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 will never contain any significant words used in the question itself (i.e., "crossword rules"). You are permitted one 8.5x11 inch sheet of paper (double-sided) containing notes; all other aids (other than your brain) are forbidden. Questions may be brought to the instructor.

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 on every page:	1 / 1
	I. Reading Quiz Redux:	5 / 5
	II. Very Short Answer:	24 / 24
Contents (blanks for graders only):	**III. Short Answer:**	40 / 40
	IV. DBQs:	30 / 30
	V. Extra Credit:	5 / 0
	Total:	105 / 100

I. Reading Quiz Redux (5pts)

2. (1pt) **Static Analysis, Part 2: TRUE** or **FALSE**: to use the verifier, engineers were taught how to use a special, declarative programming language that was not similar to their regular development language (C). The author's ICSE paper reports on how easy it was to teach this language to C developers.

3. (1pt) **Debugging, Part 2: TRUE** or **FALSE**: delta debugging requires a test to prove that each circumstance is really failure inducing.

4. (1pt) **Code-level Design:** Name an advantage of `black` over the other Python linters discussed in the Yelp whitepaper. (< 5 words)

   any of: opinionated; resolves errors automatically; consistency

5. (1pt) **Tech Debt, Part 2:** The author claims that most programmers, when asked about the system they're working on, "think the old code is a mess". He posits this is due to a "fundamental law of programming". Which one?

   A reading code is harder than writing code

   B the halting problem

   C given enough eyeballs, all bugs are shallow

6. (1pt) **Software Architecture, Part 1:** The author argues that which of the following should drive the design of a software system's architecture:

   A the existing implementation

   B a set of guidelines from an architecture book

   C the system's quality requirements

II. Multiple Choice and Very Short Answer (24pts). 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.

7. (2pt) A **sound** / **complete** program analysis always answers "I don't know" unless there is definitely a bug in the program being analyzed.

8. (2pt) Which of the following could make a good milestone in a software project? Circle all that apply.
   A the code is "50% done"
   B a user story
   C the end of a particular sprint
   D a particular test passes

9. (2pt) When interviewing for a software engineering role, which of the following are you likely to be evaluated on? Circle all that apply.
   A your actual ability to do the job of a software engineer
   B your knowledge of coding and specific algorithms
   C your niceness and personality
   D how well you are dressed

10. (2pt) When naming a method, it is a best practice to use a verb-like name if and only if the method has **side-effects**

11. (2pt) The **Turing machine** and the **lambda calculus** are two alternative, equivalent formalizations of computability. Match them with the programming paradigms they inspired:
    Imperative: **Turing machine**
    Functional: **lambda calculus**

12. (4pt) Give two advantages of static type systems over dynamic type systems and two advantages of dynamic type systems over static type systems.
    Advantages of static type systems: **early detection of errors**,
    types are documentation
    Advantages of dynamic type systems: **faster prototyping**,
    no false positives

13. (2pt) An engineer 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. This is an example of **mocking**

14. (2pt) You are an engineer at Goggle, a tech-focused search company that helps its users find better eyewear. Your team has been tasked with developing a new microservice that will serve ads to people looking for sunglasses. You and your teammates gather around a whiteboard and draw a diagram of the new service's **architecture**

15. (2pt) When a version control system like `git` determines that two changes are conflict-free, which of the following are possible for the merged code? Circle all that apply.

    A compilation errors

    B multiple changes to the same line

    C test failures

16. (2pt) You are an engineer at UTube, a video-streaming platform. You're having trouble with a particularly difficult programming problem, so you schedule a call with one of your coworkers. Together, you write the difficult code. This is **pair programming**

17. (2pt) Why is exhaustive testing not possible for most programs in practice? Answer in five words or fewer. **Input space is too large.**

III. Short answer (40pts). Answer the questions in this section in at most two sentences.

18. Consider the following code snippet:

```java
    static double computeCircumference(double r) {
      return 2 * 3.14 * r;
    }
```

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) **replace 3.14 with a PI constant**

(b) (3pt) **rename "r" to "radius" or rename "computeCircumference" to "circumference"**

(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.

