Chapter 10: Interfaces

Test Bank

Multiple Choice

1. Which of the following statements about a Java interface is NOT true?

a) A Java interface defines a set of methods that are required.

b) A Java interface must contain more than one method.

c) A Java interface specifies behavior that a class will implement.

d) All methods in a Java interface must be abstract.

Answer: b

Section Reference: 10.1 Using Interfaces for Algorithm Reuse

Title: Which statement about a Java interface is NOT true?

Difficulty: Medium

2. A method that has no implementation is called a/an \_\_\_\_ method.

a) interface

b) implementation

c) overloaded

d) abstract

Answer: d

Section Reference: 10.1 Using Interfaces for Algorithm Reuse

Title: What is a method with no implementation called?

Difficulty: Easy

3. Which statement about methods in an interface is true?

a) All methods in an interface are automatically private.

b) All methods in an interface are automatically public.

c) All methods in an interface are automatically static.

d) All methods in an interface must be explicitly declared as private or public.

Answer: b

Section Reference: 10.1 Using Interfaces for Algorithm Reuse

Title: Which statement about methods in an interface is true?

Difficulty: Medium

4. Which of the following statements about abstract methods is true?

a) An abstract method has a name, parameters, and a return type, but no code in the body of the method.

b) An abstract method has parameters, a return type, and code in its body, but has no defined name.

c) An abstract method has a name, a return type, and code in its body, but has no parameters.

d) An abstract method has only a name and a return type, but no parameters or code in its body.

Answer: a

Section Reference: 10.1 Using Interfaces for Algorithm Reuse

Title: Which statement about abstract methods is true?

Difficulty: Hard

5. Which of the following statements about an interface is true?

a) An interface has methods and instance variables.

b) An interface has methods but no instance variables.

c) An interface has neither methods nor instance variables.

d) An interface has both public and private methods.

Answer: b

Section Reference: 10.1 Using Interfaces for Algorithm Reuse

Title: Which statement about an interface is true?

Difficulty: Medium

6. Which of the following statements about interfaces is NOT true?

a) Interfaces can make code more reusable.

b) Interface types can be used to define a new reference data type.

c) Interface types can be used to express common operations required by a service.

d) Interfaces have both private and public methods.

Answer: d

Section Reference: 10.1 Using Interfaces for Algorithm Reuse

Title: Which statement about interfaces is NOT true?

Difficulty: Medium

7. To use an interface, a class header should include which of the following?

a) The keyword extends and the name of an abstract method in the interface

b) The keyword extends and the name of the interface

c) The keyword implements and the name of an abstract method in the interface

d) The keyword implements and the name of the interface

Answer: d

Section Reference: 10.1 Using Interfaces for Algorithm Reuse

Title: To use an interface, what must a class header include?

Difficulty: Medium

8. \_\_\_\_ can reduce the coupling between classes.

a) Static methods

b) Abstract methods

c) Interfaces

d) Objects

Answer: c

Section Reference: 10.1 Using Interfaces for Algorithm Reuse

Title: \_\_\_\_ can reduce the coupling between classes.

Difficulty: Easy

9. Consider the following code snippet:

public class Inventory implements Measurable

{

. . .

public double getMeasure();

{

return onHandCount;

}

}

Why is it necessary to declare getMeasure as public ?

a) All methods in a class are not public by default.

b) All methods in an interface are private by default.

c) It is necessary only to allow other classes to use this method.

d) It is not necessary to declare this method as public.

Answer: a

Section Reference: Common Error 9.1

Title: Why is it necessary to declare getMeasure as public?

Difficulty: Hard

10. Which of the following statements about interfaces is NOT true?

a) Interfaces can make code more reusable.

b) An interface provides no implementation.

c) A class can implement only one interface type.

d) Interfaces can reduce the coupling between classes.

Answer: c

Section Reference: 10.1 Using Interfaces for Algorithm Reuse

Title: Which statement about interfaces is NOT true?

Difficulty: Medium

11. \_\_\_\_ methods must be implemented when using an interface.

a) Abstract.

b) Private.

c) Public.

d) Static

Answer: a

Section Reference: 10.1 Using Interfaces for Algorithm Reuse

Title: \_\_\_\_ methods must be implemented when using an interface.

