

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 standard for data fields of a class? *Why?*

B. *Explain* a situation in which a good design might use a visibility modifier other than the standard (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: Object-Oriented Design Redux (40 points)

Aseem is very interested in rain and other forms of precipitation. He wants software that will inform him any time it rains, snows, sleet, or hails. For rain, snow, and sleet, he wants to know the type and amount of precipitation (in inches) as well as the maximum rate at which it fell. For hail, he wants to know the diameter of the largest hailstones in the storm (in inches). In addition, for frozen precipitation, Aseem wants to know if it is accumulating or melting off.

Aseem is also interested in weather systems containing precipitation. For each weather system, he wants to know what type(s) of precipitation it contains, its locations, its start time, its duration, and its maximum, minimum, and mean temperature.

Finally, Aseem wants to ensure that all of this data is persistent, in case one of the storms knocks out the computer running his software.

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.

Please note that you *may* use multiple inheritance in this design if you believe it would be appropriate.

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

D. 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.

E. If you used interface(s) (in the sense of a group of related methods signatures with empty bodies) anywhere in your UML for Part A above, describe where you used interface(s) in your design and *explain why* using interface(s) improves this design. If you did not use interface(s) anywhere in your UML for this design, *explain* a situation in which using interface(s) would improve OO design.

Question 3: Unit Testing (10 points)

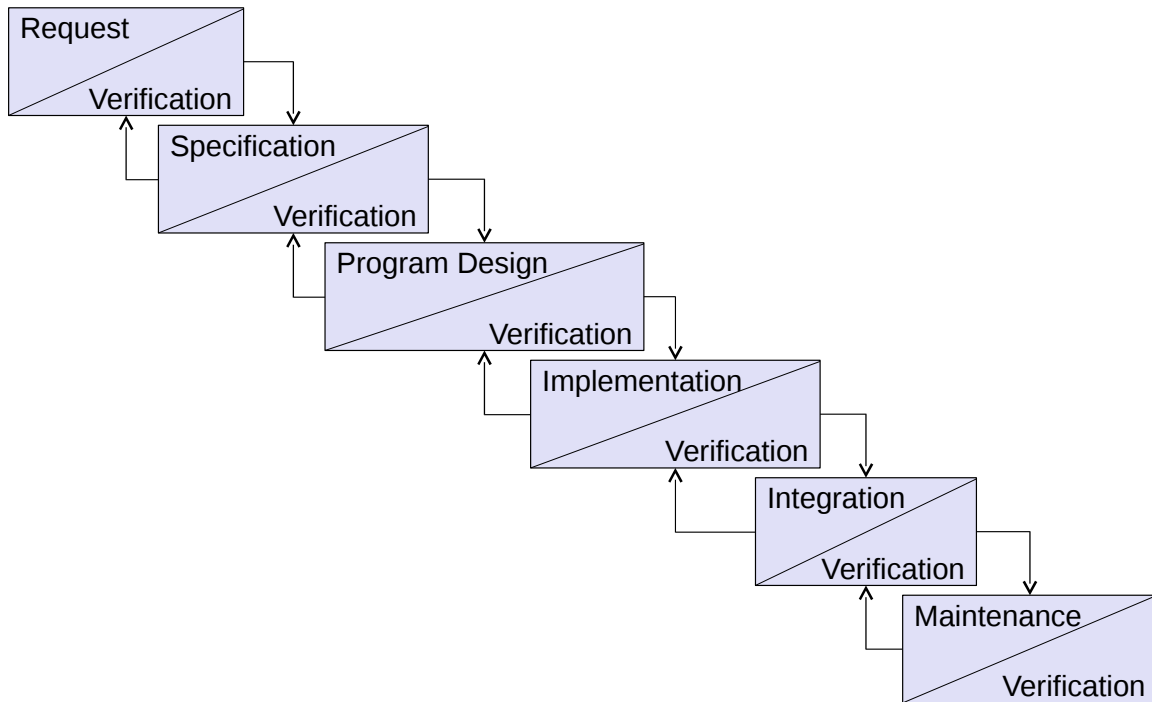
Explain two units tests you should perform on a binary search method. For each test, say what you are testing for and describe how you would go about constructing the test. (You may describe the steps in English, pseudo-code, or actual Java code, as you see fit.)

A. Test one.

B. Test two.

Question 4: Object Oriented Software Development (10 points)

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



Question 5: Java Collections Framework, Generics, and Polymorphism (20 points)

A. Consider a method with the following signature:

```
public void someMethod(List<Employee> employeeList)
```

Explain how `employeeList` in this signature could be used for subclass assignment.

B. Consider a method with the following signature:

```
public void someMethod(List<? extends Employee> employeeList)
```

Explain how `<? extends Employee>` in this signature allows for additional polymorphism regarding `employeeList`.