

1. (1pt) **Name:** _____

INSTRUCTIONS: Carefully read each question, and write the answer in the space provided. If answers to free response questions are written obscurely, zero credit will be awarded. The correct answer to a free response question with a short answer (i.e., one word or phrase) will never contain any significant words used in the question itself (i.e., “crossword rules”). You are permitted to use one 8.5x11 inch sheet of paper (double-sided) containing **hand-written** notes; all other aids (other than your brain) are forbidden. Questions may be brought to the instructor.

You have **150 minutes** to complete the exam.

For **TRUE** or **FALSE** and multiple choice questions, circle your answer.

On free response questions only, you will receive **20%** credit for any question which you leave blank (i.e., do not attempt to answer). Do not waste your time or mine by making up an answer if you do not know. (Note though that most questions offer partial credit, so if you know part of the answer, it is almost always better to write something rather than nothing.)

To get credit for this question, you must:

- Print your name (e.g., “Martin Kellogg”) in the space provided on this page.
- Print your UCID (e.g., “mjk76”) in the space at the top of **each** page of the exam.

	Writing your name and UCID:	_____ / 1
	I. Very Short Answer:	_____ / 33
	II. Matching:	_____ / 24
	III. Short Answer:	_____ / 45
Contents (blanks for graders only):	IV. “Your Choice” Reading Quiz	_____ / 3
	V. DBQs:	_____ / 44
	VI. Extra Credit:	_____ / 0
	Total:	_____ / 150

I. Multiple Choice and Very Short Answer (33pts). In the following section, either circle your answer (possible answers appear in **bold**) or write a very short (one word or one phrase) answer in the space provided. No partial credit is possible in this section.

2. (2pt) **TRUE / FALSE:** software engineers typically assume that end users will describe bugs they encounter in a helpful manner.
3. (2pt) Because human attention spans are short, it's important to make sure that the edit-test-debug cycle for a project takes less than about ten _____.
 - A milliseconds
 - B seconds
 - C minutes
 - D hours
4. (2pt) **TRUE / FALSE:** it is possible to “foreclose” on a project that has too much technical debt by taking ownership of it in exchange for forgiving the technical debt.
5. (2pt) Which of these programming paradigms' key mathematical formalism is a Turing machine?
 - A functional
 - B object-oriented
 - C imperative
6. (3pt) Match the terms in the right-hand column with the definitions in the left-hand column, in the context of a model-view-controller software architecture, by drawing a line between the term and its matching definition.

Model	accepts input and converts it to commands
View	representation of information
Controller	the application's dynamic data structure
7. (2pt) **TRUE / FALSE:** the type system in every statically-typed language proves the same property.
8. (2pt) **TRUE / FALSE:** path coverage subsumes branch coverage.
9. (2pt) Which of the following testing techniques might be especially useful while refactoring? (Select all that apply.)
 - A partition testing
 - B regression testing
 - C differential testing
 - D stress testing
10. (2pt) Before bugs are assigned to developers, they need to be _____ to determine which bug reports have the highest priority.
11. (2pt) Good requirements are _____, rather than relying on informal and ambiguous natural language.

12. (2pt) Version control systems' "conflict-free" detection can have:
- A false positives
 - B false negatives
 - C both of the above
 - D none of the above
13. (2pt) The Liskov Substitution Principle states that "any _____ object should be safe to use in place of a _____ object at run time."
14. (2pt) In the **cathedral** / **bazaar** model of open-source development, users are treated as co-developers and projects are democratically organized.
15. (2pt) Which of the following are common causes of emergencies in deployed services? (Select all that apply.)
- A incorrect error handling
 - B configuration changes
 - C hardware failures
 - D human/process error
16. (2pt) Netflix's Chaos Monkey testing system randomly chooses a server and _____ it during its usual hours of activity.
17. (2pt) **TRUE** / **FALSE**: technical interviews like those commonly conducted by tech companies when hiring software engineers have been shown by academic researchers to correctly predict employee's job performance.