Difficulty: Easy

12. Suppose you are writing an interface called Resizable, which includes one void method called resize.

public interface Resizable

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}

Which of the following can be used to complete the interface declaration correctly?

a) private void resize();

b) protected void resize();

c) void resize();

d) public void resize() { System.out.println("resizing ..."); }

Answer: c

Title: Identify correct method declaration in interface

Difficulty: Easy

Section Reference 1: 10.1 Using Interfaces for Algorithm Reuse

13. Consider the following declarations:

public interface Encryptable

{

void encrypt(String key);

}

public class SecretText implements Encryptable

{

private String text;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

{

// code to encrypt the text using encryption key goes here

}

}

Which of the following method headers should be used to complete the SecretText class?

a) public void encrypt()

b) public void encrypt(String aKey)

c) public String encrypt(String aKey)

d) void encrypt(String aKey)

Answer: b

Title: Identify correct method header to use when implementing interface

Difficulty: Medium

Section Reference 1: 10.1 Using Interfaces for Algorithm Reuse

14. Consider the following code snippet:

public class Inventory implements Measurable

{

. . .

double getMeasure();

{

return onHandCount;

}

}

What is wrong with this code?

a) The getMeasure() method must be declared as private.

b) The getMeasure() method must include the implements keyword.

c) The getMeasure() method must be declared as public.

d) The getMeasure() method must not have any code within it.

Answer: c

Section Reference: Common Error 10.1

Title: What is wrong with this code about interfaces?

Difficulty: Medium

15. Consider the following code snippet:

public interface Sizable

{

int LARGE\_CHANGE = 100;

int SMALL\_CHANGE = 20;

void changeSize();

}

Which of the following statements is true?

a) LARGE\_CHANGE and SMALL\_CHANGE are automatically public static final.

b) LARGE\_CHANGE and SMALL\_CHANGE are instance variables

c) LARGE\_CHANGE and SMALL\_CHANGE must be defined with the keywords private static final.

d) LARGE\_CHANGE and SMALL\_CHANGE must be defined with the keywords public static final.

Answer: a

Section Reference: Special Topic 10.1

Title: Which statement about interface definitions is true?

Difficulty: Hard

16. Consider the following code snippet:

public class Inventory implements Measurable

{

. . .

double getMeasure();

{

return onHandCount;

}

}

The compiler complains that the getMeasure method has a weaker access level than the Measurable interface. Why?

a) All of the methods in a class have a default access level of package access, while the methods of an interface have a default access level of private.

b) All of the methods in a class have a default access level of package access, while the methods of an interface have a default access level of public.

c) The variable onHandCount was not declared with public access.

d) The getMeasure method was declared as private in the Measurable interface.

Answer: b

Section Reference: Common Error 10.1

Title: What is wrong with this code implementing an interface?

Difficulty: Hard

17. Which of the following is true regarding a class and interface types?

a) You can convert from a class type to any interface type that is in the same package as the class.

b) You can convert from a class type to any interface type that the class implements.

c) You can convert from a class type to any interface type that the class defines.

d) You cannot convert from a class type to any interface type.

Answer: b

Section Reference: 10.2 Working with Interface Variables

Title: Which is true regarding a class and interface types?

Difficulty: Hard

18. Consider the following code snippet.

public interface Measurable

{

double getMeasure();

}

public class Coin implements Measurable

{

public double getMeasure()

{

return value;

}

...

}

public class DataSet

{

...

public void add()

{

...

}

}

public class BankAccount

{

...

public void add()

{

...

}

}

Which of the following statements is correct?

a)

Coin dime = new Coin(0.1, "dime");

Measurable x = dime;

b)

Coin dime = new Coin(0.1, "dime");

Dataset x = dime;

c)

Coin dime = new Coin(0.1, "dime");

DataSet x == (Measureable)dime;

d)

Coin dime = new Coin(0.1, "dime");

BankAccount x = dime;

Answer: a

Section Reference: 10.2 Working with Interface Variables

Title: Which code statement is correct?

Difficulty: Medium

19. Which of the following statements about converting between types is true?

a) When you cast number types, you take a risk that an exception will occur.

b) When you cast number types, you will not lose information.

c) When you cast object types, you take a risk of losing information.

d) When you cast object types, you take a risk that an exception will occur.

Answer: d

Section Reference: 10.2 Working with Interface Variables

Title: Which statement about converting between types is true?

Difficulty: Medium

20. Which of the following statements about interfaces is true?

a) You can define an interface variable that refers to an object of any class in the same package.

b) You cannot define a variable whose type is an interface.

c) You can instantiate an object from an interface class.

d) You can define an interface variable that refers to an object only if the object belongs to a class that implements the interface.

Answer: d

Section Reference: Common Error 10.2

Title: Which statement about interfaces is true?

Difficulty: Hard

21. You have created a class named Motor that implements an interface named Measurable. You have declared a variable of type Measureable named motorTemperature. Which of the following statements is true?

a) The object to which motorTemperature refers has type Measurable.

b) The object to which motorTemperature refers has type Motor.

c) This declaration is illegal.

d) You must construct a motorTemperature object from the Measurable interface.

