

1. In property binding, if I bind a target A to a source B (A.bind(B)):
 - a. If B changes, A will change value
2. What is TRUE about an event handler/listener?
 - a. It processes the event fired from a source object
3. An event handler / listener must be registered with the source object in order to handle events from it.
 - a. True
4. An inner class CANNOT access the private data/methods of its outer class.
 - a. False
5. An inner class can be protected or private.
 - a. True
6. For a non-static inner class B defined within a class A, I MUST have an instance of class A in order to create an instance of class B OUTSIDE of class A.
 - a. False
7. In Java, a lambda expression can be used to define an interface with multiple abstract methods.
 - a. False
8. Given a Button muffinButton, call setOnAction() and pass in a lambda expression that prints "MUFFIN" to STDOUT.

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.Pane;
import javafx.stage.Stage;

public class A extends Application {
    public void start(Stage primaryStage) {
        Pane pane = new Pane();
        Scene scene = new Scene(pane, 255, 255);
        Button muffinButton = new Button();

        muffinButton.setOnAction(e -> {
            System.out.println("MUFFIN!");
        });

        pane.getChildren().add(muffinButton);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String [] args) {
        launch(args);
    }
}
```

9. Let's say I do NOT need more than one instance of A, but I do need more than one method in A. Which of the following is the BEST choice?
 - a. Inner class
10. Let's say Edible only contains ONE abstract method. I do NOT need any fields or additional methods. What is the BEST choice?
 - a. Lambda expression