LATE

minutes remaining

Hide Time

Manual Save

Navigation

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

For each statement below, input the letter of the term that is *best* 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's Hypothesis	F. — Dynamic Analysis	G. — Formal Code Inspection	H. — Fuzz Testing
I. — Integration Testing	J. — Milestone	K. — Mocking	L. — Oracle
M. — Pair Programming	N. — Pass Around Code Review	O. — Perverse Incentive	P. — Priority
Q. — Race Condition	R. — Regression testing	S. — Risk	T. — Severity
U. — Static Analysis	V. — Streetlight Effect	W. — Triage	X. — Unit Testing
Y. — Waterfall Model			

Q1.1: Figmo inc. creates a new policy where developers earn $1,000 per file created. This results in each developer adding more than 100,000 empty files to the codebase.

Q1.2: Henry got a score of 1/5 on their EECS280 project. However, after changing one line they got 5/5 and passed all test cases. The change on that one line was changing a subtraction sign to an addition sign.

Q1.3: Developers at Fire Nation Company are trying to find bugs in their code. They use BugCatcher to find runtime errors in their codebase. BugCatcher is a helpful tool that reads through the source code without actually running it to make sure no severe bugs are introduced to Fire Nation Company's codebase.

Q1.4: Larry is told by his supervisor to work with Youcef. To start off they decide that Larry will code and Youcef will review each line as it is written. They will switch roles every 30 minutes.

Q1.5: Mesla inc.'s developers have over 1,000 bugs to fix. To decide which bugs to prioritize, the defect report uses *this term* to categorize each of the bugs with the impact the bug would have on Mesla inc.'s operations. Here, *this term* is a concept covered in EECS 481.

Q1.6: Zuko is writing a program to convert the weather from Fahrenheit to Celsius. They know that 32 F = 0 C and wrote a test case to test whether converting 32 F returns 0 C. In this case, 0 C is the ...

Q1.7: Wenxin is in charge of Bank of Michigan's development team. They plan to take a particular approach to develop their software. In particular, they will first gather requirements from customers, then design and implement the code, and finally test the code.

Q1.8: EECSon mobile is a new streaming service company. They are planning a new feature, where users can recommend shows to other users. To evaluate if users would like this feature, they plan on using the following approach. They will release the new feature to half of their users, who were selected randomly. After two weeks, they will collect feedback from all users through a survey.

Q1.9: MoonChips inc. has recently been made aware of a bug that causes their links to be unclickable. Fanghao is trying to fix this bug, and looks over recently made changes, as those changes were fresh in Fanghao's mind. The bug was ultimately caused by an obscure line elsewhere, where the URL is set to null.

Q1.10: Veecsa co. requires all changes to have another developer's approval, before merging to the codebase. This work can be done offline too, without them meeting in person.

Q1.11: Max is writing an application (which is implemented as a multi-threaded program) for students to reserve a study room. Unfortunately a bug allows multiple people to sign up for the same study room at the same exact time.

Q1.12: Software development tool, Phabricator assigns labels of “Unbreak Now!”, “High”, “Normal”, “Low”, and “Lowest” to indicate the importance or urgency of fixing a defect.

Q1.13: A new streaming service company GlazeBook wants to test their new chatbot. To do this, they randomly generate 100,000 inputs with valid and invalid characters, and see if the chatbot responds appropriately.

Q1.14: Gogo inc. is trying to create a social media app. They have created the login and the dashboard and want to test how they work together to ensure that users will not have any issues navigating between the two pages.

LATE

minutes remaining

Hide Time

Manual Save

Navigation

Question 2. Code Coverage (20 points)

You are given the following code. (Please scroll down to see the all functions.) Assume that statement coverage applies only to statements marked STMT_#. In this question, we consider the entire program when calculating coverage. The code starts at `main()`, but even if some methods are not executed during the program execution for a given input, we still consider coverage with respect to **all 8 STMTs**.

```
1  void checkStrings(string a, string b) {
2      if (a != "") {
3          STMT_1;
4          if (strlen(a) > 4) {
5              STMT_2;
6          }
7          STMT_3;
8      }
9      else {
10         return;
11     }
12
13     if (a[0] != b[0]) {
14         if (a[0] == 'b') {
15             STMT_4;
16         }
17         STMT_5;
18     }
19 }
20
21 int sumDigits(int x) {
22     int s = 0;
23     while (x > 0) {
24         STMT_6;
25         s = s + (x % 10);   // Note: '%' is the modulo operator
26         x = x / 10;
27     }
28     return s;
29 }
30
31 int main(string a, string b, int x, int y) {
32     int s1 = sumDigits(x);
33     int s2 = sumDigits(y);
34     if (s1 == s2 && s1 != 0 && s2 != 0) {
35         STMT_7;
36         checkStrings(a, b);
37     }
38     STMT_8;
39
40     return 0;
41 }
42
```