Answer: b

Section Reference: 10.2 Working with Interface Variables

Title: Which statement about a variable whose type is an interface variable is true?

Difficulty: Medium

22. \_\_\_\_ occurs when a single class has several methods with the same name but different parameter types.

a) Casting

b) Polymorphism

c) Overloading

d) Instantiation

Answer: c

Section Reference: 10.2 Working with Interface Variables

Title: \_\_\_\_ occurs when a single class has several methods with the same name.

Difficulty: Easy

23. Consider the following code snippet:

public class Demo

{

public static void main(String[] args)

{

Point[] p = new Point[4];

p[0] = new Colored3DPoint(4, 4, 4, Color.BLACK);

p[1] = new ThreeDimensionalPoint(2, 2, 2);

p[2] = new ColoredPoint(3, 3, Color.RED);

p[3] = new Point(4, 4);

for (int i = 0; i < p.length; i++)

{

String s = p[i].toString();

System.out.println("p[" + i + "] : " + s);

}

return;

}

}

This code is an example of \_\_\_\_.

a) overloading

b) callback

c) early binding

d) polymorphism

Answer: d

Section Reference: 10.2 Working with Interface Variables

Title: his code is an example of \_\_\_\_.

Difficulty: Medium

24. If you have multiple classes in your program that have implemented the same interface in different ways, how is the correct method executed?

a) The Java virtual machine must locate the correct method by looking at the class of the actual object.

b) The compiler must determine which method implementation to use.

c) The method must be qualified with the class name to determine the correct method.

d) You cannot have multiple classes in the same program with different implementations of the same interface.

Answer: a

Section Reference: 10.2 Working with Interface Variables

Title: How does the correct method get called?

Difficulty: Medium

25. Consider the following declarations:

public interface Displayable

{

void display();

}

public class Picture implements Displayable

{

private int size;

public void increaseSize()

{

size++;

}

public void decreaseSize()

{

size--;

}

public void display()

{

System.out.println(size);

}

public void display(int value)

{

System.out.println(value \* size);

}

}

What method invocation can be used to complete the code segment below?

Displayable picture = new Picture();

picture.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

a) increaseSize()

b) decreaseSize()

c) display()

d) display(5)

Answer: c

Title: Identify legal method invocation for variable of interface type

Difficulty: Medium

Section Reference 1: 10.2 Working with Interface Variables

26. Which of the following can potentially be changed when implementing an interface?

a) The parameters of a method in the interface.

b) The name of a method in the interface.

c) The return type of a method in the interface.

d) You cannot change the name, return type, or parameters of a method in the interface.

Answer: d

Section Reference: Common Error 10.3

Title: Which can be changed when implementing an interface?

Difficulty: Medium

27. Consider the following class:

public class Player implements Comparable

{

private String name;

private int goalsScored;

// other methods go here

public int compareTo(Object otherObject)

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

return (goalsScored – otherPlayer.goalsScored);

}

}

What statement can be used to complete the compareTo() method?

a) Player otherPlayer = otherObject;

b) Object otherPlayer = otherObject;

c) Player otherPlayer = (Player) otherObject;

d) Object otherPlayer = (Player) otherObject;

Answer: c

Title: Identify correct statement to complete compareTo() method in sample class

Difficulty: Medium

Section Reference 1: 10.3 The Comparable Interface

28. The method below is designed to print the smaller of two values received as arguments. Select the correct expression to complete the method.

public void showSmaller(Comparable value1, Comparable value2)

{

if ( \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ )

System.out.println(value1 + " is smaller.");

else

System.out.println(value2 + " is smaller.");

}

a) value1 < value2

b) value1.compareTo(value2) > 0

c) value1.compareTo(value2) == 0

d) value1.compareTo(value2) < 0

Answer: d

Title: Identify correct statement to compare two Comparable objects

Difficulty: Medium

Section Reference 1: 10.3 The Comparable Interface

29. Which of the following statements about a callback is NOT true?

a) A callback can allow you to implement a new method for a class that is not under your control.

b) A callback can be implemented using an interface.

c) A callback is a mechanism for specifying code to be executed later.

d) A callback method declared in an interface must specify the class of objects that it will manipulate.

Answer: d

Section Reference: 10.4 Using Interfaces for Callbacks

Title: Which statement about a callback is NOT true?

Difficulty: Medium

30. In Java, \_\_\_\_ can be used for callbacks.

a) Objects

b) Interfaces

c) Classes

d) Operators

Answer: b

Section Reference: 10.4 Using Interfaces for Callbacks

Title: In Java, \_\_\_\_ can be used for callbacks.

Difficulty: Easy

31. You wish to implement a callback method for an object created from a system class that you cannot change. What approach does the textbook recommend to accomplish this?

a) Create a new class that mimics the system class.

b) Extend the system class.

c) Use an inner class in the interface.

d) Use a helper class that implements the callback method.

