

WELCOME TO 18-654

Software Verification and Testing

Fall 2023



<https://canvas.cmu.edu/courses/36564>

Instructor: Hakan Erdogmus

- ECE Teaching Professor, MS-SE Program Lead
- Email: hakane@andrew.cmu.edu
- Office hours:
 - Via Zoom, see the Syllabus page on Canvas
 - Physical office located in Pittsburgh

Teaching Assistants



Varun Agarwal
varunaga@andrew.cmu.edu



Divyam Shah
dbshah@andrew.cmu.edu



Anni Yuan
anniy@andrew.cmu.edu

Office Hours: See the Syllabus page on Canvas

Schedule - Weekly Format

Tuesday 1:00PM - 2:50PM PT @ Rm 118

Thursday 1:00PM - 2:50PM PT @ Rm 118

- *For course plan & calendar, including holidays and breaks, see the Syllabus page on Canvas!*
- *May make adjustments as necessary.*

Coming Up

This week

- **Thursday:** Lab 0 - Unit Testing Exercises with Vocareum
 - *Set up Eclipse, create a Java project (> 1.8) with JUnit 4.12 (not JUnit 5)*
 - *add hamcrest-all-1.3.jar (or higher) to build path before JUnit, and make sure they work*

Next Week

- **Tuesday:** Assignment 0 - Unit Testing released

Date (Week)	Day	Lecture Content	Slide Decks / Lecture Notes / Assignment/Lab (Topic) / Due Dates / Notes
August			
08/29 (1)	Tue	<ul style="list-style-type: none">• Welcome to 18654• SVT Landscape	<ul style="list-style-type: none">• CourseInfo (42)• SVTIntro (17)
08/31 (1)	Thu	<ul style="list-style-type: none">• Testing Basics and Approaches; Unit Testing Basics• L0: Simple Unit Testing Exercises and JUnit (Stack) (G) - <i>Explained/started in class, completed at home</i>	<ul style="list-style-type: none">• Testing Basics (44)• UnitTestingBasics (20)• Release L0 (Unit Testing Warmup): <u>Prep Required</u>
September			
09/05 (2)	Tue	<ul style="list-style-type: none">• (Unit) Testing Principles• Advanced Unit Testing: Designing for Testability	<ul style="list-style-type: none">• UnitTestingPrinciples (53)• DesignForTest (20/82)• L0 due• <u>Release A0 (Unit Testing Exercises)</u>
09/07 (2)	Thu	<ul style="list-style-type: none">• Advanced Unit Testing: Designing for Testability (cont'd)	<ul style="list-style-type: none">• DesignForTest (62/82)

Learning Objectives

- Appreciate diversity of SVT approaches and techniques
- Appreciate costs of benefits of different SVT techniques
- Create and analyze software models
- Learn and apply core SVT techniques...
 - ... covering both practice and theory
 - ... covering common and niche techniques
 - ... using a variety of tools
- Select and combine core SVT techniques as appropriate

Acquire applied knowledge of basic testing and formal verification approaches using representative tools.

Class Format and Etiquette

- In-person for students:
 - Broadcast to this room in SV - *Zoom link is for instructor and remote TAs, not available to students*
- Arrive on time:
 - we won't wait for everyone to start labs and class exercises
- No multi-tasking please
 - Do not check your email, surf, social network
- The periods are long!
 - Feel free to stand up, stretch your legs, walk around any time
- Engage
 - You may leave the room if you are bored: attendance not taken (but you need to be present for graded in-class components)

Are you ready to take this course?

- Have you taken 18652 or are taking it now?
 - 18652 - *can be taken concurrently, but having taken it before will help*
 - Conversely, skills learned in Part 1 of this course will help with 18652 project
 - Some overlap with the testing module of 18652
- Do you know enough Java?
 - If not, you may struggle: consider dropping

Your Experience

What is your experience with SVT?

What are your expectations from this course?

Raise your hand or just unmute and speak!

