LATE

minutes remaining

Hide Time

Manual Save

Navigation

General Reminders

- You **may** use *free* generative AI tools on this exam.
- The exam is open note, open lecture recordings, open course webpage, open computer (including Visual Studio, Python, etc.), and open Internet in general. You may use general webpages (e.g., Stack Overflow, Wikipedia, etc.).
- You may **not** use live help from another human.
- If you leave a sub-question entirely **blank** then you will receive one third of its points rounded down. For example, if a sub-question is marked "(4 points)", you would receive 1 point if you left it entirely blank.
- If you are asked to support or refute a position, indicate which option you selected. One word can suffice.
- If you are asked to provide a quote from multiple possible sources (e.g., either the lecture slides or a reading), indicate which source you selected.
- If you are asked to provide a quote to help support or justify your arugment, your quote must be directly relevant to your argument.

Question 1. Word Bank Matching (1 point each, 18 points)

For each statement below, input the letter of the term that is *best* (e.g., most specifically) described. Note that you can click each word (cell) to mark it off. Each word is used at most once.

A. — A/B Testing	B. — Agile development	C. — Alpha Testing	D. — Beta Testing
E. — Competent Programmer Hypothesis	F. — Constructive Cost Model	G. — Dynamic Analysis	H. — Formal Code Inspection
I. — Fuzz Testing	J. — Integration Testing	K. — McNamara Fallacy	L. — Milestone
M. — Mocking	N. — Oracle	O. — Pair Programming	P. — Pass Around Code Review
Q. — Priority	R. — Race Condition	S. — Regression testing	T. — Risk
U. — Spiral Development	V. — Streetlight Effect	W. — Triage	X. — Uncertainty
Y. — Unit Testing	Z. — Waterfall Model		

Q1.1:

Yushu reports an issue that must be addressed before any other work is done.

Q1.2:

MunchyRoll corporation has a policy that for every created function, there must be corresponding inputs, outputs and oracles to assess that function.

Q1.3:

Wes is a project manager for Veecsa. Wes wants to make use of historical company data to predict how long the next development activity will take.

Q1.4:

Developers at Bank of Michigan deliver an increasingly-complete series of prototypes, while accounting for risk.

Q1.5:

Leena decides to employ a software development approach usually takes longer but results in fewer defects.

LATE

minutes remaining

Hide Time

Manual Save

Navigation

- Question 1
- Question 2
- Question 3
- Question 4
- Question 5
- Extra Credit
- Pledge & Submit

Q1.6:

Software development processes can mitigate, but not remove, *this*.

Q1.7:

In Peter's banking application, if a withdrawal and a deposit are made at almost the same time, neither transaction completes.

Q1.8:

Developers at Miscord find that modules that work well separately do not always work well when put together.

Q1.9:

Gwen is a project manager for Boogle. Gwen wants to make use of historical company data, but does not know if those past numbers exactly match current realities.

Q1.10:

This dynamic analysis creates random test inputs and can be useful even when oracles are not available.

Q1.11:

Developers at GlazeBook complete an internal study and find that most defects are caused by simple typos.

Q1.12:

Devforce corporation requires that at least one developer familiar with the language and at least one developer familiar with the module approve all changes to that module's code.

Q1.13:

Managers at 481andMe are focusing exclusively how many users they have for a newly-deployed game. They believe that having more users means they are doing better than their competitors and that their game will win quality awards and turn a profit.

Q1.14:

This software development process allocates resources to issues based on their impacts and their urgencies.

Q1.15:

Aamir is tasked with determining what the correct chart produced by a jFreeChart test *should* look like.

Q1.16:

This software development process mitigates certain biases by having different people write the code, present the code, and analyze the code.

Q1.17:

Yuxuan is developing a database backend. To allow others to test that interface before it is finished, a simple implementation that stores data in text files is created.

Q1.18:

Momo is familiar with the OS abstraction layer but has recently struggled with a low-priority issue where text is displaying in red instead of green. It was easy to examine the system calls responsible for rendering text, but Momo was still unable to fix the bug.

LATE

minutes remaining

Hide Time

Manual Save

Navigation

- Question 1
- Question 2
- Question 3
- Question 4
- Question 5
- Extra Credit
- Pledge & Submit

Question 2. Testing (25 points)

Answer the following questions about testing.