Answer: d

Section Reference: 10.4 Using Interfaces for Callbacks

Title: How to implement a callback method for a system class.

Difficulty: Medium

32. Which of the following statements about helper classes is true?

a) Helper classes must be inner classes.

b) Helper classes must implement interfaces.

c) Helper classes help reduce coupling.

d) Helper classes cannot contain instance variables.

Answer: c

Section Reference: 10.4 Using Interfaces for Callbacks

Title: Which statement about helper classes is true?

Difficulty: Easy

33. Consider the following declarations:

public interface Measurer

{

int measure(Object anObject);

}

public class StringLengthMeasurer implements Measurer

{

public int measure(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

{

String str = (String) anObject;

return str.length();

}

}

What parameter declaration can be used to complete the callback measure method?

a) Object anObject

b) String anObject

c) Object aString

d) String aString

Answer: a

Title: Identify correct parameter declaration to complete callback method

Difficulty: Easy

Section Reference 1: 10.4 Using Interfaces for Callbacks

34. Assuming that interface Resizable is declared elsewhere, consider the following class declaration:

public class InnerClassExample

{

public static void main(String[] args)

{

class SizeModifier implements Resizable

{

// class methods

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ // missing statement

}

}

Which of the following declarations can be used to complete the main method?

a) Resizable something = new Resizable();

b) Resizable something = new SizeModifier();

c) Resizable something = new InnerClassExample();

d) SizeModifier something = new Resizable();

Answer: b

Title: Identify correct declaration using inner class to complete main method

Difficulty: Medium

Section Reference 1: 10.5 Inner Classes

35. Which of the following statements about an inner class is true?

a) An inner class can only be defined within a specific method of its enclosing class.

b) An inner class can only be defined outside of any method within its enclosing class.

c) An inner class must implement an interface.

d) An inner class may be anonymous.

Answer: d

Section 10.5 Inner Classes

Title: Which statement about an inner class is true?

Difficulty: Easy

36. A/an \_\_\_\_ class defined in a method signals to the reader of your program that the class is not interesting beyond the scope of the method.

a) A class cannot be defined within a method

b) abstract

c) interface

d) inner

Answer: d

Section 10.5 Inner Classes

Title: A/an \_\_\_ class defined in a method signals what?

Difficulty: Easy

37. Which of the following statements about an inner class is true?

a) An inner class that is defined inside a method is publicly accessible.

b) An inner class that is defined inside a method is not publicly accessible.

c) An inner class that is defined inside an enclosing class but outside of its methods is not available to all methods of the enclosing class.

d) An inner class is used for a utility class that should be visible elsewhere in the program.

Answer: b

Section 10.5 Inner Classes

Title: Which statement about an inner class is true?

Difficulty: Hard

38. Which of the following is a good use for an anonymous class?

a) Use an anonymous class to implement polymorphism.

b) Use an anonymous class to implement a callback.

c) Use an anonymous class when all methods of the class will be static methods.

d) Use an anonymous class when the program will only need one object of the class.

Answer: d

Section Reference: Special Topic 9.2

Title: Which is a good use for an anonymous class?

Difficulty: Hard

39. Consider the following code snippet:

myImage.add(new Rectangle(10,10,10,10));

This code is an example of using \_\_\_\_.

a) An anonymous class.

b) An anonymous object.

c) An abstract object.

d) An abstract class.

Answer: b

Section Reference: Special Topic 10.2

Title: This code is an example of using \_\_\_\_.

Difficulty: Hard

40. Which of the following statements about a mock class is true?

a) A mock class does not provide an implementation of the services of the actual class.

b) A mock class provides a complete implementation of the services of the actual class.

c) A mock class provides a simplified implementation of the services of the actual class.

d) A mock class must be an interface.

Answer: c

Section 10.6 Mock Objects

Title: Which statement about a mock class is true?

Difficulty: Easy

41. What role does an interface play when using a mock class?

a) The mock class should be an interface that will be implemented by the real class.

b) The real class should be an interface that will be implemented by the mock class.

c) Interfaces are not involved when using mock classes.

d) An interface should be implemented by both the real class and the mock class to guarantee that the mock class accurately simulates the real class when used in a program.

Answer: d

Section 10.6 Mock Objects

Title: What role does an interface play in a mock class?

Difficulty: Medium

42. Which of the following statements about events and graphical user interface programs is true?

a) Your program must instruct the Java window manager to send it notifications about specific types of events to which the program wishes to respond.

b) The Java window manager will automatically send your program notifications about all events that have occurred.

c) Your program must respond to notifications of all types of events that are sent to it by the Java window manager.

