LATE

minutes remaining

Hide Time

Manual Save

Navigation

Question 1. Word Bank Matching (1 point each, 14 points total)

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

A. — A/B Testing	B. — Alpha Testing	C. — Beta Testing	D. — Competent Programmer Hypothesis
E. — Deliverables	F. — Formal Code Inspection	G. — Fuzz Testing	H. — Instrumentation
I. — Integration Testing	J. — Invariant	K. — Maintainability	L. — Mocking
M. — Oracle	N. — Pair Programming	O. — Passaround Code Review	P. — Perverse Incentive
Q. — Race Condition	R. — Regression Test	S. — Requirements	T. — Risk
U. — Spiral Development	V. — Streetlight Effect	W. — Test-driven Development	X. — Threat to Construct Validity
Y. — Threat to External Validity	Z. — Unit Testing		

Q1.1: Bruce is developing a video game. He creates a class Player and a class Car. His intended functionality is that the player should be able to get inside the car and drive it. He then writes a set of test cases to ensure the interaction between these two classes is functioning properly.

Q1.2: Mike is writing an application that allows moviegoers to reserve seats at the theater. Unfortunately, there is a bug that allows multiple users to reserve the exact same seat at the same time. If only Mike had included safeguards to prevent multiple reservations of the same seat.

Q1.3: Valeria is failing a public test case provided in the project spec. She uses a debugger to identify the issue and promptly fixes it. She writes a test case afterwards to alert her if the issue resurfaces as she continues the project.

Q1.4: Sasha has just finished making a small change in a file within her company's codebase. Prior to pushing it to production, she emails her colleague Jean to look over her changes at his convenience. Jean gives his approval a few hours later and Sasha pushes her code, committing the change to the main repository.

Q1.5: Jan writes a function for sin(x). He knows sin(90) is equal to 1, so he accordingly writes a test case. In this instance, 1 is the...

Q1.6: After discovering that her submissions to the Autograder are failing due to a timeout, Xiaoyu realizes that she must make some runtime optimizations. She begins by looking at her recently-written functions with the most lines of code since they are fresh in her mind. Unfortunately, the real bug is in a single innocuous line in which her code calls an external library function incorrectly.

Q1.7: Antoinette is interested in learning about how quickly it takes humans to understand code snippets. She conducts a study on a group of PhD students and concludes that humans, on average, take 45 seconds to understand how merge sort works. Her advisor is quick to chime in that her conclusion is flawed for this reason...

Q1.8: Eren is working on the backend of a website. He implements a function that makes a rather expensive database query. When writing test cases, he substitutes a hard-coded string in place of the query.

Q1.9: Anthony is working on his HW6 open-source contribution assignment and submits a pull request to Runelite. His code consists of many functions, but it is rejected because the functions lack test cases. If only he had implemented this quality assurance strategy...

Q1.10: Jordan is developing a video chat app for a client. He spends two months adding voice chat functionality and presents it to the client for feedback. He then spends two months adding live video before getting feedback from the client. He continues this iterative process.

LATE

minutes remaining

Hide Time

Manual Save

Navigation

Q1.11: Larry rigorously tests a mathematical function he has written and realizes that the output is always greater than or equal to the input.

Q1.12: Jeremy has just written a file that will be used in the software for a pacemaker, a device used for heart arrhythmias. He schedules a meeting with members of his team where they will sit down and go through his code, line by line, with the goal of identifying bugs.

Q1.13: Roxanne is preparing to release a new social media app. To catch bugs that may surface during common use, she hires a group of social media influencers to test out the app and report any issues.

Q1.14: Annie is a developer for an online retailer. She is considering changing the color of the 'BUY NOW' button from green to blue. She decides to change the color for a subset of customers and compares the difference in purchasing activity before coming to a decision.

Question 2: Coverage (18 points total)

You are given the Python function below.

```
def awesome_grizzly (j: bool, k: bool, l: bool):
    STMT_1
    if (( j or k) and (not k and l)):
        STMT_2
    else:
        STMT_3
    if ((j and l) and not (j or k) and l):
        STMT_4
    elif ((not j and l) or not (not k)):
        STMT_5
        if (k and not l):
            STMT_6

```

Q2.1 (4 points) Calculate the minimum statement coverage attainable using one test input and provide such an input (i.e., values of j, k, and l).

Calculate the minimum statement coverage attainable using one test input and provide such an input (i.e., values of values of j, k, and ls).