(a) (5 points)

In Tillmann and de Halleux's *Pex — White Box Test Generation for .NET*, why do the authors introduce the notion of "reachable statements"? Support your answer with a quote from the paper. (After copying the quote, use at most four sentences.)

Your answer here.

(b) (5 points) What are the two specific relationships between code review and testing described in Sections 2.0-2.5 of *Software Engineering at Google* by Henderson? Support your answer with a quote from the paper. (After copying the quote, use at most four sentences.)

Your answer here.

(c) (5 points) Support or refute the position that the particular use of Randoop in the context of our Homework #2 assignment is *black box testing* (i.e., does not require peering into jfreechart's internal structure). Your answer must include a relevant quote from the HW2 webpage, Randoop documentation, or another class reading (name the reading when providing the quote). (After copying the quote, use at most four sentences.)

Your answer here.

(d) (5 points) Consider the Oracle-Comparator Model of testing. Consider the use of AFL in Homework #2 and the use of Randoop in Homework #2. For each of those two activities, indicate whether it involves explicit oracles, implicit oracles, or no oracles. For each activity, provide a sentence elaborating or explaining your answer. Use at most four sentences total.

Your answer here.

(e) (5 points) Describe a scenario in which a test suite that achieves 100% statement coverage might miss a bug in a program. Then describe what other approach (testing, coverage, analysis, etc.) could find that bug. Include a quote from the slides or readings (your choice, but indicate which one) to support your argument. (After copying the quote, use at most four sentences.)

Your answer here.

Question 3. Mutation and Measurement (20 points)

Answer the following questions about mutation analysis and measurement.

LATE

minutes remaining

Hide Time

Manual Save

Navigation

- Question 1
- Question 2
- Question 3
- Question 4
- Question 5
- Extra Credit
- Pledge & Submit

(a) (5 points) In the `floorSqrt` example near the end of https://eecs481.org/lectures/se-06-inputsoracles.pdf , what is a first-order mutation that falsifies the invariant `x >= retval*retval` with respect to the four tests mentioned there? Indicate your mutation (e.g., "change xyz to abc"). Indicate one of those four test inputs that falsifies the invariant for your mutant. Indicate what value the original returns and what value your mutant returns.

Your answer here.

(b) (5 points) Suppose that a group of researchers find very convincing evidence *against* the competent programmer hypothesis but also find very convincing evidence *for* the coupling effect hypothesis. What would this mean for mutation testing? Include a direct quote from *An Analysis and Survey of the Development of Mutation Testing* (part of the HW3 spec) to support your argument. (After copying the quote, use at most four sentences.)

Your answer here.

(c) (5 points) Consider Buse and Zimmermann's *Information Needs for Software Development Analytics*. First, according to that report, which group makes the most use of code coverage information: developers, managers, customers or students? (Use at most one sentence.) Second, consider the report's section on "Targeting Testing" and its discussion of re-targeting testing. Support or refute the claim that mutation testing would be a good fit for re-targeting based on code churn. Include a quote from the reading to justify your argument. (After copying the quote, use at most four sentences.)

Your answer here.

(d) (5 points)

Consider Anda et al.'s *Variability and Reproducibility in Software Engineering: A Study of Four Companies that Developed the Same System*, as described in the second half of https://eecs481.org/lectures/se-02-risk.pdf. First, among the 35 companies that responded with bids, what relationship was found between the price a company would charge and that company's emphasis on analysis and design (development process). (Use at most one sentence.) Second, support or refute the claim that the results from that study argue *against* the use of constructive cost modeling (cocomo). Include a quote from the slides or reading to justify your answer. (After copying the quote, use at most four sentences.)

(Excerpts of this paper were described in the lecture and the slides, but the full paper was also an optional reading. You can answer this question using only the information from lecture, but you are also welcome to use the full text.)

Your answer here.

Question 4. Dynamic Analyses and Human Processes (24 points)

Answer the following questions about dynamic analyses or human processes in software development.

You are instrumenting the program below for a dynamic analysis. You want to be able to analyze the instrumented log later to determine the set of values that were passed to the `external_api()` function. At each possible instrumentation point you can either add instrumentation code to log (i.e., print) the value of a **single variable** or you can add nothing.