D) Your program must override the default methods to handle events.

Answer: a

Section 10.7 Event Handling

Title: Which statement about events and GUI programs is true?

Difficulty: Medium

43. Consider the following class:

public class ClickListener implements ActionListener

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

{

System.out.println("mouse event ...");

}

}

Which of the following method headers should be used to complete the ClickListener class?

a) public void actionPerformed(ActionEvent event)

b) public void actionPerformed(ClickListener event)

c) public void actionPerformed()

d) public void actionPerformed(ActionListener event)

Answer: a

Title: Identify correct actionPerformed method header

Difficulty: Easy

Section Reference 1: 10.7 Event Handling

44. \_\_\_\_ are generated when the user presses a key, clicks a button, or selects a menu item.

a) Listeners

b) Interfaces.

c) Events.

d) Errors.

Answer: c

Section 10.7 Event Handling

Title: \_\_\_\_ are generated when the user interacts with GUI components.

Difficulty: Easy

45. A/an \_\_\_\_ belongs to a class whose methods describe the actions to be taken when a user clicks a user-interface graphical object.

a) Event listener

b) Event source

c) Action listener

d) Action method

Answer: a

Section 10.7 Event Handling

Title: A/an \_\_\_\_ belongs to a class whose methods describe actions to be taken.

Difficulty: Easy

46. Which of the following is an event source?

a) A JButton object.

b) An event listener.

c) An inner class.

d) An event adapter.

Answer: a

Section 10.7 Event Handling

Title: Which is an event source?

Difficulty: Easy

47. To respond to a button event, a listener must supply instructions for the \_\_\_\_ method of the ActionListener interface.

a) actionEvent.

b) actionPerformed

c) eventAction

d) eventResponse

Answer: b

Section 10.7 Event Handling

Title: To respond to a button event, a listener must implement the \_\_\_\_ method.

Difficulty: Medium

48. To associate an event listener with a JButton component, you must use the \_\_\_ method of the JButton class.

a) addEventListener.

b) addActionListener.

c) addButtonListener.

d) addListener

Answer: b

Section 10.7 Event Handling

Title: How to associate an event listener with a JButton component.

Difficulty: Easy

49. The methods of a/an \_\_\_\_ describe the actions to be taken when an event occurs.

a) event source

b) event listener

c) event interface

d) action source

Answer: b

Section 10.7 Event Handling

Title: The methods of a/an \_\_\_\_ describe the actions when an event occurs.

Difficulty: Easy

50. When an event occurs, the event source notifies all \_\_\_\_.

a) components

b) panels

c) interfaces

d) event listeners

Answer: d

Section 10.7 Event Handling

Title: When an event occurs, the event source notifies all \_\_\_\_.

Difficulty: Easy

51. Which container is used to group multiple user-interface components together?

a) text area

b) table

c) panel

d) rectangle

Answer: c

Section 10.7 Event Handling

Title: Which container groups user-interface components?

Difficulty: Easy

52. Which of the following statements will compile without error?

a)

public interface AccountListener()

{

void actionPerformed(AccountEvent event);

}

b)

public interface AccountListener()

{

void actionPerformed(AccountEvent);

}

c)

public interface AccountListener

{

void actionPerformed(AccountEvent event);

}

d)

public abstract AccountListener

{

void actionPerformed(AccountEvent event);

}

Answer: c

Section 10.7 Event Handling

Title: Which interface definition compiles without error?

Difficulty: Hard

53. Which of the following statements about listener classes is true?

a) A listener class cannot be declared as an anonymous inner class.

b) A listener class cannot be declared as an inner class.

c) A listener class must be declared as an inner class.

d) A listener class must be declared as an anonymous inner class.

Answer: a

Section 10.7 Event Handling

Title: Which statement about listener classes is true?

Difficulty: Hard

54. Consider the following code snippet:

JFrame frame = new JFrame();

JPanel panel = new JPanel

Which statement would add the panel to the frame?

a) frame.add(panel);

b) frame.add(JPanel panel);

c) frame.addComponent(panel);

d) frame.addComponent(JPanel panel);

Answer: a

Section 10.8 Building Applications with Buttons Title: How to add a panel to a frame.

Difficulty: Easy

55. Consider the following code snippet:

import \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

import java.awt.event.ActionListener;

/\*\*

An action listener that prints.

\*/

public class ClickListener implements ActionListener

{

public void actionPerformed(ActionEvent event)

{

System.out.println("I was clicked.");

}

}

Which of the following statements will complete this code?

a) java.swing.event.ActionEvent;.

b) javax.swing.event.ActionEvent;

c) javax.awt.event.ActionEvent;

d) java.awt.event.ActionEvent;

Answer: d

Section 10.7 Event Handling

Title: Which statement will complete this code?