- | | | | |
|------------------------------|---------------------------|---------------------------|---------------------------------|
| A. Availability | B. A/B testing | C. Stress testing | D. Pair programming |
| E. Factory pattern | F. Adapter pattern | G. Code review | H. Static analysis |
| I. Microservice | J. Cohesion | K. Type system | L. Partition testing |
| M. Canary testing | N. Hard fork | O. Anti-pattern | P. Pipe-and-filter |
| Q. Regression testing | R. Hermetic build | S. Project manager | T. Industrial researcher |

II. Matching (24pts). This section contains a collection of terms discussed in class in an “Answer Bank” (choices **A.** through **T.**). Each question in this section describes a situation associated with an answer in the Answer Bank. Write the letter of the term in the Answer Bank that best describes each situation. Each answer in the Answer Bank will be used at most once.

18. (2pt) _____ Andreas has implemented a new login feature for their web application. They deploy it to a subset of users for two weeks and record what percentage of signups succeed in both the new and old login flows.
19. (2pt) _____ Joel insists that his team’s code can be shipped to a fresh VM without any manual setup.
20. (2pt) _____ When deploying a change, Jeff uses a tool to automatically only update a small set of servers before the change is deployed to the whole fleet.
21. (2pt) _____ Martin’s team insists that he simplifies code or adds a code comment instead of just explaining complexity to them.
22. (2pt) _____ Charity decides to do a major refactoring, because there are several modules in her team’s system that seem to have several jobs, and she wants to split them apart.
23. (2pt) _____ Hillel writes a script that automatically checks that his team’s webservice is responding to calls once every minute.
24. (2pt) _____ Management at Mike’s company organizes each team around a single business capability, and Mike’s team’s software architecture reflects that.
25. (2pt) _____ Jesse implements a method that returns a generic interface rather than a specific subtype, to hide implementation details from callers.
26. (2pt) _____ Ken is having trouble with a particularly difficult technical problem, so he schedules a call with one of his coworkers. Together, they write the difficult code.
27. (2pt) _____ Eric notices several copies of the same duplicated code in his team’s codebase.
28. (2pt) _____ Mary is working on an AI product that requires her to stay aware of the latest work in AI.
29. (2pt) _____ Goran chooses inputs that represent specific sub-domains of the inputs he expects his program to take.

III. Short answer (45pts). Answer the questions in this section in at most three sentences.

30. (4pt) Support or refute the following claim: modern code review is useful even for legacy systems whose absolute quality is poor.

31. Consider the following code snippet:

```
1  static boolean checkInBox(int x, int y) {
2      return (this.x < x && x < this.x + 10) && \
3          (this.y < y && y < this.y + 10);
4  }
```

What are two code-level design improvements that you could make to this method? The two improvements cannot both be the same kind of change.

(a) (3pt) _____

(b) (3pt) _____

(c) (3pt) Select one of your answers to the previous question. Give a one-sentence justification for why your change improves the method's code-level design.

32. (4pt) Support or refute the following claim: the distinction between “free software” and “open-source software” is no longer important in modern software engineering.

33. Consider the following pairs of italicized tools, techniques, or processes. For each pair, give a class of defects or a situation for which the first element performs better than the second (i.e., is more likely to succeed and reduce software engineering effort and/or improve software engineering outcomes) and explain both why the better choice performs better and why the worse choice performs worse.
- (a) (3pt) *mutation testing* better than *statement coverage*
- (b) (3pt) *constructive cost model* better than *tshirt size estimation*
34. (4pt) You are a junior software engineer at GitHub, an e-commerce marketplace, on the team that handles payments. Through a series of unfortunate events, you discover that starting tomorrow, you will be the only person on your team who is pageable: everyone else is either sick or on vacation. However, one senior engineer is available this afternoon to help you prepare. What is the most important thing that you can ask your senior colleague to do to help you prepare, and why?
35. (6pt) Support or refute the following claim: choosing test inputs by hand is a form of toil. As part of your argument, you should analyze each of the six aspects of toil that we discussed in class.

36. (12pt) Suppose that you are friends with the following set of people: Alice, Bob, Charlie, Dave, Eve, Faythe, Grace, and Harry. You hold a party, and discover that the night ends in drama when you invite everyone. You decide to hold a series of smaller parties to figure out who doesn't get along with whom. To decide who to invite to each smaller party, you decide to use the delta debugging algorithm that you learned about in CS 490. Suppose further that the ground truth is that any party involving both Alice and Eve and some third participant will end in drama, because Eve will eavesdrop on a conversation between Alice and some counterparty, but Eve is terrible at avoiding detection. Moreover, you cannot attend any of these parties, because as a neutral scientific observer you don't want to disturb the experiment. Execute the delta debugging algorithm to determine the set of parties that you will throw, showing your work at each step. You may abbreviate all of your "friends" by the first letter of their first names in your answer (e.g., you may write "A" for "Alice", "B" for "Bob", etc.).

