University of Illinois at Urbana-Champaign
Department of Computer Science

No Final Exam

CS 427: Software Engineering I
Fall 2015

December 16, 2015

TIME LIMIT = 3 Hours
COVER PAGE + 2 PAGES

Write your name and netid neatly in the space provided below; **write your netid** in the upper right corner of **every page**.

Name: ______________________________
Netid: ______________________________

This is a closed book, closed notes examination. You may not use calculators or any other electronic devices. Any sort of cheating on the examination will result in a zero grade.

We cannot give any clarifications about the exam questions during the test. If you are unsure of the meaning of a specific question, write down your assumptions and proceed to answer the question on that basis.

Do all the problems in this booklet. Do your work inside this booklet, using the backs of pages if needed. The problems are of varying degrees of difficulty so please pace yourself carefully, and answer the questions in the order which best suits you. **Answers to essay-type questions should be as brief as possible.** If the grader cannot understand your handwriting, you will get 0 points.

There are 5 questions on this exam and the maximum grade on this exam is 9 points.

Page	Points	Score
1	5	
Total:	5	

Page	Points	Score
2	4	
Total:	4	

1. Regression Testing

2 (a) Regression testing is commonly performed in practice.

i. Based on the slides, what is one reason why performing regression testing is good?

ii. Based on the slides, what is one reason why performing regression testing can be bad?

2. Testing And Coverage

2 (a) There are situations in which achieving coverage is infeasible. Construct a small Java example showing a statement that cannot be covered.

3. Debugging and Reverse Engineering

1 (a) Debugging activities include tracing, backtracking, and cause elimination. Describe one way to perform one of these activities.

4. Requirements Engineering and Risk Management

2 (a) The lectures slides on **risk management** list several risk items that students may face in course project. List **two** of them.

2 5. Rep Exposure

In MP3 code, we had a private field that you could not easily access, something like this:

```
public class Collection {
  private List elements; // allow arbitrary Objects in the list
  public Collection(List elements) { this.elements = elements; }
}
```

One could make that field public, but it is a bad design, because some remote class could change the field. A seemingly good design is to have a getter that returns the field value:

```
public class Collection {
  private List elements; // allow arbitrary Objects in the list
  public Collection(List elements) { this.elements = elements; }
  List getElements() { return elements; }
}
```

However, this exact implementation suffers from what is called "rep(resentation) exposure" because it returns an internal object and allows someone remote to change it. A better implementation may be to return a copy of "elements", e.g.:

```
public class Collection {
  private List elements; // allow arbitrary Objects in the list
  public Collection(List elements) { this.elements = elements; }
  List getElements() { return new ArrayList<>(elements); }
}
```

Write a JUnit test to check whether the class (with `getElements`) suffers from "rep exposure".

```
@Test public void checkRepExposure() {
// TODO: Create some collection(s) and assert something about "getElements"









}
```