FIRST QUIZ: CS4472A Tuesday, 3 October 2017, 7:10pm, Room MC17

NAME AS APPEARS ON STUDENT ID:

STUDENT ID NUMBER:

UWO/CONFLUENCE USER NAME:

REMINDERS:

1. (from course outline) The quiz will be closed book, closed notes, with no electronic devices allowed, with particular reference to any electronic devices that are capable of communication and/or storing information.
2. Write neatly. If the marker can't read it, it is wrong.
3. This exam shouldn't take long to write. On the other hand, time will pass. It is a 30 minute quiz with 20 questions. If you complete a question every minute you will still have 10 minutes at the end to double check that everything is in order.
4. While you are not allowed to open the exam booklet until the proctor says you can, you can fill out the information on the cover page. You should also get out your student id and make sure your pencils and pens are in order. If you need to get something out of your jacket or knapsack once the exam has started, raise your hand and wait til a proctor comes to you to oversee the matter.

1. MicroTest (MiniTest subset) uses public_instance_methods to ANSWER
   ANSWER=

2. The motivation behind multiple merges per day per developer is to ANSWER
   ANSWER=

3. The corporate policy of developers merging their working copies into the main line of the branch repository several times a day is called ANSWER
   ANSWER=

4. The D in SOLID stands for ANSWER
   ANSWER=

5. One study of 198 user major failure reports on 5 widely used distributed systems found that nearly all failures were caused by coding mistakes in ANSWER
   ANSWER=

6. The four phases of testing (according to Whittaker) are: 1) modeling the software environment, 2) selecting test cases, 3) running and checking test cases, and 4) ANSWER
   ANSWER=

7. Many of the ideas of the Capability Maturity Model were adapted to individual developers under the name ANSWER
   ANSWER=

8. One study of 198 user major failure reports on 5 widely used distributed systems found statement coverage testing could have caught nearly ANSWER of the causes.
   ANSWER=

9. While the notation looks odd, in RSpec, it is actually implemented in Ruby as an ANSWER
   ANSWER=

10. Using combinatorial testing, if I have 10 binary inputs, I only need to use ANSWER test cases (each a setting of each of the 10 inputs) to expect to find 98 per cent of the errors in the program.
    ANSWER=

11. The Capability Maturity Model for US government contractors distinguishes 5 levels of company software development process. Level 5 is characterized as ANSWER
    ANSWER=

12. The paper Orthogonal defect classification-a concept for in-process measurements was an example of people at IBM analyzing records of defects in order to ANSWER
    ANSWER=

13. It is easy to make up test inputs, but it can be tricky to know what the right output for a given input should be. This is refered to as the ANSWER problem
    ANSWER=

14. In the Testing Maturity Model, at Level 5, we aim at ANSWER rather than defect detection
    ANSWER=

15. One study of 100 large open source Java programs compared better code coverage with number of post-release defect reports and found ANSWER
    ANSWER=

16. A study by Ahmed et al found that the probability of errors in untested code was ANSWER the probability of errors in tested code
    ANSWER=

17. The scripts that were designed to aid the practice process assume that you will be uploading a copy of your work to BitBucket every time you ANSWER
ANSWER=

18. The pattern where you create an object whose job is to create other objects (rather than using new to create other objects) is called ANSWER
ANSWER=

19. MicroTest (MiniTest subset)'s usage pattern is for the test class to inherit from Test so that Class.inherited can be used to ANSWER
ANSWER=

20. Testing is generally about finding errors that have already been made. This course also covers the topic of ANSWER, which is about trying to prevent errors from being made in the first place.
ANSWER=

21. The total amount of practice time you can get credit for during a practice week is ANSWER
ANSWER=

22. According to Robert Martin who first promoted the SOLID methodology, the S doesn't refer to functions, but to ANSWER
ANSWER=

23. RSpec and Cucumber are tools designed to support the ANSWER style of software development
ANSWER=

24. The Capability Maturity Model for US government contractors distinguishes 5 levels of company software development process. Level 1 is characterized as ANSWER
ANSWER=

25. The practice technique advocated in this class is a modification of the ANSWER
ANSWER=

26. When multiple methods of a class have the same parameters, this is a code smell called ANSWER
ANSWER=

27. The protocols for practice expect that the longest amount of time that you will practice before recording a note is ANSWER
ANSWER=

28. When multiple methods of a class have the same parameters, that generally indicates that those parameters should ANSWER
ANSWER=

29. The per cent of the total mark allocated for all the weekly practices is ANSWER
ANSWER=

30. The number of quizzes CS4472 will have this semester is ANSWER
ANSWER=

31. When I say that in RSpec, expect x.to eq y, eq an object that inherits from ANSWER, meeting the requirements of to
ANSWER=

32. Modified condition/decision coverage is often a requirement (regulatory or contractual) in ANSWER
ANSWER=

33. To illustrate the relation between testing and software design, we will look at the programming technique ANSWER
ANSWER=