For 1. above: "avoid magic numbers" or "more self-documenting". For 2. above: "radius" is more descriptive than "r", or "compute" is a verb, but the method returns a noun-like type.

19. Suppose that you are a manager at AmiZone, a French social networking company. Radhia, one of your engineers, comes to you with a proposal to redirect 30% of the effort you currently spend on integration testing to instead run a static analysis.

(a) (3pt) Describe a situation where Radhia's proposal is a good choice for AmiZone: that is, a situation in which it is clearly a better allocation of your limited development resources.

Answers can vary. "AmiZone is concerned about security bugs." is the shortest, best answer; short answers that identify a class of bugs that static analysis is good at will get full credit..

(b) (4pt) Describe, in one sentence each, two risks associated with Radhia's proposal.

Any two of the following sentences would be a good answer (other answers are possible). Reducing integration testing may increase the risk that bugs at the boundary between modules are not detected. Static analysis may produce too many false positive alarms. Static analysis may be unsound or may miss bugs (even of the kind it is supposed to find). Static analysis errors may be too hard to understand.

20. Consider the following items from the "Joel Test", which we read before lecture 2:

    - **A**: Do you use source control?
    - **B**: Can you make a build in one step?
    - **C**: Do you have a bug database?
    - **D**: Do you fix bugs before writing new code?
    - **E**: Do you have an up-to-date schedule?

    (a) (2pt) Identify one of the above Joel Test items **A-E** that your group project did better than the other items. Explain why you were better at this item than the others and how it benefited your project.

    Answers vary.

    (b) (2pt) Identify one of the above Joel Test items **A-E** that your group project did worse than the other items. Explain, via specific examples, how failing to meet this Joel Test requirement impacted your project.

    Answers vary.

21. Consider the following pairs of 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 why.

    (a) (3pt) Agile development model better than waterfall development model

    Agile is best when requirements are not fully known in advance and the customer is easy to access, because it relies on fast feedback cycles between your prototypes and the customer.

    (b) (3pt) Fuzzing better than regression testing

    Fuzzing is better if the existing test suite is very small. Another reasonable answer is that fuzzing can find new bugs, but regression testing only prevents bug that have occurred before.

    (c) (3pt) Interpreted languages better than compiled languages

    Interpreted languages are better for rapid prototyping and are usually easier to write code in.

22. You are the customer asking a software company to build you a program that sorts a list of numbers. You both agree that the output of sort must be in descending order. The company delivers a program that always returns the empty list. You meet to correct this. The company then delivers a program that deterministically tries all permutations of the input list until one is found that is sorted. (This is sometimes called "bogosort".) Explain two elements that may have gone wrong during requirements elicitation (at any stage).

(a) (3pt) The first element:

There was an omission in the functional requirements: the customer failed to state that the returned list must have the same elements as the input list.

(b) (3pt) The second element:

Quality requirements were not discussed: bogosort is O(n!), which is much slower than sorting a list ought to be (O(nlogn)). Other well-reasoned answers are also possible.

23. Consider a program with six sequential if statements that accepts six boolean inputs. Assuming each condition evaluates a single unique input, what is the minimum number of test cases required to achieve:

(a) (1pt) 100% branch coverage? **2: one all false, one all true**

(b) (1pt) 100% condition coverage? **12 = 2*6**

24. (3pt) Support or refute the claim "You cannot have continuous integration without hermetic builds." Answer in three sentences or fewer.

Likely support. Continuous integration does not work well without hermetic builds, because to test in CI you need to be able to build from scratch.

IV. Document-based Questions (30pts). All questions in this section refer to a documents **A-B**. 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 concern **Document A**.