My Expectations

- Interest in both theory and practice of SVT
- Installing/trying tools and coming to class ready
- Be comfortable with some math
- Emphasis on analytic thinking, design thinking, modeling
- Submitting clean, professional deliverables
- Active participation

Course Materials

... are available on Canvas at

<https://canvas.cmu.edu/courses/36564>

... will be updated/posted just-in-time

After each lecture, always download updated lecture slides

Submission of Reports/Assignments



Canvas



Technical questions about course topics by students



accessible via Canvas

- For non-administrative questions
- Only for course content, subject matter, concepts, technical questions
 - Please avoid posting anonymously (note that poster's identity is always visible to instructors)

Activate now from Canvas (uses your andrew credentials)

Announcements by Teaching Team



- Info on assessments, lectures, course schedule, etc.
- Logistics, deadlines, extensions

Activate now from Canvas (uses your andrew credentials)

Private, administrative, logistical, schedule (deadlines, extensions) and grade-related questions

By Email: Use your andrew email to contact both

hakane@andrew.cmu.edu

dbshah@andrew.cmu.edu

Do not use Piazza for these!
Do not use your personal email!

It's a tool-rich course!



JUnit

EclEmma



checkstyle



PROMELA



SpotBugs





For assignments/labs:

Use a dependency management tool
with your IDE/Editor

e.g., **Maven**[™]

I use Eclipse and provide examples/starter code and
setup instructions for Eclipse/Maven



Optional:

- Version control (git) your homework code
- Push your code to a private github repo in the cmusv-svvt organization

Course Organization

Part 1: Testing

Part 2: Static Analysis & Model Checking

Progression

Practical



Theoretical

Ad-hoc



Systematic

Widely
Applicable



Specialized

Topics -- Part 1

- Testing Basics and Approaches - *Review*
- Unit Testing - Principles and Application
- Systematic Testing
 - Spec-Based Testing
 - Structural Testing
 - Test Adequacy

Lectures, Labs, Project, Assignments, Midterm

Topics -- Part 2

- Static Analysis
- Model Checking for Finite-State Verification of Concurrent Systems
- Applying Model Checking using Promela/SPIN

Lectures, Labs, Assignment, Final

Final is only on Part 2

Technologies/Tools

Part 1: Java-based

- JUnit 4
- Mockito
- Fit/FitNesse

Part 2: Tools & Special-purpose

- Static Analysis Tools (CheckStyle, PMD, SpotBugs)
- Promela/Spin

Install and test tools before class activity, lab, assignment

- *Tool installation may take time & troubleshooting*

Grading Rubric

15%	Project
15%	Labs
30%	Assignments
20%	Midterm Exam
20%	Final Exam
5%	Extra

Deductions for failure to complete a few required chores

Project (15%)

- ~4 weeks of work
- Team: Max 5 students
- Team formation:
 - Self-select teams
- Deliverables submitted to Canvas/Vocareum
- Released **September 26**
- Due **November 9** Presentation date



Assignments (30%)

- Take-home
- Individual or Partnered
- Normally due in one week
- Submitted to Vocareum
- Grade is automated or partially automated

Labs (15%)

- Requires preparation (essential)
 - *set-up and study material*
 - *no help with set-up during lab*
 - *skip prep at your own expense*
- Performed or started in class
- Individual or partnered
- Some requires short lab report
- Requires attendance (must attend corresponding class session to receive credit)
- Deliverables usually due end of day
- Submitted to Vocareum or Canvas



Exams (x2: 20 + 20 = 40%)

- Midterm: **Thursday, October 12**
- Final: **Exam Week - TBD - scheduled by the Registrar**
- Closed book
- Online on Canvas using Lockdown Browser
- Mix of short-answer/essay, problem solving, coding, and multiple-choice questions
- Mix of quantitative and qualitative (interpretational) questions
- Cover all elements (lectures, graded and ungraded labs/assignments, project work, mandatory readings/videos)

Extras (5%) - extra credit

For attending and completing miscellaneous activities and take-home/class exercises

- Mockito Exercises and Prep
- Promela/Spin Prep
- Promela/Spin Exercises
- *Additional bonus points may be granted solely at instructor's discretion at the end of the semester for exceptional contribution, participation, or performance*

Working in a team and Contributing to a team project takes effort and planning

- Agree on rules (example Team Charter available on Canvas under Course Info)
 - Agree on a plan
 - Start right away (only 4-weeks for the project)
 - Meet regularly
 - Pull your weight
 - Hold each other accountable
-
- Project grades maybe adjusted by 360-degree peer evaluations or contribution reports

Repo for Projects

- You may create private repos for your team in course github organization



cmusv-svvt

- *Will ask for your github account to add you to this organization*
- *Need your full real name and recognizable headshot photo added to your github profile (otherwise you won't be added)*



Late Submissions

- Subject to penalties 10% per day, until grades/solution are published up to a maximum of 4 days (0% afterward)
- An extension with no penalty may be considered in exceptional cases at instructor's discretion
- See policies on the Syllabus page

Lab Reports can be...