The next questions ask you to add the minimum amount of logging possible so that the values that were passed to `external_api()` can still be determined from the log. (We want the minimum amount to avoid slowing the program down too much).

LATE

minutes remaining

Hide Time

Manual Save

Navigation

- Question 1
- Question 2
- Question 3
- Question 4
- Question 5
- Extra Credit
- Pledge & Submit

For each possible instrumentation point below, *either* indicate that you must log a variable's value there and indicate which one *or* explain why you need not log any variable's value at that point.

```python
import random

def ant(a):
  return 0

def bat(b):
  return b + random.randint(1,4)

def michigan(a,b,c):
  # possible instrumentation point 1
  external_api(a)

  # possible instrumentation point 2
  d = ant(a)
  external_api(d)

  if (a < 10):
    a = bat(b)
    # possible instrumentation point 3
    external_api(a)

  elif (a == b):
    # possible instrumentation point 4
    external_api(b)

  a = 7
  # possible instrumentation point 5
  external_api(c)

  # possible instrumentation point 6
  external_api(a+b)

def main():
  x = random.randint(1,100)
  y = random.randint(1,100)
  z = random.randint(1,100)
  michigan(x,y,z)

main()

```

(a) (1 points) At possible instrumentation point **1**

Your answer here.

(b) (1 points) At possible instrumentation point **2**

Your answer here.

(c) (2 points) At possible instrumentation point **3**

Your answer here.

(d) (2 points) At possible instrumentation point **4**

Your answer here.

(e) (2 points) At possible instrumentation point **5**

Your answer here.

LATE

minutes remaining

Hide Time

Manual Save

Navigation

- Question 1
- Question 2
- Question 3
- Question 4
- Question 5
- Extra Credit
- Pledge & Submit

(f) (2 points) At possible instrumentation point 6

Your answer here.

You are a software engineering manager who must decide whether or not to employ pair programming. Use the same sort of mathematical reasoning generally described in slides 18 and 19 of the lecture (https://eecs481.org/lectures/se-20-pairskill.pdf). Do not double-count factors. For each task, write the solo programming cost (as a number of hours), and then a newline, and then the pair programming cost (as a number of hours). This may be graded automatically, so please follow the format of one number (solo cost), and then a newline, and then the second number (pair cost). If relevant, use two figures after the decimal point (e.g., 1.23). The interpretation of the various features (e.g., "fewer total lines" or "fewer total defects") is as covered in class.

(g) (2 points) Task 1: 50,000 LOC program, Coding at 25 LOC/hour, Defect rate of 10 defects / KLOC, Defect fix time of 20 hours / defect, Pair programming is 20% slower for code writing, Pair programming results in 15% fewer total defects, Pair programming results in 15% fewer total lines of code.

Your answer here.

(h) (2 points) Task 2: 10,000 LOC program, Coding at 100 LOC/hour, Defect rate of 20 defects / KLOC, Defect fix time of 15 hours / defect, Pair Programming is 30% slower for code writing, Pair programming results in 10% fewer total defects, Pair programming results in 20% fewer total lines of code.

Your answer here.

(i) (5 points) Identify two potential human biases that are avoided by formal code inspection (when done correctly) but may be present in passaround code review. For each one, support it with a quote from the slides or reading (the quote could either show the human bias if things are done badly or show the good thing that happens if things are done well). Focus on aspects specific to code inspection. (After any quotes, use at most four sentences total.)

Your answer here.

(j) (5 points) Consider the Equifax breach timeline from Slide 5 of https://eecs481.org/lectures/se-03-measure.pdf. Consider defect report Severity and Priority as part of triage; for simplicity, each may separately either be *low*, *medium* or *high*. Note that the vulnerability was found in February but the patch was not deployed until just before August. Support or refute the claim that its internal Priority (for developers writing the patch) was likely high. Then support or refute the claim that its internal Severity (for developers writing the patch) was likely high. Justify one part of your argument with a quote from the reading or slides. After copying the quote, use at most four sentences.

Your answer here.