(a) (3 points)

Give a test input for `main()` that achieves EXACTLY **25%** statement coverage.
In the context of this question, you have to pick inputs from the following strings: `{ red, blue, green, black, brown, pink }` and the following ints: `{ 4444, 97, 790, 2462, 718, 1091 }`.
This is important.

Please write your answer in the following format, as a call to `main()`, because an auto-grader will be used to facilitate grading.
Ex: `main('blue', 'blue', 97, 97)`
If no such input exists, write `IMPOSSIBLE` (all capital letters).

Your answer here.

(b) (3 points)

Give a test input to `main()` that achieves the **lowest** statement coverage. Also, what is its coverage?
You can use any string or integer for the input parameters. No need to restrict to those in (a).
This is important.
Please enter your answer as a call to `main()` followed by the statement coverage (as a percentage), in the following format.
Ex: `main('blue', 'blue', 97, 97) 75%`

Your answer here.

(c) (2 points)

In regards to path coverage, how many paths does the input you gave in (b) cover?
This is important.
Please enter **ONLY** the number of paths (as a number, like 4) in the text box below.

Your answer here.

(d) (2 points)

What is the minimum number of test cases needed to get exactly **80%** branch coverage?
For this question, we only consider branches created by the 5 if-statements (we do not consider branches from loops), meaning there are 10 branches total.
This is important.
Please enter **ONLY** the number of test cases (as a number, like 1).

Your answer here.

```
1  int main (string a, string b, int x, int y) {
2     int s1 = sumDigits(x);
3     int s2 = sumDigits(y);
4
5     STMT_7;
6     checkStrings(a, b);
7     STMT_8;
8
9     return 0;
10 }
11
```

(e) (4 points)

Suppose `main()` in the previous questions is changed to the `main()` **above**. The differnce is that, now, `STMT_7` and `checkStrings(a,b)` are no longer within an if-statement and instead are always executed.

How does the **highest** possible branch coverage for a single test input change, if at all? Recall that we only consider branches created by if-statements. Hint: After the change, there are now 4 if-statements, meaning that there are 8 branches.
This is important
Please enter the **old highest** coverage and **new highest** coverage (both as percentages) on **one line seperated by a comma** (space doesn't matter). For example: `10%,10%`

Your answer here.

(f) (6 points)

A company is attempting to locate bugs in their software by maximizing one coverage metric.
For **each** of the following metrics (branch coverage, statement coverage, path coverage), discuss one benefit of maximizing it and one area where it falls short.
Please limit your entire answer to at most 6 sentences. We suggest you use two sentences for each metric (one for the benefit, and one for the shortcoming).

LATE

minutes remaining

Hide Time

Manual Save

Navigation

LATE

minutes remaining

Hide Time

Manual Save

Navigation

Your answer here.

Question 3. Short Answer (25 points)

For each of the two bugs below, please describe (1) an example where the bug would have high severity and (2) a scenario where the bug would have low severity. Explain why in both cases.
Please limit your answer to at most 4 sentences, for each bug.

(a) (3 points)

A bug that causes the color of text to change.

Your answer here.

(b) (3 points)

A bug that causes your app to occasionally close unexpectedly.

Your answer here.

Each of the following questions (c)-(e) gives a pair of concepts. It can be a pair of techniques, or a pair of tools, or a pair of processes, etc. Given a pair, please explain why the first one **could** be better (e.g., more likely to succeed and reduce software engineering effort, or to improve software engineering outcomes) than the second. Your explanation may depend on the specific pair given. For full credit, please also briefly describe a situation to illustrate why the first is better than the second. Note that your answer should not only explain why the first may be good, but also why the second may not be good. For each question, please use no more than four sentences.

(c) (3 points)

Integration Testing vs. Regression Testing

Your answer here.

(d) (3 points)

Alpha Testing vs. A/B Testing

Your answer here.

(e) (3 points)

Unit Testing vs. Static Dataflow Analysis

Your answer here.

(f) (5 points)

Suppose `Test A` has lower statement coverage than `Test B`.

