Java 8-2: Inner classes exercises

imagesJComboBox.addItemListener(

**35** new ItemListener() // anonymous inner class

**36** {

**37** // handle JComboBox event

**38** @Override

**39** public void itemStateChanged(ItemEvent event)

**40** {

**41** // determine whether item selected

**42** if (event.getStateChange() == ItemEvent.SELECTED)

**43** label.setIcon(icons[

**44** imagesJComboBox.getSelectedIndex()]);

**45** }

**46** } // end anonymous inner class

**47** ); // end call to addItemListener

Questions: Please check JavaDocs for any classes, methods, interfaces you have not seen before that may help you answer the questions. The object imagesJComboBox is an instance of class JComboBox. The actual details of the overridden method are not critical here nor is it important for you to have a good understanding of AWT or Swing.

1. In line 35 we see the ItemListener interface and it looks like we are trying to create a new instance of this interface, and it even appears there is a constructor. What is actually occurring here with the anonymous class?
2. What method is being overridden here and why?
3. What is the purpose of the addItemListener method?
4. What is the parameter for the addItemListener method?
5. You can replace this anonymous inner class with a lambda expression at a later stage after we discuss more about Java 8 features. Is ItemListener a functional interface?
6. What is a static member class? Show an example in code.
7. How can you access a static member class outside the containing class?
8. What is a non-static inner class? Show an example in code.
9. What is a method local class? Show an example. What variables in terms of scope can a method local class access?
10. What are some advantages of using inner classes?
11. What are some disadvantages of using inner classes?
12. Do you like the idea that Java supports inner classes? Would you recommend using inner classes to other developers on your team?