

Introduction to Software Testing

Final Exam Preview

Software Testing & Maintenance

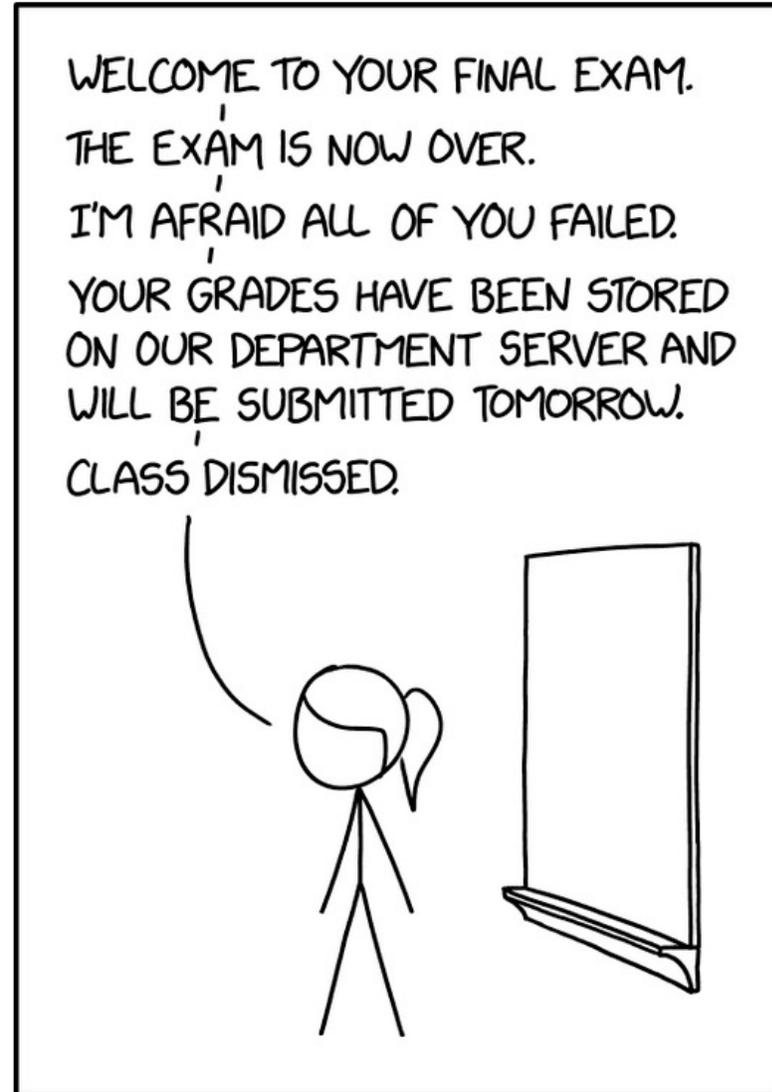
SWE 437/637

go.gmu.edu/SoftwareTestingFall24

Dr. Brittany Johnson-Matthews

(Dr. B for short)

"At Least it's not the algorithms exam!"



Exam Schedule

Final exam date:

Thursday December 12 @ 4:30 pm (regular class day, time, and location)

Talk to me about alternatives if you have other exams scheduled that day

The exam will be **on paper** and will have a **time limit** (3 hr class time)

You will be allowed a **1-page cheat sheet** (no larger than 8.5x11)