Difficulty: Hard

56. Event listeners are often installed as \_\_\_\_ classes so that they can have access to the surrounding fields, methods, and final variables.

a) Inner

b) Interface

c) Abstract

d) Helper

Answer: a

Section 10.7 Event Handling

Title: Event listeners are often installed as \_\_\_\_ classes.

Difficulty: Medium

57. Which of the following statements about an inner class is true?

a) An inner class may not be declared within a method of the enclosing scope.

b) An inner class may only be declared within a method of the enclosing scope.

c) An inner class can access variables from the enclosing scope only if they are passed as constructor or method parameters.

d) The methods of an inner class can access variables declared in the enclosing scope.

Answer: d

Section 10.7 Event Handling

Title: Which statement about an inner class is true?

Difficulty: Medium

58. Consider the following code snippet:

public class ClickListener implements ActionListener

{

public void actionPerformed(ActionEvent event)

{

System.out.println("I was clicked.");

}

}

public class ButtonTester

{

public static void main(String[] args)

{

JFrame frame = new JFrame();

JButton button = new JButton("Click me!");

frame.add(button);

ActionListener listener = new ClickListener();

button.addActionListener(listener);

...

}

}

Which of the following statements is correct?

a) Class ButtonTester is an interface type.

b) A ClickListener object can be added as a listener for the action events that buttons generate.

c) Class ClickListener is an interface type.

d) Class ButtonTester implements an interface type.

Answer: b

Section 10.7 Event Handling

Title: Select the correct statement about a button's event code.

Difficulty: Medium

59. An inner class can access local variables from the enclosing scope only if they are declared as \_\_\_\_.

a) private

b) public

c) static

d) final

Answer: b

Section 10.7 Event Handling

Title: An inner class can access local variables declared as \_\_\_\_.

Difficulty: Medium

60. Which of the following code statements creates a graphical button which has "Calculate" as its label ?

a) Button JButton = new Button("Calculate").

b) button = new Button(JButton, "Calculate").

c) button = new JButton("Calculate").

d) JButton button = new JButton("Calculate").

Answer: d

Section 10.8 Building Applications with ButtonsTitle: Which statement creates a button?

Difficulty: Easy

61. To build a user interface that contains graphical components, the components \_\_\_\_.

a) Must be added directly to a frame component.

b) Must each be added to a separate panel.

c) Must be added to a panel that is contained within a frame.

d) Must be added to a frame that is contained within a panel.

Answer: c

Section 10.8 Building Applications with ButtonsTitle: To build a user interface, the graphical components \_\_\_\_.

Difficulty: Easy

62. How do you specify what the program should do when the user clicks a button?

a) Specify the actions to take in a class that implements the ButtonListener interface.

b) Specify the actions to take in a class that implements the ButtonEvent interface .

c) Specify the actions to take in a class that implements the ActionListener interface.

d) Specify the actions to take in a class that implements the EventListener interface.

Answer: c

Section 10.8 Building Applications with ButtonsTitle: How to specify action for a button?

Difficulty: Medium

63. A(n) \_\_\_\_ has an instance method addActionListener() for specifying which object is responsible for implementing the action associated with the object.

a) JFrame

b) JSlider

c) JButton

d) JLabel

Answer: c

Section 10.8 Building Applications with ButtonsTitle: How to specify action for a button.

Difficulty: Medium

64. Consider the following code snippet:

public static void main(String[] args)

{

Order myOrder = new Order();

JButton button = new JButton("Calculate");

JLabel label = new JLabel("Total amount due");

. . .

class MyListener implements ActionListener

{

public void actionPerformed(ActionEvent event)

{

label.setText("Total amount due " + myOrder.getAmountDue());

}

}

}

What is wrong with this code?

a) label must be declared as final.

b) myOrder must be declared as final

c) label and myOrder must be declared as final

d) label and button must be declared as final.

Answer: c

Section 10.8 Building Applications with ButtonsTitle: What is wrong with this listener code?

Difficulty: Medium

65. Consider the following code snippet:

public static void main(String[] args)

{

final Order myOrder = new Order();

JButton button = new JButton("Calculate");

final JLabel label = new JLabel("Total amount due");

. . .

class MyListener implements ActionListener

{

public void actionPerformed(ActionEvent event)

{

. . .

}

}

}

Which of the local variables can be accessed within the actionPerformed method?

a) Only button can be accessed..

b) All of the local variables can be accessed.

c) label and myOrder can be accessed.

d) Only myOrder can be accessed.

Answer: c

Section 10.8 Building Applications with ButtonsTitle: Which local variables can be accessed?

