

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 Redux (40 points)

You are asked to develop software for cycles of various kinds. The particular kinds of cycles your software will need to keep track of are unicycles (peddled cycles with one wheel), bicycles (peddled cycles with two wheels), tricycles (peddled cycles with three wheels), and motorcycles (motorized cycles with two wheels). Each of these kinds of cycles will require its own class. However, since these types of cycles are related to each other, you should create a class hierarchy that relates all of these classes to one another. If you think it is appropriate, you could place additional classes or interfaces in this hierarchy.

A. Determine a reasonable inheritance hierarchy that you could implement in Java for these classes and draw a simplified UML diagram to describe this hierarchy. (Note that you don't need to show variables or methods in this diagram. However, you do need to indicate in the diagram which classes are abstract and which are concrete and you need to clearly mark any interfaces as interfaces.)

B. For each required class, *explain* why you made it abstract or concrete.

C. If you added any additional classes or interfaces to your hierarchy, *explain* what they represent and why you added them. For each additional class (if any), *explain* why you made it abstract or concrete. For each additional interface (if any), explain why you made it an interface.

D. For each relationship in this inheritance hierarchy, *explain* why you chose the particular relationship you chose.

E. Add to your simplified UML diagram a class for wheels. (Again, you don't need to show variables or methods in this diagram. However, you do need to indicate in the diagram whether this additional class is abstract or concrete.)

F. For each relationship you added when you added the wheel class, *explain* why you chose the particular relationship you chose.

Question 2: Overriding v. Overloading (10 points)

A. What is overriding? Where is it used? *Why* is it useful?

B. What is overloading? Where is it used? *Why* is it useful?

Question 3: Polymorphism & Generics (20 points)

A. *Explain* the type of polymorphism known as “subclass assignment.”

B. What is “type safety” and how is it related to subclass assignment?

C. How does the use of generics in Java help to ensure type safety?

D. Why is using generics to ensure type safety an advantage over type checking using the `getClass()` method (as was typically done prior to Java 1.5)?

Question 4: Java Collections Framework (20 points)

A. *Explain* one advantage of using the **LinkedList** class over using the **ArrayList** class?

B. *Explain* one advantage of using the **ArrayList** class over using the **LinkedList** class?

C. The **Set** interface does not add any new methods to those already in the **Collections** interface. Given that fact, *explain* why it is still useful to have a **Set** interface at all.

D. *Explain* the difference between ordered and sorted.

Question 5: Generics and Overloading (10 points)

Is it possible to overload methods that have already been defined to be generic? *Explain why or why not.*