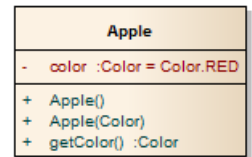


1. (5 points) In your own words, **describe** what an instance is.

2. (5 points) Consider the Apple class in the UML diagram on the right. **Write** a couple lines of code to call the instance method `getColor()` from **outside** of the Apple class. Declare all variables that you use.



3. (5 points) **Write** two differences between an abstract class and an interface.

4. (5 points) **Name** two Java classes/interfaces – one whose instances are “event sources”, and one whose instances are “event handlers.”

5. (5 points) JavaFX provides a framework for responding to user actions. When a button is pressed, some of your code should be run. **Describe** how the framework determines what code to run.

(I don't expect you to need the space below this line.)

6. (5 points) Consider this code-snippet.

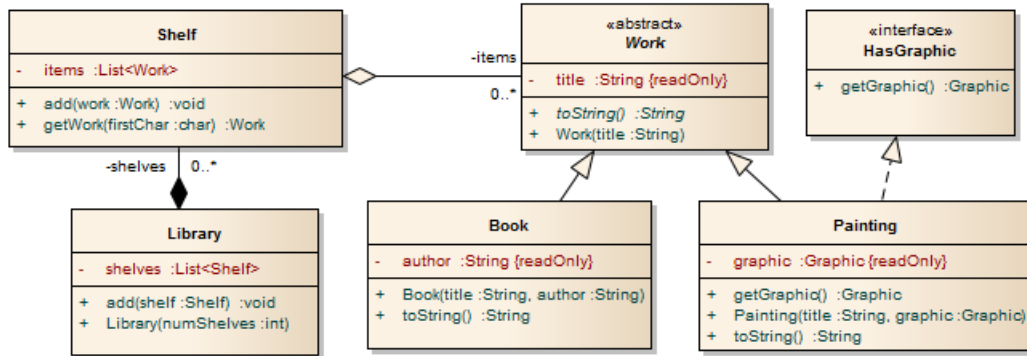
```
public class Gui extends Application {
    private String title;
    ...
    @Override
    public void start(Stage primaryStage) {
        ...
        Button b = new Button("Press me");
        b.setOnAction(e->System.out.println(title));
        ...
    }
}
```

Write whether or not it is legal for the lambda expression above to access the instance variable `title`, and **explain** your answer.

7. (10 points) **Write** an anonymous inner class implementing the `EventHandler<ActionEvent>` interface. This interface contains one method: `public void handle(ActionEvent event);`. Your handler should set the text of the variable `textLabel` to "hi" when it is called. **Assign** the variable `a` to point to an instance of your anonymous inner class.
8. (5 points) Consider an `ActionHandler` that handles multiple buttons. **Describe** one technique the handler could use to determine which button was clicked.
9. (5 points) Suppose you call a method that throws a `FileNotFoundException`, a checked exception. **Describe** the consequences of not catching this exception with a try-catch block.

10. (17 points – 2 points for each multiple choice, 1 point for the true/false.)

Consider the UML diagram for the program below. This program is similar to the one on Exam 1, but there are several key differences.



- a. **Select** one. The relationship between *Work* and *Book* is
 - i. Composition
 - ii. Aggregation
 - iii. Inner-class
 - iv. Inheritance
 - v. Implementation

- b. **Select** one. The relationship between *Library* and *Shelf* is
 - i. Composition
 - ii. Aggregation
 - iii. Inner-class
 - iv. Inheritance
 - v. Implementation

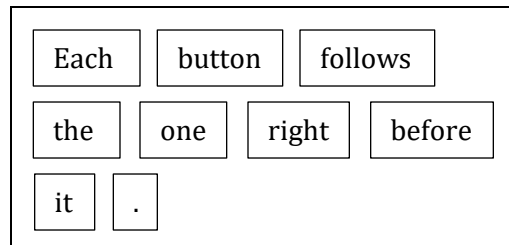
- c. **Select** one. The relationship between *HasGraphic* and *Painting* is
 - i. Composition
 - ii. Aggregation
 - iii. Inner-class
 - iv. Inheritance
 - v. Implementation

- d. **Select** one. As indicated on the diagram, the `toString` method of *Work* is...
 - i. abstract
 - ii. volatile
 - iii. void
 - iv. static
 - v. final

- e. **Select** one. As indicated on the diagram, the `title` variable of *Work* is...
 - i. abstract
 - ii. volatile
 - iii. void
 - iv. static
 - v. final

(continued from previous page – see figure there)

- f. **Select** one. Which of the following statements is valid?
- Book b = new Book();
 - Work w = new Book("John Hancock","Declaration of Independence");
 - Book b = new Work();
 - Work w = new Work("My Masterpiece");
- g. **Select** one. Which of the following statements is valid if lib is a Library?
- lib.add(new Book());
 - lib.add(new Shelf("Top shelf"));
 - lib.add(new Book("Dean & Dean","Java"));
 - lib.add(new Shelf());
- h. **Circle** one: true / false: A shelf can contain more than one book.
- i. **Select** one. (Unrelated to the diagram on the previous page.) Which pane would be best for designing this layout:



- FlowPane
 - VBox
 - HBox
 - TilePane
11. (8 points) Considering the UML diagram from the previous problem, write the entire add method for the Library. (The Library's constructor takes a numShelves argument. This is the initial number of shelves and does not need to limit the total number of shelves that the Library has.)