Difficulty: Medium

66. Consider the following code snippet which is supposed to show the total order amount when the button is clicked:

public static void main(String[] args)

{

final Order myOrder = new Order();

JButton button = new JButton("Calculate");

final JLabel label = new JLabel("Total amount due");

. . .

class MyListener implements ActionListener

{

public void actionPerformed(ActionEvent event)

{

label.setText("Total amount due " + myOrder.getAmountDue());

}

}

ActionListener listener = new MyListener();

}

What is wrong with this code?

a) button should be declared as final

b) The listener has not been attached to the button.

c) The listener cannot access the methods of the myOrder object.

d) This code will display the total order amount.

Answer: b

Section Reference: Common Error 10.4

Title: What is wrong with this listener code?

Difficulty: Medium

67. Use the \_\_\_\_ method to specify the height and width of a graphical component when you add it to a panel.

a) setPreferredDimensions.

b) setInitialDimensions.

c) setPreferredSize.

d) setInitialSize.

Answer: c

Section Reference: Common Error 10.5

Title: Use the \_\_\_\_ method to specify dimensions of a graphical component .

Difficulty: Easy

68) Assuming that the ClickListener class implements the ActionListener interface, what statement should be used to complete the following code segment?

ClickListener listener = new ClickListener();

JButton myButton = new JButton("Submit");

JPanel myPanel = new JPanel();

myPanel.add(myButton);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ // missing statement

a) myPanel.addActionListener(listener);

b) myButton.addActionListener(listener);

c) myPanel.addActionListener(myButton);

d) myButton.addActionListener(ClickListener);

Answer: b

Title: Identify correct statement to add event listener object to a graphical object

Difficulty: Medium

Section Reference 1: 10.8 Building Applications with Buttons

69) Assume that the TimerListener class implements the ActionListener interface. If the actionPerformed method in TimerListener needs to be executed every second, what statement should be used to complete the following code segment?

ActionListener listener = new TimerListener();

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ // missing statement

timer.start();

a) Timer timer = new Timer(listener);

b) Timer timer = new Timer(1, listener);

c) Timer timer = new Timer(100, listener);

d) Timer timer = new Timer(1000, listener);

Answer: d

Title: Identify correct statement to initialize a Timer object

Difficulty: Easy

Section Reference 1: 10.9 Processing Timer Events

70. The Timer class is found in the \_\_\_\_ package.

a) java.awt.

b) javax.awt.

c) java.swing.

d) javax.swing.

Answer: d

Section 10.9 Processing Timer Events

Title: The Timer class is found in which package?

Difficulty: Easy

71. When you use a timer, you need to define a class that implements the \_\_\_\_ interface.

a) TimerListener

b) TimerActionListener

c) ActionListener

d) StartTimerListener

Answer: c

Section 10.9 Processing Timer Events

Title: The Timer component requires which interface?

Difficulty: Medium

72. The \_\_\_\_ class in the javax.swing package generates a sequence of events, spaced apart at even time intervals.

a) TimeClock

b) Timer

c) Clock

d) StopWatch

Answer: b

Section 10.9 Processing Timer Events

Title: Which class generates events at even time intervals?

Difficulty: Easy

73. Consider the following code snippet:

final RectangleComponent component = new RectangleComponent();

MouseListener listener = new MousePressListener();

component.addMouseListener(listener);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

frame.add(component);

frame.setSize(FRAME\_WIDTH, FRAME\_HEIGHT);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setVisible(true);

Which of the following statements completes this code?

a) private static final int FRAME\_WIDTH = 300;

b) private static final int FRAME\_HEIGHT = 400;

c) Frame frame = new Frame();

d) JFrame frame = new JFrame();

Answer: d

Section 10.10 Mouse Events

Title: Which statement completes this code?

Difficulty: Medium

74. Consider the code snippet below:

public class RectangleComponent extends JComponent

{

private Rectangle box;

private static final int BOX\_X = 100;

private static final int BOX\_Y = 100;

private static final int BOX\_WIDTH = 20;

private static final int BOX\_HEIGHT = 30;

public RectangleComponent()

{

// The rectangle that the paint method draws

box = new Rectangle(BOX\_X, BOX\_Y,

BOX\_WIDTH, BOX\_HEIGHT);

}

public void paintComponent(Graphics g)

{

super.paintComponent(g);

Graphics2D g2 = (Graphics2D) g;

g2.draw(box);

}

public void moveTo(int x, int y)

{

box.setLocation(x, y);

repaint();

}

}

Which statement causes the rectangle to appear at an updated location?

a) repaint();

b) g2.draw(box);

c) box.setLocation(x, y);

d) private Rectangle box;

Answer: a

Section 10.9 Processing Timer Events

Title: Which statement causes the rectangle to appear at an updated location?

