THIRD QUIZ: CS4472A Tuesday, 5 December 2017, 7:10 pm, 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. The Capability Maturity Model for US government contractors distinguishes 5 levels of company software development process. Level 1 is characterized as ANSWER
   ANSWER=

2. Sandi Metz's approach to unit testing of objects also distinguishes two types of messages: command and ANSWER
   ANSWER=

3. In defect causal analysis, the information you are trying to extract from the problem reports in order to classify them are: when the defect that caused the problem was inserted into the software; when the problem was detected; and ANSWER
   ANSWER=

4. In Confident Code, Avi Grimm presented techniques for refactoring code to improve its readability, i.e., narrative structure. One approach is to replace usages of nil with a special ANSWER that responds to messages by just returning itself. This means method invocations can be pipelined with constant checks that the previous stage didn't produce a nil.
   ANSWER=

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

6. The version of Agile that we looked at in class and that lead to the development of cucumber envisions interacting with the customer in a structured way through the discussion of ANSWER
   ANSWER=

7. When considering mocking for tests, we should consider the reason for the test itself. For example, if we are doing acceptance tests for a customer, then mocking is ANSWER
   ANSWER=

8. An important assumption in testing is that although any particular program failure may result from a complex sequence of events, there is generally a first event that if it hadn't gone wrong the failure wouldn't have occurred. Thus it is sufficient to test for these simple event failures rather than having to build tests targetted at complicated interaction failures. This effect is called ANSWER
   ANSWER=

9. When using the Ruby DSL for dynamic types, one can use builtin types like Num, but one can also define one's own types, like Even that requires a value of that type to be an even number. To do this, we create a class Even that contains the method ANSWER, which checks to see if its input parameteris even.
   ANSWER=

10. In our Agile approach to determining what tasks we are to do for the customer next, the high level information that we want from the customer is what is the anticipated role in the organization of the person who would do the task, what is it that they want to do, and ANSWER
    ANSWER=

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

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

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

14. Structural testing is another name for ANSWER
    ANSWER=

15. 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=

16. 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=

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

18. 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=

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

20. 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=

```
exam_database_file= examdatabase.json
exam_format= latex
dump_database= false
line_width= 72
question_count= 20
create_exam= false
answer_key= true
sample_seed= 2322
shuffle_seed= 245
["C1", "C10", "C2", "C3", "C4", "C5", "C6", "C7", "C8", "C9"]
["C1", "C10", "C2", "C3", "C4", "C5", "C6", "C7", "C8", "C9"]
```

1. - chaotic
   - ad hoc
2. query
3. - what type of mistake was made
   - what type of defect was introduced
4. Null Object
5. 21
6. user stories
7. not appropriate
8. the coupling effect
9. valid?
10. - what benefit they expect to get from the task
    - what reward they expect to get from the task
11. 4
12. running tests in random order
13. 30 minutes
14. - code-based testing
    - white-box testing
15. continuous integration
16. improve their process
17. Personal Software Process
18. twice
19. poorly designed
20. regression testing