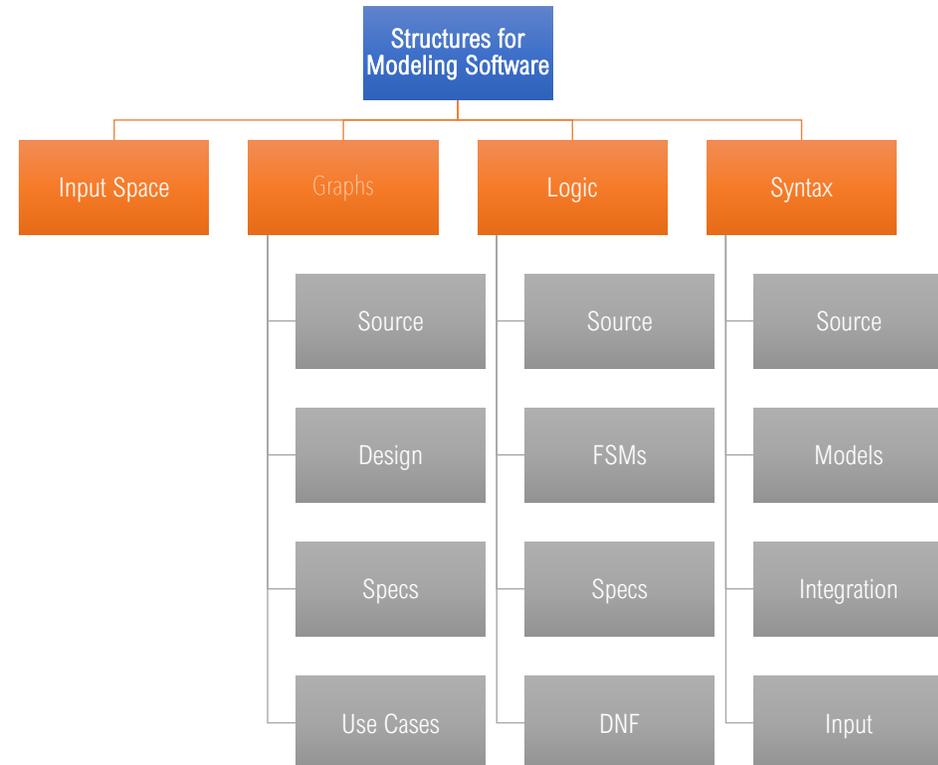
By University policy, there is no makeup for unexcused absences

Be sure to coordinate with me *ahead of time*

Exam Structure

The final exam will have **5 sections**, each worth 20 points:

1. Short answers
2. Input Space Coverage
3. Graph Coverage
4. Logic Coverage
5. Syntax Coverage
6. Bonus



Section 1 – short answers

Expect **10 questions** (2 points each) taken randomly from the following categories:

Agile testing and basic definitions, such as fault/error/failure and the RIPR model

Input space coverage

Graph coverage

Logic coverage

Syntax coverage

Questions will be a combination of:

Multiple choice

True/false

Fill in the blank

Section 2 – input space coverage

This section will have one (possibly multi-part) problem worth 20 points

Topics may include:

Developing sets of characteristics

Identifying complete, disjoint partitions

Identifying test requirements for base-choice and pairwise criteria

Identifying and replacing infeasible test requirements

Section 3 – graph coverage

This section will have one (possibly multi-part) problem worth 20 points =

Topics may include:

Drawing/identifying a graph from a set of nodes or from source code

Finding test requirements to satisfy node, edge, edge-pair, prime path, DU-pair/path coverage

Identifying prime paths and test paths

WILL NOT include 7.4-7.6

Section 4 – logic coverage

This section will have one (possibly multi-part) problem worth 20 points

Topics may include:

Building truth tables

Identifying when a clause determines the predicate using truth tables and/or the XOR method

Building K-maps and using them to identify prime implicants

Identifying test requirements to satisfy GACC, CACC, and/or RACC coverage

WILL NOT include ICC, MUTP, CUTPNFP or MUMCUT problems (though these topics may appear in Section 1 questions)

Section 5 – syntax coverage

This section will have one (possibly multi-part) problem worth 20 points

Topics may include:

Using mutation operators to create mutants

Strong vs. weak mutation

Identifying killable, equivalent, and trivial mutants (not dominator or productive mutants)

Writing test to kill mutants

*Will include only source code mutation testing, **WILL NOT** include BNF or other chapter 9 topics*

Test strategies

By this point in your academic journey, you likely have your own test-taking strategies.

A few suggestions:

- Skip questions that you don't immediately know the answer to

- Try not to reference notes or resources for every question

- Only work on the bonus once you've completed the exam questions

If you stay ready, you don't have to get ready!