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

   - test driven development

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

   - coverage
   - mutation

3. 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

   - Oracle

4. 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.

   - quality assurance

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

   - minitest

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

   - reek

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

   - 3

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

   - 10

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

   - 4

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

    - 21

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

    - 30

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

    - 49

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

    - Personal Software Process

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

    - you need data about your past programming

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

    - time spent
    - number of lines of code written
    - number of defects found

16. 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.

    - encode knowledge

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

    - record a note about your practice progress

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

    - 30 minutes

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

    - 3 hours

20. 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

    - checking how well the testing is going

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

    - regions of interest

22. Structural testing is another name for ANSWER

    - code-based testing
    - white-box testing

23. 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

    - regression testing

24. 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.

    - 13

25. 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.

    - Personal Software Process

26. 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

- improve their process

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

    - data clumping

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

    - be put into a class of their own

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

    - running tests in random order

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

    - get a list of test classes

31. MicroTest (MiniTest subset) uses public_instance_methods to ANSWER

    - find methods that begin test_

32. The TDD Cycle is ANSWER

    - Red Green Refactor

33. RSpec specifications are sometimes called ANSWER documentation

    - executable

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

    - domain-specific language
    - DSL

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

    - Behavior-driven development
    - BDD

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

    - the customer and the programmer

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

    - just the programmer

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

    - chaotic
    - ad hoc

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

- continually improving

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

    - Personal Software Process

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

    - defect prevention

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

    - continuous integration

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

    - minimize merge conflicts

44. 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

    - the factory pattern

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

    - method

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

    - describe

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

    - module_exec

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

    - Matcher

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

    - safety-critical applications
    - avonic systems
    - automotive systems

50. 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

    - each condition in a decision is shown to independently affect the outcome of a decision

51. 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.

    - a quarter
    - 25 per cent
    - 23 per cent

52. 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

   - the error handling code

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

   - no connection

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

   - twice

55. The S in Solid stands for ANSWER

   - single responsibility principle

56. The O in SOLID stands for ANSWER

   - open/closed principle

57. The L in SOLID stands for ANSWER

   - Liskov substition principle

58. The I in SOLID stands for ANSWER

   - interface segregation principle

59. The D in SOLID stands for ANSWER

   - dependency inversion principle

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

   - roles in the business that uses the software

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

   - poorly designed