most important aspect of your learning experience.

You are encouraged to do all the activities, try all concepts, experience all frameworks ... and do beyond the showcase's minimum requirements!!