Question 5: Dataflow Analysis (13 points total)

LATE

minutes remaining

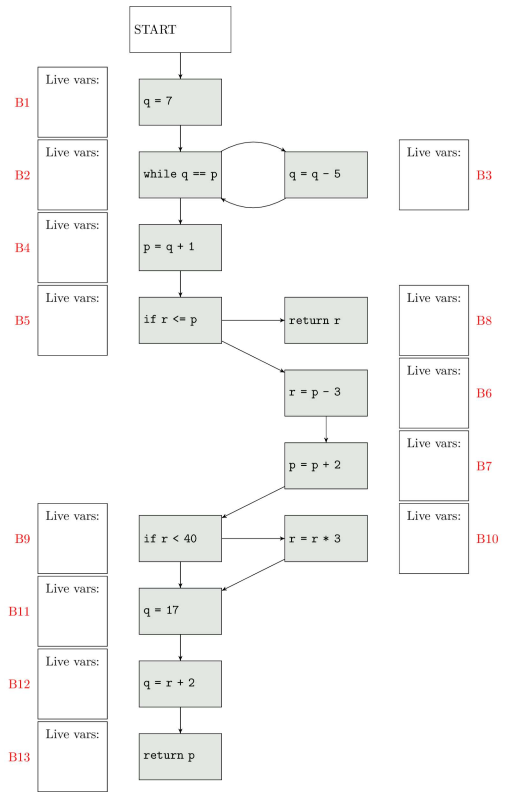
Hide Time

Manual Save

Navigation

- Question 1
- Question 2
- Question 3
- Question 4
- Question 5
- Extra Credit
- Pledge & Submit

Consider a *live variable dataflow analysis* for three variables, `p`, `q`, and `r` used in the control-flow graph below. We associate with each variable a separate analysis fact: either the variable is (1) possibly read on a later path before it is overwritten (live), or (2) it is not (dead). We track the set of live variables at each point: for example, if `p` and `q` are alive but `r` is not, we write `{p, q}`. The special statement `return` reads, but does not write, its argument. In addition, `if` and `while` read, but do not write all of the variables in their predicates. (You must determine if this is a forward or backward analysis.)



(1 point each) For each basic block `B1` through `B13`, write down the list of variables that are live *right before* the start of the corresponding block in the control flow graph above. Please list only the variable names in lowercase without commas or other spacing (e.g., use either `ab` or `ba` to indicate that `a` and `b` are alive before that block).

B1		B2		B3		B4	
B5		B6		B7		B8	
B9		B10		B11		B12	
B13							

Extra Credit

(1) What is your favorite part of the class so far? Alternatively, name a course staff member and list something you have enjoyed about that person. (1 point)

Your answer here.

(2) What is your least favorite part of the class so far? Alternatively, what improvement would you suggest for future semesters? (1 point)

Your answer here.

(3) In the context of your own experience in computing courses at Michigan, how do you think CSE should apply the lessons from Lewis and Shah's *How Equity and Inequity Can Emerge in Pair Programming* to maximize the benefits of working with partners but mitigate the risks they identified? Demonstrate that you have read the paper critically and tie it in to one of your

personal experiences (e.g., what is something that you think would work specifically here at UM, and what is something they talk about that you think specifically would be hard to do here at UM), going beyond a Generative AI summary. (2 points)

Your answer here.

LATE

minutes remaining

Hide Time

Manual Save

Navigation

- Question 1
- Question 2
- Question 3
- Question 4
- Question 5
- Extra Credit
- Pledge & Submit

(4) If you read any *other* optional reading (i.e., not Lewis and Shah), identify it and demonstrate to us that you have read it critically, going beyond a Generative AI summary. (2 points)

Your answer here.

(5) Which Generative AI tool(s) and version(s) did you use on this exam, if any? For which parts of the exam did you find them (or speculate that they would be) helpful or unhelpful? (Remember, free GenAI is allowed, so this is not cheating. This is to help us improve the course, not to get you in trouble.) If you used any other sources that you are uncertain about, you can cite them here. (1 points)

Your answer here.

Honor Pledge and Exam Submission

You must check the boxes below before you can submit your exam.

☐ I have neither given nor received unauthorized aid on this exam.

☐ *I am ready to submit my exam.*

Note that your submission will be marked as late. You can still submit, and we will retain all submissions you make, but unless you have a documented extenuating circumstance, we will not consider this submission.

Submit My Exam

Once you submit, you will be able to leave the page without issue. Please don't try to mash the button.

The exam is graded out of 100 points.