Support or Refute: It is impossible for `Test A` to expose more bugs than `Test B`. If supporting, explain why it is impossible. If refuting, give a specific example or situation of where `Test A` could expose more bugs than `Test B`. Please limit your answer to at most four sentences.

Your answer here.

(g) (5 points)

As a software engineer at 481Company, you are responsible for developing rapid changing software (which requires to be constantly updated). Your coworker Darwin Nunez suggests that the company switches from agile development to the waterfall development model due to its simple process. Do you agree or disagree with Darwin? Explain which development methodology you believe 481Company should follow, and explain why it is better than the other methodology. Please limit your answer to at most four sentences.

Your answer here.

Question 4. Mutation Testing & Invariants (15 points)

The following function `is_prime` checks whether a given integer `n` is a prime number. While the following code shows a total of five first-order mutants (see comments), in this question, you will only be asked to analyze a specific subset of them.

```
1  def is_prime(n):
2      if n < 2:                        # Mutant 1: if n <= 2:
3          return False
4
5      i = 2                            # Mutant 2: i = 1
6      while i * i <= n:                # Mutant 3: while i*i < n:
7          if n % i == 0:               # Mutant 4: if n % i == 1:
8              return False
9          i += 1                       # Mutant 5: i += 2
10
11     return True
12
```

(a) (6 points)

Complete the table below by indicating whether or not each test case kills **Mutants 2 and 5**. Note that column `oracle` shows the ground-truth (i.e., desired) return value of `is_prime(n)`. (K: Killed, N: Not killed.)

test case	n	oracle	Mutant 1	Mutant 2	Mutant 5
1	n=2	T	K	○ K ○ N	○ K ○ N
2	n=4	F	N	○ K ○ N	○ K ○ N
3	n=5	T	N	○ K ○ N	○ K ○ N

(b) (2 points)

What is the mutation adequacy score for test cases 1-3 from (a), using **only Mutant 1**?
This is important. To facilitate auto-grading, please write your answer in the following format. Round the score to two decimal points and write **ONLY** the score in the box. For example: `0.12`

Your answer here.

(c) (2 points)

What is the mutation adequacy score for test cases 1-3 from (a), using **Mutants 1, 2, and 5**?
This is important. To facilitate auto-grading, please write your answer in the following format. Round the score to two decimal points and write **ONLY** the score in the box. For example: `0.12`

Your answer here.

(d) (2 points)

Please write (1) one advantage and (2) one disadvantage, of higher-order mutation (i.e., multiple mutations per mutant). Limit your entire answer to at most two sentences. We suggest you allocate one sentence to advantage and one to shortcoming.

Your answer here.

(e) (3 points)

Consider the following three invariants for `is_prime`. Each invariant is evaluated right after the function terminates. In this quesiton, we define an invariant to be valid, if it always holds (i.e., evaluated to True) for all positive integers `n` (i.e., the input to `is_prime`).

For each invariant, please indicate whether or not it is valid, and if yes, whether the invariant can be falsifed by any of the Mutants 1-5.

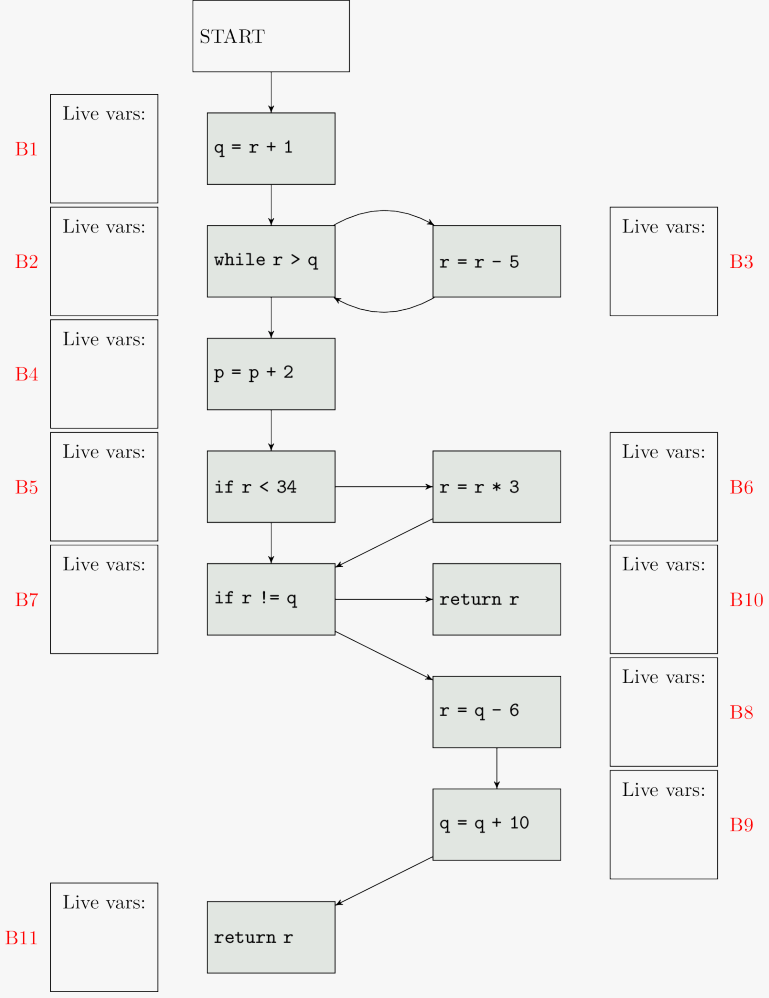
Invariant 1: `i is undefined || (n%i==0)`
- ◯ A) Invalid.
- ◯ B) Valid. At least one of the Mutants 1-5 can falsify it.
- ◯ C) Valid. None of the Mutants 1-5 can falsify it.

