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 (45 points)

A. Rachel wants to write a program to help her keep track of her digitally recorded videos. In particular, the videos she is concerned with are episodes of her favorite TV shows, movies (made by the movie industry), and home videos. For TV shows, she wants to keep track of the name of the TV series, the season (year) and episode number, the running time, and the file size. For movies, she wants to keep track of title, director, primary cast, year, running time, and file size. For home videos, she wants to keep track of year, month, names of all friends and family members in the video, the running time, and the file size. Her software will allow her to search for videos based on any of these attributes.

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.

B. Six months after completing her software, Rachel has taken up making instructional films related to her many hobbies. She decides to add these videos to her software. For each of these she wants to keep track of the title of the film, the hobby it is about, the particular topic of instruction, the month and year of creation, the running time, and the file size.

Add to or modify the UML diagram you created in part A to incorporate this additional type of video.

C. Another six months down the road, Rachel decides that she wants to do more than keep track of her videos with her software. She also wants to use her software to burn DVDs to sell or give away. (Her software won't actually burn the DVDs but once she has used it to search and find the video(s) she wants, she will click "Burn to DVD" and it will call dedicated DVD-burning software to actually do the DVD burning.) However, cognizant of copyright laws, she only wants to burn DVDs of her home videos and instructional hobby films.

Add to or modify the UML diagram to incorporate this additional functionality.

D. If you used used inheritance anywhere in your UML for any of the parts 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.

E. If you used composition or aggregation anywhere in your UML for any of the parts 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.

F. If you used interface(s) (in the sense of a group of related methods with empty bodies) anywhere in your UML for any of the parts 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 2: Object-Oriented Design, Redux (10 points)

Besides the sense of 'interface' used in Question 1F, in what other sense is 'interface' used in object-oriented design? Why are interfaces (in the sense you have just described) important to OO design?

Question 3: Encapsulation and Polymorphism (15 points)

A. How does method overriding support encapsulation? *Explain* your answer.

B. How does encapsulation make subclass assignment more useful? *Explain* your answer.

C. How does late binding support encapsulation? *Explain* your answer.

Question 4: Encapsulation (5 points)

Do private methods help with encapsulation or do they detract from it? *Explain your answer.*

Question 5: Coding Conventions (5 points)

Describe one coding convention related to mutator methods. *Explain why* it is important to follow this convention.

Question 6: Java Collections Framework (JCF) and Generics (20 points)

A. List one reason to use `ArrayList` rather than arrays. *Explain your answer.*

B. List one reason to use `ArrayList` rather than `Vector`. *Explain your answer.*

C. List one reason to use `Vector` rather than `ArrayList`. *Explain your answer.*

D. List one advantage of the current JFC that uses generics over the older JCF that did not use generics. *Explain your answer.*