Q2.2 (6 points) Provide a single minimum set of test inputs(s) that achieves maximum statement AND maximum path coverage for this particular program. Consider only feasible paths and reachable statements. In one sentence, explain why this is the smallest number of test inputs that can maximize both statement and path coverage.

Provide a single minimum set of test inputs(s) that achieves maximum statement AND maximum path coverage for this particular program. Consider only feasible paths and reachable statements. In one sentence, explain why this is the smallest number of test inputs that can maximize both statement and path coverage.

Q2.3 (5 points total, 1 point per selection) Next, consider the C code below. For simplicity, assume that `bool` means an integer 0 for False, and 1 for True. Make selections for the operators `P`, `Q`, `R`, `S`, and `T` such that:

1. The *statement coverage* induced by executing the single test case `ecstatic_bohr(0, 1, 1)` is maximized.

Assume that covering all of STMT_1-STMT_8 is 100% statement coverage (i.e., ignore any other lines), though this may not be the maximum attainable.

```
void ecstatic_bohr (bool j, bool k, bool l) {
    if (j  •P•  k) {
```

LATE

minutes remaining

Hide Time

Manual Save

Navigation

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

```
3           STMT_1;
4           if (k  •Q•  l) {
5               STMT_2;
6           } else {
7               STMT_3;
8           }
9       } else {
10          STMT_4;
11      }
12
13      if (j   •R•  l) {
14          STMT_5;
15      } else if (k •S• j) {
16          STMT_6;
17      }
18
19      if (l  •T•  k) {
20          STMT_7;
21      } else {
22          STMT_8;
23      }
24  }
25
```

Operator	Maximize Statement Coverage
P	◯ != ◯ < ◯ ≥
Q	◯ != ◯ < ◯ ≥
R	◯ != ◯ < ◯ ≥
S	◯ != ◯ < ◯ ≥
T	◯ != ◯ < ◯ ≥

Q2.4 (3 points) Support or refute: it is harder to maximize branch coverage for code with a lower Maintainability Index. Use at most four sentences.

Support or refute: it is harder to maximize branch coverage for code with a lower Maintainability Index.

Question 3: Short Answer and Potpourri (28 points total)

Provide answers to each question below.

Q3.1 (3 points) Risk and Measurement

You are a software engineering manager. You are considering a proposal in which 30% of the resources currently used for integration testing would instead be reallocated and used for a different dynamic analysis (e.g., something like Chaos Monkey or Driver Verifier, etc.). Identify two risks associated with this proposal and one benefit associated with this proposal. For each, identify one associated measurement that might be taken to reduce uncertainty (i.e., to determine the degree to which that positive or negative outcome occurred).

Your answer here.

Q3.2 (2 points each; 6 points total) Pair programming and Process

You are a manager at WebFlix and need to decide whether or not to employ pair programming for a series of tasks. Since pair programming tends to produce code of higher quality, you are willing to opt for pair programming for a particular task so long as there is not an increase in total costs of more than 59%. The table below summarizes the various costs and benefits of using pair programming for each task.

For "Pair Programming increase in Cost per Hour (%)", 100% would mean that pair programming carries twice the cost of solo programming. Similarly, a "Pair Programming decrease in Total Hours (%)" of 40% means that a task that takes 10 hours solo would take 6 hours with pair programming.

Task	Total Hours	Cost Per Hour	Pair Programming decrease in Total Hours (%)	Pair Programming increase in Cost per Hour (%)
A	27	10	39	100
B	34	5	83	113
C	12	17	84	139

(2 points each) For each of the following tasks, decide whether to employ pair programming.

A
- ◯ Yes, use Pair Programming
- ◯ No, do not use Pair Programming

B
- ◯ Yes, use Pair Programming
- ◯ No, do not use Pair Programming

C
- ◯ Yes, use Pair Programming
- ◯ No, do not use Pair Programming

Q3.3 (3 points) Development Processes

In three sentences or fewer, describe the differences between spiral development and waterfall development.

Your answer here.

Q3.4 (3 points) Generating Test Inputs

Compare and contrast fuzz testing and constraint-based solvers for generating test inputs: what aspects do they share and where do they differ? Give one example program for which we would expect a fuzzer to outperform a constraint-based solver. Give one example of a program for which we would expect a constraint-based solver to outperform a fuzzer. Use at most six sentences.

Your answer here.