Table 1: “Your Choice” Readings:

- A. Garlan’s *Software Architecture*
- B. Kellogg et al.’s *Verifying Object Construction*
- C. Kim et al.’s *A Field Study of Refactoring Challenges and Benefits*
- D. Malkawi’s *The Art of Software Systems Development: Reliability, Availability, Maintainability, Performance*
- E. Scully et al.’s *Machine Learning: The High-Interest Credit Card of Technical Debt*
- F. Dean and Barroso’s *The Tail at Scale*
- G. Xu et al.’s *Do Not Blame Users for Misconfiguration*
- H. Terrell et al.’s *Gender Differences and Bias in Open Source: Pull Request Acceptance of Women versus Men*

IV. “Your Choice” Reading Quiz

37. Answer the question below about one of the “Your Choice” readings from the list above.

- (a) Write the letter of the “Your Choice” reading about which you are answering the next part of the question: _____
- (b) (3pt) Write something you learned from the article that is not obvious from reading the title. To receive credit, whatever you answer must be *explicitly discussed* in the article.

V. Document-based Questions (44pts). All questions in this section refer to a documents **A-C**. These documents appear at the end of the exam (I recommend that you tear them out and refer to them as you answer the questions).

Questions on this page and the next refer to **Documents A, B, C, D, and E**.

Suppose that you a software engineer at Amazing Instant Messaging, Ltd. (“AIM”), a small company that provides instant messaging services over the internet. Your main instant messaging product uses the NSS cryptography library from Mozilla to encrypt communications between your users. **Document A** is an excerpt from a blog post describing a security vulnerability in NSS. **Documents B, C, and D** are supporting documents for the excerpt. **Document E** is an email from the CEO of AIM to you about the vulnerability.

38. (6pt) Support or refute the following claim using evidence from the documents: a fuzzer should be able to discover this vulnerability.

39. (6pt) Suppose that this bug existed for a long time (in real life, it existed for at least 9 years before being discovered!), even though the code in question was actively being fuzzed regularly by Google’s `oss-fuzz` project. Give a plausible explanation for why a fuzzer might not have discovered the bug, despite this code being “covered” by the fuzzer.

40. (4pt) Support or refute the following claim: even if none of your customers were impacted, you need to write a postmortem about this bug.

41. (16pt) **Document E** contains an email to you from the CEO of AIM. Write an appropriate, professional email response on the rest of this page.

Questions on this page refer to **Document F**, which contains some simplified (Java) code for a weather monitoring and display system.

42. (4pt) Support or refute the following claim: the `WeatherStation` and `Display` classes are tightly-coupled.
43. (8pt) Describe a refactoring that you could make to this code to de-couple these classes. (Hint: consider using a design pattern.)

VI. Extra Credit. Questions in this section do not count towards the denominator of the exam score.

44. (1pt) In section II (Matching), there is a theme to the names used in the situation descriptions. What is the theme?
45. (1pt) In the engineer panel, one of the engineers interrupted another to make a specific point about postmortems (which they also called “CoE”s, because that’s what Amazon calls them). What was the specific point?
46. (1pt) If you would be willing to serve on an engineer panel in the future, write an email address that you will continue to monitor after you graduate in this space. If you’re unwilling, you can write “no” in this space to receive full credit on this question.
47. (1pt) What was something that surprised you about CS 490?

This page intentionally left blank (you may use it as scratch paper, and the proctor will have more scratch paper at the front if you need more).

Document A

Network Security Services (NSS) is Mozilla's widely used, cross-platform cryptography library. When you verify an ASN.1 encoded digital signature, NSS will create a `VFYContext` (see **Document B**) structure to store the necessary data. This includes things like the public key, the hash algorithm, and the signature itself.

The maximum size signature that this structure can handle is whatever the largest union member is, in this case that's RSA at 2048 bytes. That's 16384 bits, large enough to accommodate signatures from even the most ridiculously oversized keys.

Okay, but what happens if you just....make a signature that's bigger than that?

Well, it turns out the answer is memory corruption. Yes, really.

The untrusted signature is simply copied into this fixed-sized buffer, overwriting adjacent members with arbitrary attacker-controlled data.