34. The number of weekly practices that CS4472 will have this semester is ANSWER
ANSWER=

35. The S in Solid stands for ANSWER
ANSWER=

36. A main theme behind the practice technique advocated in this class is that in order to improve your programming, ANSWER
ANSWER=

37. Structural testing is another name for ANSWER
ANSWER=

38. The differences between RSpec and Cucumber result from the intent that RSpec test files are meant to be readable by ANSWER
ANSWER=

39. The O in SOLID stands for ANSWER
ANSWER=

40. According to Michael Feathers, code that is difficult to test is ANSWER
ANSWER=

41. Unlike MiniTest which is implemented as a class library, RSpec is implemented in Ruby as an ANSWER
ANSWER=

42. Although we often think of programs as taking inputs and producing outputs, a higher level view of what is going on is to think of the programs as ANSWER about how to take inputs and produce outputs.
ANSWER=

43. An important concept we will look at related to the question of when has one done enough testing is ANSWER
ANSWER=

44. MicroTest (MiniTest subset) discourages the writing of tests that depend on side-effects of the previous test by ANSWER
ANSWER=

45. The four requirements of MC/DC are: 1) each entry and exit point is invoked, 2) each decision takes every possible outcome, 3) each condition in a decision takes every possible outcome, and 4) ANSWER
ANSWER=

46. The kind of testing we do to make sure that when we change a program we do not break something that used to work is called ANSWER
ANSWER=

47. The I in SOLID stands for ANSWER
ANSWER=

48. The first testing framework for Ruby that we are looking at is called ANSWER
ANSWER=

49. Once RSpec has created a test class, it fills in its definition by executing the Ruby method ANSWER
ANSWER=

50. The ANSWER is a method developed by Watt S. Humphrey to help individuals improve their programming skills based on existing methods that had been developed to help organizations improve their product development capabilities.
ANSWER=

51. The per cent of the total mark allocated for all the practice reviews is ANSWER
ANSWER=

52. The TDD Cycle is ANSWER
ANSWER=

53. A common piece of information for people interested in programmer productivity to track is ANSWER
ANSWER=

54. The testing technique called boundary value partition starts with the notion of breaking the space of inputs into ANSWER
ANSWER=

55. In MiniTest, we write test classes that inherit from Test, but in RSpec these test classes are actually being created at runtime by ANSWER
ANSWER=

56. The per cent of the total mark allocated for all the quizzes is ANSWER
ANSWER=

57. The L in SOLID stands for ANSWER
ANSWER=

58. RSpec specifications are sometimes called ANSWER documentation
ANSWER=

59. The differences between RSpec and Cucumber result from the intent that Cucumber test files are meant to be readable by ANSWER
ANSWER=

60. The first tool for checking code quality for programs written in Ruby is ANSWER, which is described as a code smell detector.
ANSWER=

61. The number of practice reviews that CS4472 will have this semester is ANSWER
ANSWER=

```
exam_database_file= examdatabase.json
exam_format= latex
dump_database= false
line_width= 72
question_count= 173
create_exam= false
answer_key= true
sample_seed= 222
shuffle_seed= 2345
["C1", "C2", "C3", "C4", "C5", "C6"]
["C1", "C2", "C3", "C4", "C5", "C6"]
```

1. find methods that begin test_
2. minimize merge conflicts
3. continuous integration
4. dependency inversion principle
5. the error handling code
6. checking how well the testing is going
7. Personal Software Process
8. - a quarter
   - 25 per cent
   - 23 per cent
9. method
10. 13
11. continually improving
12. improve their process
13. Oracle
14. defect prevention
15. no connection
16. twice
17. record a note about your practice progress
18. the factory pattern
19. get a list of test classes
20. quality assurance
21. 3 hours
22. roles in the business that uses the software
23. - Behavior-driven development
    - BDD

24. - chaotic
    - ad hoc
25. Personal Software Process
26. data clumping
27. 30 minutes
28. be put into a class of their own
29. 30
30. 3
31. Matcher
32. - safety-critical applications
    - avonic systems
    - automotive systems
33. test driven development
34. 10
35. single responsibility principle
36. you need data about your past programming
37. - code-based testing
    - white-box testing
38. just the programmer
39. open/closed principle
40. poorly designed
41. - domain-specific language
    - DSL
42. encode knowledge
43. - coverage
    - mutation
44. running tests in random order
45. each condition in a decision is shown to independently affect the outcome of a decision
46. regression testing
47. interface segregation principle
48. minitest
49. module_exec
50. Personal Software Process
51. 49

52. Red Green Refactor
53.
    - time spent
    - number of lines of code written
    - number of defects found
54. regions of interest
55. describe
56. 21
57. Liskov substition principle
58. executable
59. the customer and the programmer
60. reek
61. 4