- Light-weight
- Brief
- Simple

but must be...

- Professional like an official work memo
 - Format: prefer bullet points, tables
 - Formatting: keep it simple, plain text is fine
 - Language: simple, but well-written; short sentences check for grammar and spelling
 - Images: embedded in report (no separate image files)
 - No abrasive, disrespectful, angry, colloquial language

No broken/colloquial English please! Feel free to use ChatGPT to polish your report!
Generative AI use policy on the Syllabus page!

Lab Reports - Do's and Don'ts

Answer reflection and lessons-learned questions insightfully, focusing on the learning objectives of the lab...

Bad

- “I like it.”
- “It was too difficult.”
- “It was good experience.”
- “Technique XYZ was interesting.”
- “I completed the lab, I finished all the tasks, I applied technique X as required.”

Appropriate only in last section (comments/suggestions/improvements) - be specific so I can correct

- “My Java knowledge is poor, so I’ll study it well next time.”
- “Time management was a problem.”
- “The starter code had a bug.”
- “It would have been better to rename method X to method Y.”
- “Tasks were confusing.”
- “Instructions were inconsistent.”
- “I couldn’t get coverage working in my IDE. It took me long time to figure out the dependencies.”

Good

- “The Acme tool proved difficult to use. Setup was cumbersome and some results were inconsistent. For example, ...”
- “The super-graph coverage technique was not a good fit for this testing problem because the input domain was too restricted.”
- “Extracting test cases using the systematic method was easy, however the tests were challenging to automate. When run manually, they took over 30 minutes. This highlights the importance of automated tools: without them the method would not be is feasible for repeated execution.”

Readings/Videos (no books to buy)

- **Mandatory**

- Will help with labs, assignments, project, exams
- Covered in exams
- *Indicated on Canvas with (M)*

Will have questions on them on the exam

- **Optional**

- Extra resources or reference material
- For students with specific interests in a topic
- Will help with labs, assignments, project, exams

Optional Reference Text Books - Check Syllabus Page

- Young, M. and Pezzé, M. (2008). *Software Testing and Analysis: Process, Principles, and Techniques*. Wiley. ISBN-13: 978-0471455936
- Koskela, L. (2013). *Effective Unit Testing: A Guide for Java Developers*. Shelter Island, NY: Manning. ISBN-13: 978-1935182573
- Holzmann, G. (2004). *The SPIN Model Checker: Primer and Reference Manual*. Boston, MA: Addison-Wesley. ISBN: 0-321-77371-3
- Ammann, P. and Offutt, J. (2013). *Introduction to Software Testing*. Cambridge, New York, NY: Cambridge University Press. ISBN: 978-0-521-88038-1

It's your responsibility to know and understand all Course Policies, including Grading Policies, Generative AI and Online Sources Use Policy, and Academic Integrity Policies

IMPORTANT

- Make sure to review the policies on the Syllabus page
- Zero tolerance for academic integrity violations
- Don't be caught off-guard
- Ignorance of policies is no excuse



More on Academic Integrity

Be careful about these sanctionable offenses:

- resubmitting/updating assignment/lab deliverables after the deadline or grace period
- updating exam deliverables after the exam is over

even if they are accepted by the submission tools used, which may have to remain open after deadlines for grading and testing purposes.

The above behaviors will be considered cheating!



Chores and Deductions

Your profiles on Canvas, SIO, and the GitHub account used for the course requires photos and full real names:

- No photo on your Canvas Profile by the end of the 2nd week of classes: -.5 deduction on your total grade
 - No photo on SIO by the end of the 2nd week of classes: -0.5 deduction on your total grade
 - Photo and full real name missing in your GitHub account's profile: will be refused access to the course's GitHub organization for your project and assignments
-

Grading Schema

Remember: you have a 5% cushion from the Extras component

A	100% to 95%	
A-	<95% to 90%	
B+	<90% to 85%	Safe for you 97-unit Program QPA
B	<85% to 80%	
B-	<80% to 75%	
C+	<75% to 70%	Unsafe for you 97-unit Program QPA
C	<70% to 65%	
C-	<65% to 60%	Degree requirements cannot be satisfied - effectively Fail
R	<60% to 0%	



Questions
