CSC 203 Quiz 3 **Your name**: ______________________________

April 23, 2025

Note: You may use your notes or my notes from my website. No other resources may be used during this quiz.

We have a `Record` class, and each `Record` is made up of a collection of `Song`s. (The `Song` class is not shown since it's not relevant for this example.)

After a `Record` is released, its name can never be changed (for Grammy reasons). Naturally, its sales numbers (`numUnitsSold`) can be increased from the initial value, and it must have one or more songs.

Consider the code below that depicts a stub of the `Record` class.

```java
// Record.java
import java.util.List;

public class Record {
    // Questions are about the instance variable declarations for the following:
    // * name
    // * songs
    // * numUnitsSold

    /**
    * Constructor to create a new record with the specified
    * name and songs.
    */
    public Record(String name, List<Song> songs) {
        this.name = name;
        this.songs = songs;
        this.numUnitsSold = 0;
    }

    // Assume that the appropriate getters and setters have been implemented.
}
```

Question 1. (2 points) What is the most appropriate declaration for the `Record` class's `name` variable? Why?

1. `public String name`
2. `private String name`
3. *`private final String name`* | *because the name can't change after initialisation*
4. `private static String name`

Question 2. (6 points) Consider the constructor in the `Record` class above and answer this question.

Here's the signature for the `Record` class's constructor. Consider it while answering these questions.

```
public Record(String name, List<Song> songs)
```

For each of the following code snippets, indicate *whether or not it will compile*. If it will not compile, indicate *which line* will have the compiler error and why.

a)

```
1.| ArrayList<Song> songList1 = new ArrayList<Song>();
2.| Record record = new Record(‘‘Magical Mystery Tour’’, songList1);
```

This code will compile.

b)

```
1.| List<Song> songList1 = new LinkedList<Song>();
2.| Record record = new Record(‘‘Magical Mystery Tour’’, songList1);
```

This code will compile.

c)

```
1.| Object songList1 = new ArrayList<Song>();
2.| Record record = new Record(‘‘Magical Mystery Tour’’, songList1);
```

This code will not compile. It will give an error on line 2, because the `Record` constructor expects a `List` but `songList1` static type is `Object` and java won't automatically downcast `Object` down to `List`.

Question 3. (1 point)

```
List<String> myList = new LinkedList<>();
```

What is the **static type** of `myList`? What is its **dynamic type**?

1. static type is `LinkedList`, dynamic type is `List`
2. this code will not compile
3. *static type is `List`, dynamic type is `LinkedList`*