

NAME: _____

SAMPLE FINAL EXAM
CS 3398, Spring or Fall 20XX
Maximum Points: 100

Multiple choice questions (34 pts): circle the letter of the **best** answer:

1. **(2 pts)** Performance optimizations often reduce which of the following desired characteristic of a good implementation?
 - (a) Maintainability
 - (b) Traceability
 - (c) Correctness
 - (d) Completeness

2. **(2 pts)** Which model might be included in the Software Requirements Specification, in the Overall Description section (2.1) to indicate the system boundaries and give a general idea of what the system will and will not do?
 - (a) UML Class diagram.
 - (b) Entity-Relationship (ER) diagram
 - (c) UML State diagram.
 - (d) Simple context diagram.

3. **(2 pts)** What is software reengineering?
 - (a) Re-constructing the original source code from the machine code program.
 - (b) Translating a program to a new source code language.
 - (c) Scrapping the source code of a program and re-writing it entirely from scratch.
 - (d) Re-structuring or re-writing part or all of a software system without changing its functionality.

4. **(2 pts)** This is the stage of object oriented design which has the goal of mapping the object model (a class diagram) to code.
 - (a) Object Oriented Analysis
 - (b) Object design
 - (c) System design
 - (d) Implementation

5. **(2 pts)** A set of input values and expected results that exercises a system with the purpose of detecting faults is called a
 - (a) Unit Test
 - (b) Test case
 - (c) Test stub
 - (d) Test driver

(17 multiple choice questions in all, these 5 are fairly easy ones)

Problems and written answer questions (66 pts): Write your answers on the blank papers provided. Number your answers and write your name on the papers! You may use both sides.

1. **(12 pts)** Draw a control flow diagram for the following source code: Then:
 - assign a number to each node.
 - list the unique paths through the control flow diagram (use a sequence of node numbers to identify each path).
 - for each unique path, generate a test case by giving values to use for a, b, and c to make the program run through that path. Also give the expected result (the value of d).

```
d = 0;
if (a > b)
    d = d+1;
if (b > c)
    d = d+2;
cout << d << endl;
```

2. **(10 pts)** Given the UML class diagram on the next page decompose it into a set of components (subsystems) that has at least 2 (but no more than 4) components. Use the goals of **loose coupling** and **high cohesion** (and any other heuristics that may apply) for deciding which classes should go in each component.
3. **(12 pts)** Draw a class diagram showing the structure of data about employees of a given company. The employees attributes include name, street address, city, state, zip, and an id number. The employees also have an annual salary. Departments have names and are composed of a collection of employees, but each employee can be in only one department. Employees work on one or more projects, which also have names. Projects may have multiple employees assigned to them. Include multiplicity, and attributes in your diagram.
4. **(10 pts)** List and describe at least three techniques for deriving test cases (for any kind of software testing). Your answer should have at least three sentences (at least one for each technique).
5. **(10 pts)** List and describe (using at least one sentence each) the three types of software maintenance. Which one of the three typically gets over half (65%) of the maintenance effort? (in other words, more time is spent on this type of maintenance). Why?
6. **(12 pts)** Describe three different scenarios that occur in source code when one should apply refactoring. The author of the “Refactoring” book calls these “bad smells” in code. For each scenario, describe it in (at least) one sentence, and write another to explain how the bad smell should be “fixed” (refactored).

