Student Name: _____ Student ID # _____

UOSA Statement of Academic Integrity

On my honor I affirm that I have neither given nor received inappropriate aid in the completion of this exercise.

Signature: _____ Date: _____

Notes Regarding this Examination

Open Book(s) You may consult any printed textbooks in your immediate possession during the course of this examination.

Open Notes You may consult any printed notes in your immediate possession during the course of this examination.

No Electronic Devices Permitted You may not use any electronic devices during the course of this examination, including but not limited to calculators, computers, and cellular phones. All electronic devices in the student's possession must be turned off and placed out of sight (for example, in the student's own pocket or backpack) for the duration of the examination.

Violations Copying another's work, or possession of electronic computing or communication devices in the testing area, is cheating and grounds for penalties in accordance with school policies.

Question 1: Object-Oriented Design (20 points)

A. What visibility modifier is typical for the accessor methods of a class? *Why?*

B. *Explain* a situation in which a good design might use a visibility modifier other than what is typical (which you listed above in Part A).

C. If I want to enforce encapsulation of a data field within a class I am designing, and that data field is a reference to an object, would it be better for an accessor method for that data field to return the reference to the object, a shallow clone of the object, or a deep clone of the object? *Why?*

Question 2: More Object-Oriented Design (30 points)

Kyle wants software to keep track of universities such as the University of Oklahoma. A university is a large academic organization composed of smaller academic organizations called "colleges," such the College of Engineering, the College of Education, and the College of Arts & Sciences. The colleges, in turn, are composed of yet smaller academic organizations called "schools" or "departments," such as the School of Computer Science and the Department of Biology. Each of these academic organizations has administrators, students, professors, and a budget.

Note that administrators, students, and professors can belong to more than one academic organization. (For example, a professor might be a professor in both Computer Science and Biology.) However, each smaller academic organization can only belong to one academic organization and it must be of the next size up. (For example, the School of Computer Science cannot belong to both the College of Engineering and the College of Education. Also, the School of Computer Science cannot directly belong to the University of Oklahoma, although it could belong to the College of Engineering which belongs to the University of Oklahoma.)

Administrators, students, and professors are, of course, people. People have names. In addition, administrators and professors are employees with titles and salaries. Administrators have duties while professors have classes. Students have classes and majors. Administrators, students, and professors all have university ID numbers.

A. Draw a simplified UML class diagram that shows appropriate classes and/or interfaces to handle the types of objects described above. In this simplified UML, you do not need to include methods or the types for variables. However, class, interface, and variable names should be included along with accessibility modifiers for the variables and indications of whether each class is concrete or abstract. Also be sure to indicate in the diagram where the listed information is stored and the relationships between the classes and/or interfaces.

[Additional space for UML for Question 2, Part A.]

B. If you used inheritance anywhere in your UML for Part A above, describe where you used it in your design and *explain why* using inheritance improves this design. If you did not use inheritance anywhere in your UML for this design, *explain* a situation in which using inheritance would improve OO design.

C. If you used composition or aggregation anywhere in your UML for Part A above, describe where you used it in your design and *explain why* using composition or aggregation improves this design. If you did not use composition or aggregation anywhere in your UML for this design, *explain* a situation in which using composition or aggregation would improve OO design. In answering this question, be sure to distinguish between composition and aggregation.

Question 3: Polymorphism (5 points)

Explain a kind of polymorphism that does not require inheritance.

Question 4: Unit Testing (15 points)

Explain three units tests you should perform on a `compare` method for a class that implements `java.util.Comparator`. For each test, say what you are testing for and describe how you would go about constructing the test. (You may describe the steps in English, pseudo-code, or actual Java code, as you see fit.)

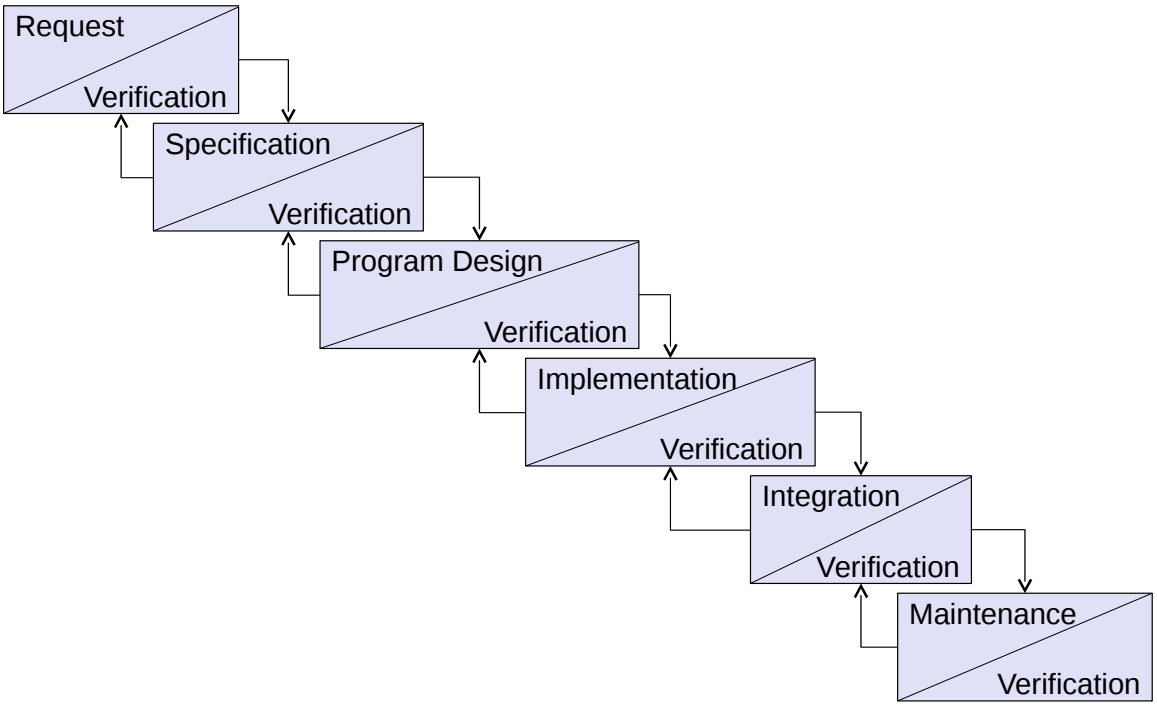
A. One test.

B. A second test.

C. A third test.

Question 5: Object Oriented Software Development (10 points)

Given the following version of the waterfall model of the software lifecycle, *explain* which step(s) should involve UML diagrams.



Question 6: Generics (10 points)

Consider the stub code below for two classes.

```
public class SomeClass {
    public Object someMethod(Object aParam, Object anotherParam){
        return null;
    }
}


public class AnotherClass<T> {
    public T anotherMethod(T aParam, T anotherParam){
        return null;
    }
}
```

Explain one advantage that the Generic class will have over the non-Generic class, once they are implemented.

Question 7: Java Collections Framework (10 points)

Currently in the Java Collections Framework, the Set, List, and Queue classes all descend from the `Collection` interface but the Map classes do not. Instead, the Map classes all descend from the `Map` interface. *Explain* whether it would be a good idea to have the Map classes descend from `Collection`.