

INTRO TO SOFTWARE TESTING

FINAL EXAM PREVIEW

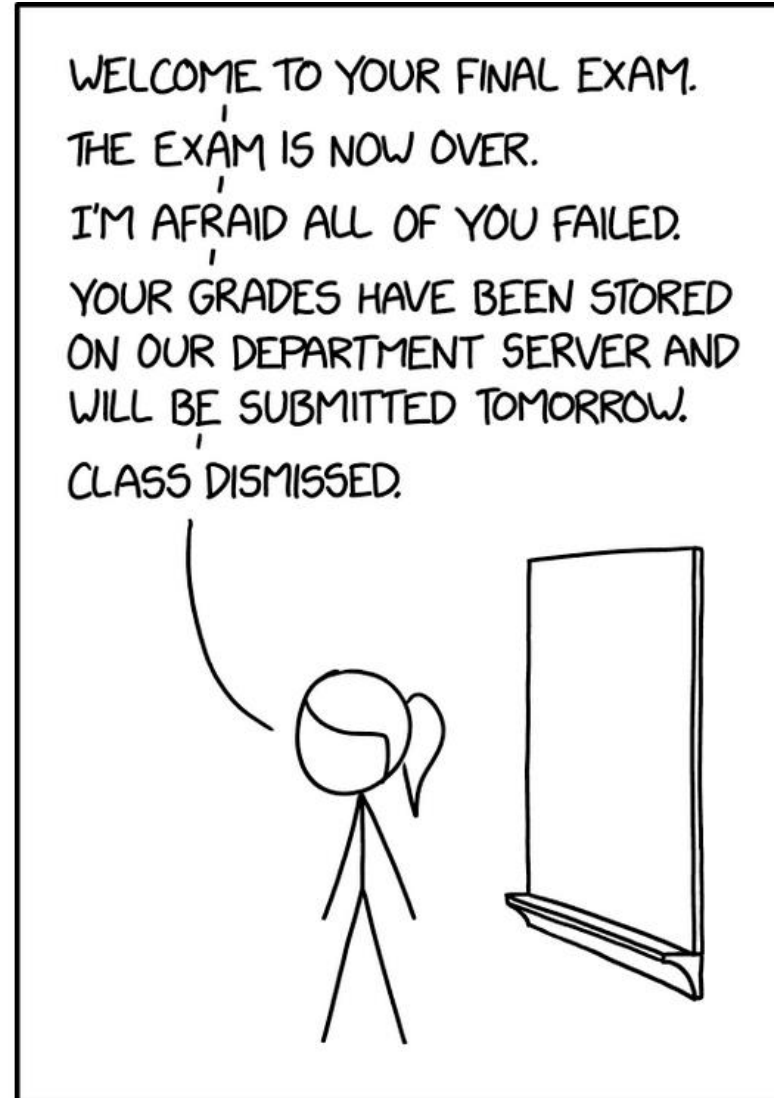
Dr. Brittany Johnson-Matthews

(Dr. B for short)

<https://go.gmu.edu/SWE637>

Adapted from slides by Jeff Offutt and Bob Kurtz

"AT LEAST IT'S NOT THE CYBERSECURITY EXAM!"



EXAM SCHEDULE

Final exam date:

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

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

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

Open everything, except classmates

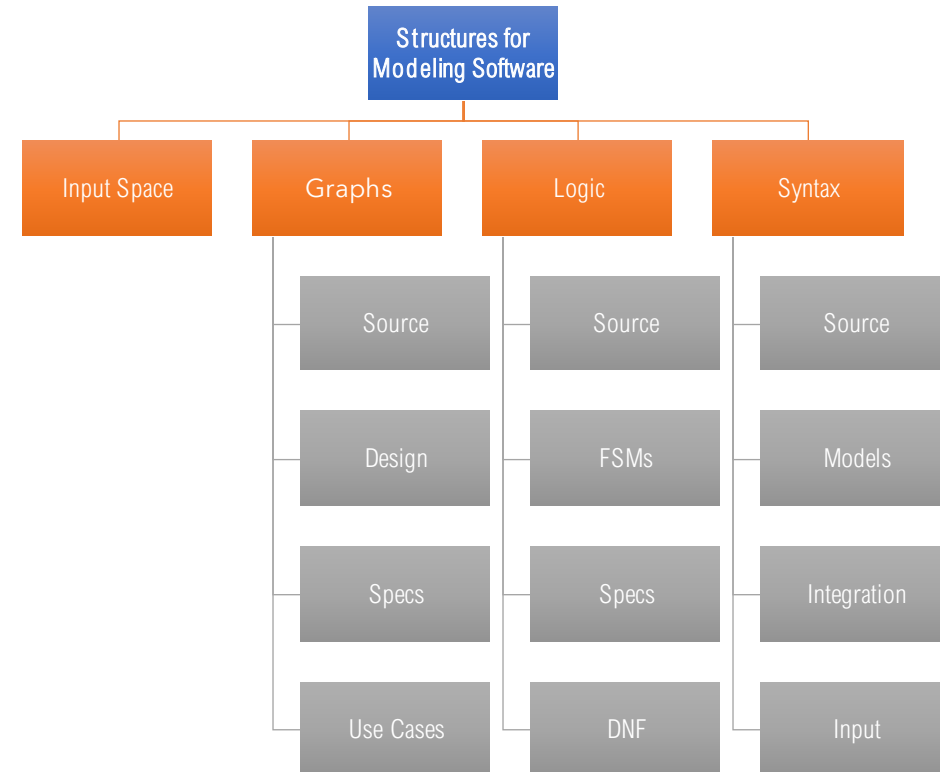
By University policy, unexcused absence from exam

Be sure to coordinate with me *ahead of time*

FINAL 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, worth 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 in total

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 in total

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 in total

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 in total

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

SECTION 6 – BONUS

This section will have one (possibly multi-part) problem worth 5 points

Topics may include:

- UTPs, NFPs, ICC, or MUTP

- Mutant subsumption or other advanced mutation testing topics

- Compare/contrast coverage criteria from different categories

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!