Invariant 2: `i is undefined || (i <= n+1)`
- ◯ A) Invalid.
- ◯ B) Valid. At least one of the Mutants 1-5 can falsify it.
- ◯ C) Valid. None of the Mutants 1-5 can falsify it.

Invariant 3: `i is undefined || (i >= 2)`
- ◯ A) Invalid.
- ◯ B) Valid. At least one of the Mutants 1-5 can falsify it.
- ◯ C) Valid. None of the Mutants 1-5 can falsify it.

Question 5: Dataflow Analysis (11 points total)

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 B11, 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

Question 6. Dynamic Analysis (15 points)

You have developed a dynamic analysis tool, called Marbles, that aims to identify potential memory leaks.

Marbles works by tracking and logging memory allocations and deallocations during program execution. The program under analysis can be run multiple times simultaneously at different speeds, with each execution being assigned with a unique thread id number (for example, Thread 1). Marbles logs information about each memory operation, including the allocated memory address, deallocated memory address (if applicable), size of allocated memory, and the thread id.

However, Marbles can be buggy while it is analyzing programs, causing it to produce false positives and/or false negatives in the log files.

Recall the following:
A *false positive* occurs when a tool incorrectly predicts the presence of a condition or event that is not actually present.
A *false negative* occurs when a tool incorrectly predicts the absence of a condition or event that is actually present.

In what follows, you will see three different programs. We have run Marbles on all of them. The log file for each program is also shown below. For each program, you will be asked to determine if Marbles would report a memory leak based on the Marbles log file. In addition, you will also be asked to determine whether Marbles has incurred any false positives/negatives during the logging process.

Using this information, answer the following subquestions (a) - (e).
NOTE: You may need to scroll down on some of the code snippets to view the full program and/or log file.

```
void p1() {
    // Dynamically allocates a pointer (4 bytes)
    int* ptr = new int;
    *ptr = 42;
}

Marbles log file:
Thread 1: Allocate 4 bytes at address 0x6000
Thread 2: Allocate 4 bytes at address 0x6100
```

LATE

minutes remaining

Hide Time

Manual Save

Navigation

(ai) (1 points) **True / False**: Marbles would report a memory leak for `p1`, based on the log file.

- ◯ True
- ◯ False

(aii) (1 points) **True / False**: Marbles incurred a false positive or false negative for `p1`, during its logging process.

- ◯ True
- ◯ False

```
void p2() {
    // Dynamically allocates a new buffer (100 bytes), a c-style array
    char* buffer = new char[100];
    // Deallocates buffer
    delete[] buffer;
    // Dynamically allocates a new buffer (50 bytes)
    buffer = new char[50];
}

Marbles log file:
Thread 1: Allocate 100 bytes at address 0x6000
Thread 2: Allocate 100 bytes at address 0x6100
Thread 2: Deallocate 100 bytes at address 0x6100
```

(bi) (1 points) **True / False**: Marbles would report a memory leak for `p2`, based on the log file.

- ◯ True
- ◯ False

(bii) (1 points) **True / False**: Marbles incurred a false positive or false negative for `p2`, during its logging process.