The bug is simple to reproduce and affects multiple algorithms. The easiest to demonstrate is RSA-PSS. In fact, just the three commands in **Document C** work!

The actual code that does the corruption varies based on the algorithm; **Document D** is the code for RSA-PSS. The bug is that there is simply no bounds checking at all; `sig` and `key` are arbitrary-length, attacker-controlled blobs, and `cx-u>` is a fixed-size buffer.

This wasn't a process failure, the vendor did everything right. Mozilla has a mature, world-class security team. They pioneered bug bounties, invest in memory safety, fuzzing and test coverage.

NSS was one of the very first projects included with `oss-fuzz`, it was officially supported since at least October 2014. Mozilla also fuzz NSS themselves with `libFuzzer`, and have contributed their own mutator collection and distilled coverage corpus. There is an extensive test suite, and nightly ASAN builds.

- Did Mozilla have good test coverage for the vulnerable areas? YES.
- Did Mozilla/chrome/oss-fuzz have relevant inputs in their fuzz corpus? YES.
- Is there a mutator capable of extending `ASN1_ITEMS`? YES.
- Is this an intra-object overflow, or other form of corruption that ASAN would have difficulty detecting? NO, it's a textbook buffer overflow that ASAN can easily detect.

Document B

The definition of the VFYContext structure from NSS:

```
1 struct VFYContextStr {
2     SEC0idTag hashAlg; /* the hash algorithm */
3     SECKEYPublicKey *key;
4     union {
5         unsigned char buffer[1];
6         unsigned char dsasig[DSA_MAX_SIGNATURE_LEN];
7         unsigned char ecdsasig[2 * MAX_ECKEY_LEN];
8         unsigned char rsasig[(RSA_MAX_MODULUS_BITS + 7) / 8];
9     } u;
10    unsigned int pkcs1RSADigestInfoLen;
11    unsigned char *pkcs1RSADigestInfo;
12    void *wincx;
13    void *hashcx;
14    const SECHashObject *hashobj;
15    SEC0idTag encAlg; /* enc alg */
16    PRBool hasSignature;
17    SECIItem *params;
18 };
```

Document C

Reproducing this vulnerability in three easy commands.

```
1 # We need 16384 bits to fill the buffer, then 32 + 64 + 64 + 64 bits to overflow to hashobj,
2 # which contains function pointers (bigger would work too, but takes longer to generate).
3 $ openssl genpkey -algorithm rsa-pss -pkeyopt rsa_keygen_bits:$((16384 + 32 + 64 + 64 + 64)) -
   ↪ pkeyopt rsa_keygen_primes:5 -out bigsig.key
4 # Generate a self-signed certificate from that key
5 $ openssl req -x509 -new -key bigsig.key -subj "/CN=BigSig" -sha256 -out bigsig.cer
6 # Verify it with NSS...
7 $ vfychain -a bigsig.cer
8 Segmentation fault
```

Document D

The signature size must match the size of the key, but there are no other limitations. `cx-u` is a fixed-size buffer, and `sig` is an arbitrary-length, attacker-controlled blob.

```
1         case rsaPssKey:
2             sigLen = SECKEY_SignatureLen(key);
3             if (sigLen == 0) {
4                 /* error set by SECKEY_SignatureLen */
5                 rv = SECFailure;
6                 break;
7             }
8
9             if (sig->len != sigLen) {
10                PORT_SetError(SEC_ERROR_BAD_SIGNATURE);
11                rv = SECFailure;
12                break;
13            }
14
15            PORT_Memcpy(cx->u.buffer, sig->data, sigLen);
16            break;
```

Document E

Hi engineering team,

I heard about this giant security bug in NSS from someone I follow on Twitter - don't we use that library? I remember you all telling me that one time like three years ago, right? Does this means we've been 100% pwnd? I'm also cc'ing marketing to do damage control if needed. Pls fix asap.

How could you all let me and our valued customers down this way? I trusted you.

Sir Ken Obi, OBE

President & CEO, AIM Ltd.

Sent from my iPhone

Document F

```
1  class WeatherStation {
2      private Display display;
3
4      public WeatherStation(Display display) {
5          this.display = display;
6      }
7
8      public void updateWeather(Weather weather) {
9          display.showWeather(weather);
10     }
11 }
12
13 class Display {
14     public void showWeather(Weather weather) {
15         System.out.println("Current weather: " + weather.toString());
16     }
17 }
```