Q3.5 (3 points) Code Review

Identify a developer expectation of modern passaround code review that is commonly met. Identify a developer expectation of modern passaround code review that is rarely met. Describe a buggy patch that modern passaround code review is unlikely to correctly reject. Use at most six sentences.

Your answer here.

Q3.6 (2 points each; 10 points total) Software Engineering Comparisons

Consider each of the following pairs of techniques, tools, or processes. For each pair, give a class of defects or a situation for which the first does better than the second (i.e., is more likely to succeed and reduce software engineering effort and/or improve software engineering outcomes) and explain why. For full credit, each explanation must include why the second is worse in that situation (simply indicating how the first is good is not sufficient). Use at most three sentences per answer.

maximizing branch coverage vs. pair programming

Your answer here.

static dataflow analysis vs. unit tests

Your answer here.

mocking vs. passaround code review

Your answer here.

the Eraser dynamic analysis vs. fuzz testing

Your answer here.

regression testing vs. formal code inspection

Your answer here.

LATE

minutes remaining

Hide Time

Manual Save

Navigation

Question 4. Mutation Testing (29 points)

Consider the code snippet below defining a function `foo`:

```
1  def foo(y):
2      if (y < 0):       # Mutant 1: y <= 0
3          # Invalid input
4          return -1
5      elif y == 0:
6          return 0
7      elif y == 1 or y == 2:  # Mutant 2: y == 1 and y == 2
8          return 8
9      else:
10         return foo(y - 1) + foo(y - 2)  # Mutant 3: foo(y - 1) - foo(y - 2)
11
```

(a) (1 point per field, 20 points total) Complete the table below by indicating whether each test kills each Mutant. (`Y` for killed and `N` for not killed). Oracle stands for the expected output of `foo` run on the corresponding input. **Be careful**: subsequent subquestions depend on correctly understanding this subquestion.

Test #	Input (y)	Oracle (`foo(y)`)	Mutant 1	Mutant 2	Mutant 3
(Q4.a.0): Test 0	`0`	#	Y/N	Y/N	Y/N
(Q4.a.1): Test 1	`1`	#	Y/N	Y/N	Y/N
(Q4.a.2): Test 2	`2`	#	Y/N	Y/N	Y/N
(Q4.a.3): Test 3	`3`	#	Y/N	Y/N	Y/N
(Q4.a.4): Test 4	`4`	#	Y/N	Y/N	Y/N

(b) (1 points) What is the mutation score for tests `0`-`4` using Mutants 1-3?

The mutation score is...

(c) (1 point) What is the mutation score for test `0` using Mutants 1-2?

The mutation score is...

(d) (1 point) What is the mutation score for tests `0`-`4` using just Mutant 1?

The mutation score is...

(e) (2 points per mutant, 6 total) Make **at most one edit** each to create **THREE NEW** and **DIFFERENT** mutants of `foo`. Exactly one of your three new first-order mutants should be killed by when provided the same test input `y=2`.

LATE

minutes remaining

Hide Time

Manual Save

Navigation

(Make at most one edit to the code to create a new mutant that is different from Mutants 1-3. Repeat this process to produce a total of 3 new, mutually different mutants and make sure the kill score with input `y=2` is 1/3. For example, you might change line 2 from `if num < 0` to `if num > 0` but not to `if num <= 0` because that's already Mutant 1 in this question.)

You should not introduce any loops as part of your mutations. Make sure that your mutants correspond to valid Python3 code — syntactically invalid mutants may receive no credit. Moreover, please do not attempt to subvert this question by modifying the code to immediately return a value — you are asked to make first-order mutants.

Attempting to submit code that infinitely loops, that interacts with any I/O, that imports other libraries, or that shells out is a violation of the honor code. Doing so will result in a 0 for the entire exam.

Mutant X:

```
1  def foo(y):
2      if (y < 0):       # Mutant 1: y <= 0
3          # Invalid input
4          return -1
5      elif y == 0:
6          return 0
7      elif y == 1 or y == 2:  # Mutant 2: y == 1 and y == 2
8          return 8
9      else:
10         return foo(y - 1) + foo(y - 2)  # Mutant 3: foo(y - 1) - foo(y - 2)
11
```

Mutant Y:

```
1  def foo(y):
2      if (y < 0):       # Mutant 1: y <= 0
3          # Invalid input
4          return -1
5      elif y == 0:
6          return 0
7      elif y == 1 or y == 2:  # Mutant 2: y == 1 and y == 2
8          return 8
9      else:
10         return foo(y - 1) + foo(y - 2)  # Mutant 3: foo(y - 1) - foo(y - 2)
11
```