Difficulty: Medium

75. Consider the following code snippet:

class MyListener implements ActionListener

{

public void actionPerformed(ActionEvent event)

{

System.out.println(event);

}

}

Timer t = new Timer(interval, listener);

t.start();

What is wrong with this code?

a) The Timer object should be declared before the MyListener class.

b) The listener has not been attached to the Timer object.

c) The Timer object must be declared as final.

d) There is nothing wrong with the code.

Answer: b

Section Reference: Common Error 10.4

Title: What is wrong with this listener code?

Difficulty: Medium

76. You have a class which extends the JComponent class. In this class you have created a painted graphical object. Your code will change the data in the graphical object. What additional code is needed to ensure that the graphical object will be updated with the changed data?

a) You must call the component object's paintComponent() method.

b) You must call the component object's repaint() method.

c) You do not have to do anything – the component object will be automatically repainted when its data changes.

d) You must use a Timer object to cause the component object to be updated.

Answer: b

Section Reference: Common Error 10.6

Title: What code is needed to update a painted component?

Difficulty: Medium

77. The \_\_\_\_ method should be called whenever you modify the shapes that the paintComponent method draws.

a) draw

b) paint

c) paintComponent

d) repaint

Answer: d

Section Reference: Common Error 10.6

Title: The \_\_\_\_ method should be called whenever you modify shapes using paintComponent.

Difficulty: Easy

78. If the user presses and releases a mouse button in quick succession, which methods of the MouseListener interface are called?

a) mousePressed, mouseReleased, and mouseClicked.

b) mousePressed, mouseReleased, and mouseDblClicked

c) Only the mouseClicked method.

d) Only the mouseDblClicked method.

Answer: b

Section 10.10 Mouse Events

Title: Which MouseListener interface methods are called?

Difficulty: Medium

79. Suppose listener is an instance of a class that implements the MouseListener interface. How many methods does listener have?

a) 0

b) 1

c) 3

d) at least 5

Answer: d

Section 10.10 Mouse Events

Title: How many methods does MouseListener have?

Difficulty: Medium

80. Use the \_\_\_\_ method to add a mouse listener to a component.

a) addListener

b) addMouseListener

c) addActionListener

d) addMouseActionListener

Answer: b

Section 10.10 Mouse Events

Title: How to add a mouse listener to a component.

Difficulty: Medium

81. A/an \_\_\_\_ is used to capture mouse events.

a) action listener

b) event listener

c) mouse listener

d) component listener

Answer: c

Section 10.10 Mouse Events

Title: A/an \_\_\_\_ is used to capture mouse events.

Difficulty: Easy

82. You wish to detect when the mouse is moved into a graphical component. Which methods of the MouseListener interface will provide this information?

a) mouseMoved

b) mouseEntered

c) mouseOver

d) mouseIncoming

Answer: b

Section 10.10 Mouse Events

Title: Which MouseListener interface methods are called when the mouse is moved?

Difficulty: Medium

83. Consider the following code snippet:

class MouseClickedListener implements ActionListener

{

public void mouseClicked(MouseEvent event)

{

int x = event.getX();

int y = event.getY();

component.moveTo(x,y);

}

}

What is wrong with this code?

a) The mouseClicked method cannot access the x and y coordinates of the mouse.

b) repaint() method was not called.

c) The class has implemented the wrong interface.

d) There is nothing wrong with this code.

Answer: c

Section 10.10 Mouse Events

Title: What is wrong with this listener code?

Difficulty: Hard

84. Consider the following code snippet:

public class MyMouseListener

{

public void mousePressed(MouseEvent event)

{

double x;

double y;

\_\_\_\_\_\_\_

System.out.println("x: " + x + ", y: " + y);

}

}

Which of the following statements should be in the indicated position to print out where the mouse was pressed?

a) x = event.getXposition(); y = event.getYposition();

b) x = (MouseEvent) getX(); y = (MouseEvent) getY();

c) x = event.printX(); y = event.printY();

d) x = event.getX(); y = event.getY();

Answer: d

Section 10.10 Mouse Events

Title: What statement completes this code?

Difficulty: Hard

85. What does the MouseAdapter class provide?

a) MouseAdapter class implements all of the methods of the MouseListener interface as do nothing methods, eliminating the need to implement the MouseListener interface.

b) MouseAdapter class allows your program to accept input from multiple mice.

c) MouseAdapter class implements all of the methods of the ActionListener interface as do nothing methods, eliminating the need to implement the ActionListener interface.

d) A class can implement the MouseAdapter class to handle mouse events.

Answer: a

Section Reference: Special Topic 10.5

Title: What does the MouseAdapter class provide?

Difficulty: Medium