- ◯ True
- ◯ False

```
void p3() {
    // Dynamically allocate array1 (20 bytes)
    int* array1 = new int[5];
    // Dynamically allocate array2 (12 bytes)
    int* array2 = new int[3];
    for (int i = 0; i < 2; ++i) {
        // Dynamically allocates a new tempArray (40 bytes)
        int* tempArray = new int[10];
        // Deallocates the new tempArray
        delete[] tempArray;
    }
    // Deallocate array1 and array2
    delete[] array1;
    delete[] array2;
}


Marbles log file:
Thread 1: Allocate 20 bytes at address 0x6000
Thread 1: Allocate 12 bytes at address 0x6100
Thread 2: Allocate 20 bytes at address 0x6400
Thread 2: Allocate 12 bytes at address 0x6500
Thread 2: Allocate 40 bytes at address 0x6600
Thread 1: Allocate 40 bytes at address 0x6200
Thread 2: Deallocate 40 bytes at address 0x6600
Thread 1: Deallocate 40 bytes at address 0x6200
Thread 2: Allocate 40 bytes at address 0x6700
Thread 2: Deallocate 40 bytes at address 0x6700
Thread 1: Allocate 40 bytes at address 0x6300
Thread 1: Deallocate 40 bytes at address 0x6300
```

LATE

minutes remaining

Hide Time

Manual Save

Navigation

(ci) (1 points) **True / False**: Marbles would report a memory leak for `p3`, based on the log file.

- ◯ True
- ◯ False

(cii) (1 points) **True / False**: Marbles incurred a false positive or false negative for `p3`, during the logging process.

- ◯ True
- ◯ False

(d) (2 points)

Describe a scenario where Marbles may produce false positives, and a scenario where it may produce false negatives. Please limit your entire answer to at most 4 sentences.

Your answer here.

(e) (2 points)

Please explain at least one limitation of a dynamic analysis tool (which is assumed to be non-buggy) for identifying memory leaks. Limit your answer to at most 3 sentences.

Your answer here.

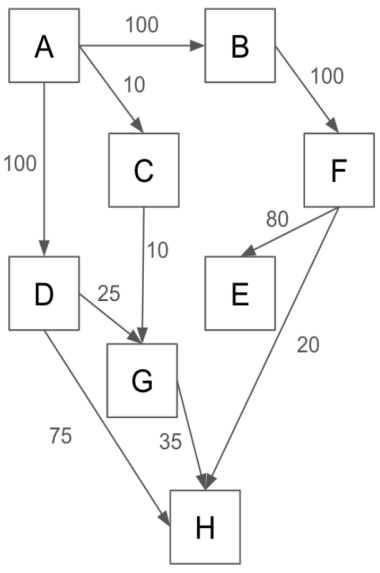
(f) (2 points)

Explain how a QA team might use a dynamic analysis tool to help test a multi-threaded program for race conditions. Additionally, list at least one pro and one con of using a dynamic analysis tool to detect race conditions.
Limit your entire answer to at most five sentences.

Your answer here.

(g) (3 points)

The QA team is now interested in profiling a program P, with the goal of understanding which execution paths are taken most frequently and how they can optimize P better on those paths.



After running the profiler, the team obtained a "path profile graph", which is shown to the right. Nodes in this graph represent disjoint portions of the program. An edge in this graph represents flow from one node (that represents a portion of P) to another. For instance, we have an edge from node A to node D in this graph for P. Assume program execution starts from node A.

The number labeled on an edge indicates the number of times that edge was executed. For instance, the edge from A to D has label 100. This means the profiler observed flows from A to D for 100 times, during the execution of P. Note that it does not matter for how long P was profiled, or how many times P was executed. All edge labels are normalized.

You want to figure out which path is the most frequently taken on P, so that you can optimize P on that path. Please list all possible paths P can take, based on the profile information, ordered from the most likely to be taken to least likely to be taken.

IMPORTANT: Please format your answer as a comma separated list, where each path is separated by a comma. An example answer entry would be formatted like: `ABFH, ADGH` (where `ABFH` is a path `A->B->F->H`).

Your answer here.

LATE

minutes remaining

Hide Time

Manual Save

Navigation

Extra Credit

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

Your answer here.

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

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) So far, how many lectures have you attended in person? How many remote? This is the first semester we try this new participation mechanism (allowing a mix of remote/in person participation). We'd love to hear feedback from you, if any. (2 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.