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

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

3. 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=

4. 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=

5. 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=

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

7. 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=

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

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

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

11. 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=

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

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

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

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

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

17. Structural testing is another name for ANSWER
    ANSWER=

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

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

20. 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=

```
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= 98247
shuffle_seed= 1245
["C1", "C2", "C3"]
["C1", "C2", "C3"]
```

1. you need data about your past programming
2. 10
3. regression testing
4. checking how well the testing is going
5. record a note about your practice progress
6. regions of interest
7. encode knowledge
8. - time spent
   - number of lines of code written
   - number of defects found
9. test driven development
10. 21
11. 13
12. 3
13. 4
14. Personal Software Process
15. reek
16. 3 hours
17. - code-based testing
    - white-box testing
18. 30 minutes
19. 49
20. Personal Software Process