Mutant Z:

```
1  def foo(y):
2      if (y < 0):       # Mutant 1: y <= 0
3          # Invalid input
4          return -1
5      elif y == 0:
6          return 0
7      elif y == 1 or y == 2:  # Mutant 2: y == 1 and y == 2
8          return 8
9      else:
10         return foo(y - 1) + foo(y - 2)  # Mutant 3: foo(y - 1) - foo(y - 2)
11
```

Question 5: Dataflow Analysis (11 points total)

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

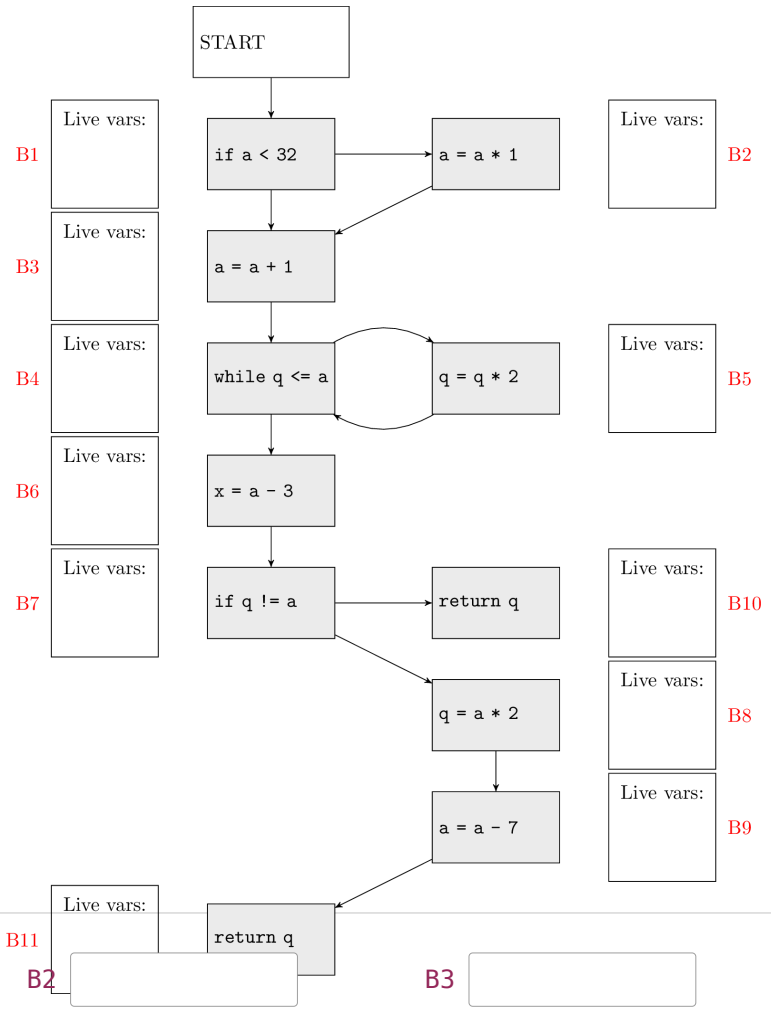
LATE

minutes remaining

Hide Time

Manual Save

Navigation



B1 B2 B3 B4

(1 point each) For each basic block B1 through B11, write down the list of variables that are live *right before* the start of the corre the control flo ase list only th n lowercase wi other spacing (e.g., use either `ab` or `ba` to indicate that `a` and `b` are alive before that block).

B5 B6 B7 B8

B9 B10 B11

Extra Credit

Each question below is for 1 point of extra credit unless noted otherwise. We are strict about giving points for these answers. No partial credit.

(1) What is your favorite part of the class so far?

Your answer here.

(2) What is your least favorite part of the class so far?

Your answer here.

(3) If you read any optional reading, identify it and demonstrate to us that you have read it critically. (2 points)

Your answer here.

(4) If you read any *other* optional reading, identify it and demonstrate to us that you have read it critically. (2 points)

Your answer here.

(5) In your own words, identify and explain any of the bonus psychology effects or ethical considerations presented in class on the colored bordered slides or in a "long instructor post" on Piazza. (2 points)

Your answer here.

Honor Pledge and Exam Submission

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

LATE

minutes remaining

Hide Time

Manual Save

Navigation

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

☐ 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.