25. (2pt) What kind of document is **Document A**?
    postmortem

26. (3pt) What was the root cause of the outage? Answer in the form of a quote from the document.
    "one of the inputs to the command was entered incorrectly"

27. (3pt) Does the document identify who was responsible for the outage? If so, give a quote from the document identifying that the responsible party. If not, give a one-sentence explanation for why not.
    The document does not identify who was responsible, because this is a blameless postmortem: it blames the system, rather than the individual. (Any answer with "blameless" or a synonym will probably be accepted.)

28. (3pt) The root cause of the incident only impacted a single service (S3 itself). Why was the "blast radius" of the incident (i.e., the number of impacted customers) so large?
    Other services relied on S3, so a cascading failure caused them to fail, as well.

29. (3pt) Describe one change to the tools used to remove capacity that the S3 team could adopt to prevent or mitigate a similar outage in the future.
    Either of the following two answers, which S3 itself gives in the full postmortem, is acceptable. Other sensible answers may also get credit or partial credit. 1) modify the tool to remove capacity more slowly, or 2) add safeguards to prevent capacity from being removed when it will take any subsystem below its minimum required capacity level

30. (3pt) Describe another distinct change to the tools used to remove capacity that the S3 team could adopt to prevent or mitiage a similar outage in the future.
    See the previous question.

Questions on this page refer to **Document B**, which is a candidate's response to you, the interviewer, presenting them with the following technical challenge:

"Write isPalindrome(), a function that returns true if parameter x is a palindrome integer. Note that an integer, like 12321, is a palindrome if it reads the same forwards and backwards."

Document B also includes two questions that the candidate asked you (and your answers) on its first two lines.

31. (2pt) Identify two test inputs where the provided `isPalidrome()` implementation returns **`false`**.

    Answers will vary. Student solutions must not be a palindrome. 123, 10 are potential answers that return false.

32. (3pt) Your answers to question 31 could be which of the following kinds of tests (circle all that apply):

    [✓ **A**] unit tests

    B integration tests

    C fuzz tests

    [✓ **D**] partition tests

33. (4pt) Identify four things that the candidate did well. (In other words, identify four properties that a company might desire in a software engineer that could potentially be shown by a candidate taking the interview and that were shown by this particular candidate.) Use at most 4 sentences.

    Answers will vary. Potential solutions include the following: 1. The candidate provided inline comments explaining some of their code. 2. The candidate asked relevant questions regarding code functionality. 3. The candidate has consistent indentation. 4. The candidate used a descriptive variable name.

34. (4pt) Support or refute the claim that the candidate's implementation of `isPalindrome()` is functionally correct. Use at most 4 sentences.

    Refute. The candidate's implementation of isPalindrome() is not functionally correct. The candidate missed a base case regarding x being a negative number. Negative integers would not count as a palindrome. -313 reversed is 313-.

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

35. (1pt) TODO: this question will be about a specific point made by one of our panelists **TODO**

36. (1pt) TODO: this question will be about a different, specific point made by one of our panelists **TODO**

37. (1pt) Name an optional reading assignment that you read but will not and have not used as a response to the "Optional Reading Response #1" or "Optional Reading Response #2" assignments on Canvas, and then give a one-sentence description of something you learned from that reading. (Note: answering this question with a particular reading means you cannot use that reading for your "Response #2", due on May 2.) **Answers vary.**

38. (1pt) Name another optional reading assignment, distinct from your answer to question 37, that you read but will not and have not used as a response to the "Optional Reading Response #1" or "Optional Reading Response #2" assignments on Canvas, and then give a one-sentence description of something you learned from that reading. (Note: answering this question with a particular reading means you cannot use that reading for your "Response #2", due on May 2.) **Answers vary.**

39. (1pt) Name a recent "hot topic" in Software Engineering research and give a one-sentence description of that work. **Expected answers are the topics covered in class on 20 April.**

Document A:

We'd like to give you some additional information about the service disruption that occurred in the Northern Virginia (US-EAST-1) Region on the morning of February 28th, 2017. The Amazon Simple Storage Service (S3) team was debugging an issue causing the S3 billing system to progress more slowly than expected. At 9:37AM PST, an authorized S3 team member using an established playbook executed a command which was intended to remove a small number of servers for one of the S3 subsystems that is used by the S3 billing process. Unfortunately, one of the inputs to the command was entered incorrectly and a larger set of servers was removed than intended. The servers that were inadvertently removed supported two other S3 subsystems. One of these subsystems, the index subsystem, manages the metadata and location information of all S3 objects in the region. This subsystem is necessary to serve all GET, LIST, PUT, and DELETE requests. The second subsystem, the placement subsystem, manages allocation of new storage and requires the index subsystem to be functioning properly to correctly operate. The placement subsystem is used during PUT requests to allocate storage for new objects. Removing a significant portion of the capacity caused each of these systems to require a full restart. While these subsystems were being restarted, S3 was unable to service requests. Other AWS services in the US-EAST-1 Region that rely on S3 for storage, including the S3 console, Amazon Elastic Compute Cloud (EC2) new instance launches, Amazon Elastic Block Store (EBS) volumes (when data was needed from a S3 snapshot), and AWS Lambda were also impacted while the S3 APIs were unavailable.

S3 subsystems are designed to support the removal or failure of significant capacity with little or no customer impact. We build our systems with the assumption that things will occasionally fail, and we rely on the ability to remove and replace capacity as one of our core operational processes. While this is an operation that we have relied on to maintain our systems since the launch of S3, we have not completely restarted the index subsystem or the placement subsystem in our larger regions for many years. S3 has experienced massive growth over the last several years and the process of restarting these services and running the necessary safety checks to validate the integrity of the metadata took longer than expected. The index subsystem was the first of the two affected subsystems that needed to be restarted. By 12:26PM PST, the index subsystem had activated enough capacity to begin servicing S3 GET, LIST, and DELETE requests. By 1:18PM PST, the index subsystem was fully recovered and GET, LIST, and DELETE APIs were functioning normally. The S3 PUT API also required the placement subsystem. The placement subsystem began recovery when the index subsystem was functional and finished recovery at 1:54PM PST. At this point, S3 was operating normally. Other AWS services that were impacted by this event began recovering. Some of these services had accumulated a backlog of work during the S3 disruption and required additional time to fully recover.

From the beginning of this event until 11:37AM PST, we were unable to update the individual services' status on the AWS Service Health Dashboard (SHD) because of a dependency the SHD administration console has on Amazon S3. Instead, we used the AWS Twitter feed (@AWSCloud) and SHD banner text to communicate status until we were able to update the individual services' status on the SHD. We understand that the SHD provides important visibility to our customers during operational events and we have changed the SHD administration console to run across multiple AWS regions.

Finally, we want to apologize for the impact this event caused for our customers. While we are proud of our long track record of availability with Amazon S3, we know how critical this service is to our customers, their applications and end users, and their businesses. We will do everything we can to learn from this event and use it to improve our availability even further.

Document B:

```
// Q: Can integer x be in range [0-9]? A: Yes.
// Q: Should I account for integer overflow? A: Yes.
bool isPalindrome(int x) {
 // a single digit is a palindrome
 if (x < 10) {
   return true;
 }
 // if x’s last digit is 0, then its first digit must be 0 in order
 // to be a palindrome. In this case, only 0 can be a palindrome.
 if (x % 10 == 0 && x != 0) {
   return false;
 }
 // revert the last half of x for comparison against first half
 int revertedNumber = 0;
 while(x > revertedNumber) {
   revertedNumber = revertedNumber * 10 + x % 10;
   x /= 10;
 }
 return x == revertedNumber || x == revertedNumber / 10;
}
```