

★ Multiple Choice Practice : Inheritance ★

Consider the following Book and AudioBook classes. (for #1 on next page)

```
public class Book
{
    private int numPages;
    private String bookTitle;

    public Book(int pages, String title)
    {
        numPages = pages;
        bookTitle = title;
    }

    public String toString()
    {
        return bookTitle + " " + numPages;
    }

    public int length()
    {
        return numPages;
    }
}

public class AudioBook extends Book
{
    private int numMinutes;

    public AudioBook(int minutes, int pages, String title)
    {
        super(pages, title);
        numMinutes = minutes;
    }

    public int length()
    {
        return numMinutes;
    }

    public double pagesPerMinute()
    {
        return ((double) super.length()) / numMinutes;
    }
}
```

#1

Consider the following code segment that appears in a class other than `Book` or `AudioBook`.

```
Line 1: Book[] books = new Book[2];  
Line 2: books[0] = new AudioBook(100, 300, "The Jungle");  
Line 3: books[1] = new Book(400, "Captains Courageous");  
Line 4: System.out.println(books[0].pagesPerMinute());  
Line 5: System.out.println(books[0].toString());  
Line 6: System.out.println(books[0].length());  
Line 7: System.out.println(books[1].toString());
```

Which of the following best explains why the code segment will not compile?

- (A) Line 2 will not compile because variables of type `Book` may not refer to variables of type `AudioBook`.
- (B) Line 4 will not compile because variables of type `Book` may only call methods in the `Book` class.
- (C) Line 5 will not compile because the `AudioBook` class does not have a method named `toString` declared or implemented.
- (D) Line 6 will not compile because the statement is ambiguous. The compiler cannot determine which `length` method should be called.
- (E) Line 7 will not compile because the element at index 1 in the array named `books` may not have been initialized.

"Is-a" vs. "has-a"

#2

A car dealership needs a program to store information about the cars for sale. For each car, they want to keep track of the following information: number of doors (2 or 4), whether the car has air conditioning, and its average number of miles per gallon. Which of the following is the best object-oriented program design?

- (A) Use one class, `Car`, with three instance variables:
`int numDoors`, `boolean hasAir`, and
`double milesPerGallon`.
- (B) Use four unrelated classes: `Car`, `Doors`, `AirConditioning`, and `MilesPerGallon`.
- (C) Use a class `Car` with three subclasses: `Doors`, `AirConditioning`, and `MilesPerGallon`.
- (D) Use a class `Car`, with a subclass `Doors`, with a subclass `AirConditioning`, with a subclass `MilesPerGallon`.
- (E) Use three classes: `Doors`, `AirConditioning`, and `MilesPerGallon`, each with a subclass `Car`.